

Distributed Message Service for Kafka

Referência de API

Edição 11
Data 2024-03-06



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. Todos os direitos reservados.

Nenhuma parte deste documento pode ser reproduzida ou transmitida em qualquer forma ou por qualquer meio sem consentimento prévio por escrito da Huawei Cloud Computing Technologies Co., Ltd.

Marcas registadas e permissões



HUAWEI e outras marcas registadas da Huawei são marcas registadas da Huawei Technologies Co., Ltd.

Todas as outras marcas registadas e os nomes registados mencionados neste documento são propriedade dos seus respectivos detentores.

Aviso

Os produtos, os serviços e as funcionalidades adquiridos são estipulados pelo contrato estabelecido entre a Huawei Cloud e o cliente. Os produtos, os serviços e as funcionalidades descritos neste documento, no todo ou em parte, podem não estar dentro do âmbito de aquisição ou do âmbito de uso. Salvo especificação em contrário no contrato, todas as declarações, informações e recomendações neste documento são fornecidas "TAL COMO ESTÃO" sem garantias ou representações de qualquer tipo, sejam expressas ou implícitas.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. Foram feitos todos os esforços na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações contidas neste documento não constituem uma garantia de qualquer tipo, expressa ou implícita.

Índice

1 Antes de começar.....	1
2 Visão geral de API.....	3
3 Chamada das APIs.....	5
3.1 Fazer uma solicitação de API.....	5
3.2 Autenticação.....	9
3.3 Valores retornados.....	11
4 Primeiros passos.....	13
5 APIs V2 (recomendado).....	15
5.1 Gerenciamento do ciclo da vida.....	15
5.1.1 Criação de uma instância.....	15
5.1.2 Listagem de todas as instâncias.....	35
5.1.3 Consulta de uma instância.....	46
5.1.4 Exclusão de uma instância.....	56
5.1.5 Modificação das informações da instância.....	59
5.1.6 Reinicialização ou exclusão de instâncias em lote.....	69
5.1.7 Obtenção de configurações de instância.....	78
5.1.8 Modificação das configurações de instância.....	83
5.2 Gerenciamento de instâncias.....	88
5.2.1 Redefinição da senha.....	88
5.2.2 Redefinição da senha do Kafka Manager.....	92
5.2.3 Reinicialização do Kafka Manager.....	96
5.2.4 Configuração da criação automática dos tópicos.....	100
5.2.5 Modificação do endereço IP privado para acesso entre VPCs.....	104
5.2.6 Consulta de metadados do cluster do Kafka.....	109
5.2.7 Consulta de detalhes do grupo de consumidores.....	113
5.2.8 Consulta de detalhes do coordenador de uma instância do Kafka.....	119
5.2.9 Reatribuição de réplicas de um tópico para uma instância do Kafka.....	123
5.2.10 Consulta do status de uso de disco dos tópicos.....	129
5.2.11 Consulta de todos os grupos de consumidores.....	134
5.2.12 Exclusão de grupos de consumidores de uma instância do Kafka em lotes.....	139
5.2.13 Criação de um grupo de consumidores.....	144
5.2.14 Edição de um grupo de consumidores.....	148

5.2.15 Inicialização da reatribuição de partição para uma instância do Kafka.....	152
5.2.16 Desativação do Kafka Manager.....	161
5.2.17 Exclusão de uma cota de usuário ou cliente.....	165
5.2.18 Consulta de cotas de usuários ou clientes.....	170
5.2.19 Criação de cotas de usuário ou cliente.....	174
5.2.20 Modificação de cotas de usuários ou clientes.....	180
5.2.21 Redefinição do deslocamento do grupo de consumidores para a posição especificada.....	185
5.3 Smart Connect.....	192
5.3.1 Ativação do Smart Connect (instância de pagamento por uso).....	192
5.3.2 Desativação do Smart Connect (instância de pagamento por uso).....	199
5.3.3 Criação de uma tarefa do Smart Connect.....	203
5.3.4 Consulta de tarefas do Smart Connect.....	225
5.3.5 Consulta de detalhes de tarefas do Smart Connect.....	234
5.3.6 Exclusão de uma tarefa do Smart Connect.....	242
5.3.7 Pausa de uma tarefa do Smart Connect.....	245
5.3.8 Reinicialização de uma tarefa do Smart Connect.....	248
5.3.9 Inicialização de uma tarefa do Smart Connect ou reinicialização de uma tarefa do Smart Connect pausada ou em execução.....	252
5.4 Gerenciamento de modificação de especificação.....	255
5.4.1 Aumento das especificações da instância.....	255
5.4.2 Consulta de informações do produto para modificação da especificação de instância.....	267
5.5 Gerenciamento de tópicos.....	276
5.5.1 Produção de mensagens para o Kafka.....	276
5.5.2 Criação de um tópico para uma instância do Kafka.....	281
5.5.3 Listagem dos tópicos de uma instância de Kafka.....	288
5.5.4 Modificação dos tópicos de uma instância de Kafka.....	293
5.5.5 Exclusão dos tópicos de uma instância de Kafka em lote.....	298
5.5.6 Consulta da lista de partições de um tópico.....	303
5.5.7 Consulta da lista atual de produtores de um tópico.....	308
5.5.8 Consulta de detalhes do tópico.....	313
5.6 Gerenciamento de usuário.....	319
5.6.1 Consulta da lista de usuários.....	319
5.6.2 Criação de um usuário.....	323
5.6.3 Exclusão de usuários em lote.....	328
5.6.4 Redefinição de uma senha do usuário.....	332
5.6.5 Modificação dos parâmetros do usuário.....	336
5.6.6 Consulta das permissões de usuário.....	341
5.6.7 Concessão de permissões de usuário.....	345
5.7 Consulta de mensagem.....	351
5.7.1 Consulta das mensagens.....	351
5.7.2 Consulta de uma mensagem com um deslocamento especificado.....	357
5.7.3 Consulta de uma mensagem com um período de tempo especificado.....	362
5.7.4 Consulta do deslocamento da mensagem mais antiga em uma partição.....	367

5.7.5 Consulta do deslocamento da última mensagem em uma partição.....	371
5.8 Gerenciamento de tarefas em segundo plano.....	375
5.8.1 Listagem das tarefas em segundo plano.....	375
5.8.2 Consulta das tarefas em segundo plano.....	380
5.8.3 Exclusão de uma tarefa em segundo plano.....	384
5.9 Gerenciamento de tags.....	387
5.9.1 Adição ou exclusão das tags em lote.....	387
5.9.2 Listagem das tags de uma instância.....	393
5.9.3 Listagem das tags de um projeto.....	397
5.10 Outras APIs.....	400
5.10.1 Listagem de janelas de tempo de manutenção.....	401
5.10.2 Listagem de informações de AZ.....	405
5.10.3 Consulta da lista de especificações do produto.....	409
5.10.4 Consulta das dimensões de monitoramento de instâncias do Kafka.....	417
6 Permissões e ações suportadas.....	425
7 APIs desatualizadas.....	433
7.1 API V1.....	433
7.1.1 APIs para gerenciamento de instâncias.....	433
7.1.1.1 Criação de uma instância.....	433
7.1.1.2 Consulta de uma instância.....	441
7.1.1.3 Modificação de uma instância.....	447
7.1.1.4 Exclusão de uma instância.....	450
7.1.1.5 Reinicialização ou exclusão de instâncias em lote.....	451
7.1.1.6 Consulta de todas as instâncias.....	454
7.1.1.7 Criação de um tópico em uma instância do Kafka.....	460
7.1.1.8 Consulta de um tópico em uma instância de Kafka.....	462
7.1.1.9 Exclusão dos tópicos em uma instância de Kafka em lote.....	465
7.1.2 Outras APIs.....	466
7.1.2.1 Consulta de informações da AZ.....	466
7.1.2.2 Consulta das especificações do produto.....	468
7.1.2.3 Consultando janelas de tempo de manutenção.....	473
7.2 API V2.....	475
7.2.1 Adição de partições a um tópico para uma instância do Kafka.....	475
7.2.2 Redefinição do deslocamento do grupo de consumidores para a posição especificada.....	477
7.2.3 Modificação das cotas de tarefas de despejo.....	484
7.2.4 Criação de um pedido para desativar os conectores de despejo de uma instância.....	487
7.2.5 Criação de uma tarefa de despejo.....	491
7.2.6 Listagem de tarefas de despejo.....	499
7.2.7 Consulta de uma tarefa de despejo.....	503
7.2.8 Exclusão de uma tarefa de despejo.....	509
8 Apêndice.....	513

8.1 Código de status.....	513
8.2 Códigos de erro.....	516
8.3 Status da instância.....	537
8.4 Obtenção de um ID de projeto.....	538
8.5 Obtenção do nome e ID da conta.....	539
A Histórico de alterações.....	540

1 Antes de começar

Bem-vindo à *Referência de API do Distributed Message Service for Kafka*. Distributed Message Service (DMS) for Kafka é um serviço de enfileiramento de mensagens baseado no Apache Kafka de código aberto. Ele fornece instâncias premium do Kafka com recursos isolados de computação, armazenamento e largura de banda. DMS for Kafka permite que você aplique recursos, configure tópicos, partições e réplicas com base nos requisitos de serviço. O serviço pode ser usado fora da caixa e libera você de implementação e O&M para que você possa se concentrar no desenvolvimento ágil de suas aplicações.

Este documento descreve as funções, sintaxe, parâmetros e exemplos das interfaces de programação de aplicações (APIs) do DMS for Kafka.

O DMS for Kafka suporta APIs REST (Representational State Transfer), permitindo que você chame APIs usando HTTPS. Para obter detalhes sobre chamada de API, consulte [Chamada das APIs](#).

Pontos de extremidade

Um ponto de extremidade é o **endereço de solicitação** para chamar uma API. Os pontos de extremidade variam de acordo com os serviços e as regiões. Para obter os pontos de extremidade de todos os serviços, consulte [Regiões e pontos de extremidade](#).

Conceitos

- Conta

Uma conta é criada após o registro bem-sucedido e tem permissões de acesso total para todos os seus serviços e recursos em nuvem. Ela pode ser usada para redefinir senhas de usuários e conceder permissões ao usuário. A conta é uma entidade de pagamento, que não deve ser utilizada diretamente para realizar gerenciamento de rotina. Para garantir a segurança da conta, crie usuários do Identity and Access Management (IAM) e conceda a eles permissões para o gerenciamento de rotina.

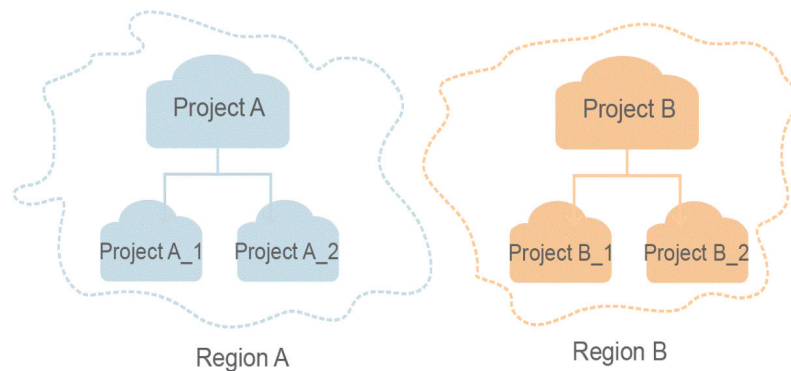
- Usuário do IAM

Um usuário do IAM é criado usando uma conta para usar os serviços em nuvem. Cada usuário do IAM tem suas próprias credenciais de identidade (senha e chaves de acesso).

O nome da conta, o nome de usuário e a senha serão necessários para a autenticação da API.

- **Região**
As regiões são divididas de acordo com as dimensões de localização geográfica e latência da rede. Serviços públicos, como Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Virtual Private Cloud (VPC), Elastic IP (EIP) e Image Management Service (IMS), são compartilhados na mesma região. As regiões são classificadas em regiões universais e dedicadas. Uma região universal fornece serviços de nuvem universal para locatários comuns. Uma região dedicada fornece serviços específicos para locatários específicos.
- **Zona de disponibilidade (AZ)**
Uma zona de disponibilidade (AZ) compreende um ou mais data centers físicos equipados com instalações independentes de ventilação, incêndio, água e eletricidade. Computação, rede, armazenamento e outros recursos em uma AZ são logicamente divididos em vários clusters. As AZs dentro de uma região são interconectadas usando fibras ópticas de alta velocidade para suportar sistemas de alta disponibilidade entre as AZs.
- **Projeto**
Projetos agrupam e isolam recursos (incluindo recursos de computação, armazenamento e rede) entre regiões físicas. Um projeto padrão é fornecido para cada região, e subprojetos podem ser criados em cada projeto padrão. Os usuários podem receber permissões para acessar todos os recursos em um projeto específico. Para um controle de acesso mais refinado, crie subprojetos em um projeto e compre recursos nos subprojetos. Os usuários podem então receber permissões para acessar apenas recursos específicos nos subprojetos.

Figura 1-1 Modelo de isolamento do projeto



- **Projeto empresarial**
Projetos empresariais agrupam e gerenciam recursos entre regiões. Os recursos em diferentes projetos empresariais são logicamente isolados. Um projeto empresarial pode conter recursos de várias regiões e os recursos podem ser adicionados ou removidos de projetos empresariais.
Para obter detalhes sobre como obter os IDs e recursos de projetos empresariais, consulte [Guia de usuário do Enterprise Management](#).

2 Visão geral de API

Tabela 2-1 APIs

APIs	Descrição
APIs de gerenciamento do ciclo de vida	Criar uma instância, listar todas as instâncias, consultar uma instância, excluir uma instância, modificar informações de instância, reiniciar ou excluir instâncias em lotes e obter ou modificar especificações de instância.
APIs de gerenciamento de instâncias	Redefinir a senha, redefinir a senha do Kafka Manager, reiniciar o Kafka Manager, configurar a criação automática de tópicos, modificar o endereço IP privado para acesso entre VPCs, consultar metadados do cluster do Kafka, consultar detalhes do grupo de consumidores, consultar detalhes do coordenador de uma instância do Kafka, reatribuir réplicas de um tópico para uma instância do Kafka, consultar o status de uso do disco de tópicos, consultar todos os grupos de consumidores, excluir em lote grupos de consumidores de uma instância do Kafka, criar um grupo de consumidores, editar um grupo de consumidores, iniciar a reatribuição de partição para uma instância do Kafka, desativar o Kafka Manager, excluir uma cota de usuário ou cliente, consultar cotas de usuário ou cliente, criar cotas de usuário ou cliente, modificar cotas de usuário ou cliente e redefinir o deslocamento do grupo de consumidores para a posição especificada.
APIs do Smart Connect	Ativar o Smart Connect (instância de pagamento por uso), desativar o Smart Connect (instância de pagamento por uso), criar uma tarefa do Smart Connect, consultar tarefas do Smart Connect, consultar detalhes da tarefa do Smart Connect, excluir uma tarefa do Smart Connect, pausar uma tarefa do Smart Connect, reiniciar uma tarefa do Smart Connect e iniciar uma tarefa do Smart Connect ou reiniciar uma tarefa do Smart Connect pausada ou em execução.
APIs de gerenciamento de modificação de especificações	Consultar informações do produto para modificação da especificação da instância e aumentar as especificações da instância.

APIs	Descrição
APIs de gerenciamento de tópicos	Criar um tópico para uma instância do Kafka, produzir mensagens para o Kafka, listar tópicos de uma instância do Kafka, modificar tópicos de uma instância do Kafka, excluir tópicos em lote de uma instância do Kafka, consultar a lista de partição de um tópico, consultar a lista de produtores atual de um tópico e consultar os detalhes do tópico.
APIs de gerenciamento de usuários	Listar usuários, criar um usuário, excluir usuários em lote, redefinir a senha do usuário, modificar parâmetros de usuário, consultar permissões de usuário e configurar permissões de usuário.
APIs de consulta de mensagens	Consultar mensagens, consultar uma mensagem com um deslocamento especificado, consultar uma mensagem com um período de tempo especificado, consultar o deslocamento da mensagem mais antiga em uma partição e consultar o deslocamento da mensagem mais recente em uma partição.
APIs de gerenciamento de tarefas em segundo plano	Listar tarefas em segundo plano, consultar uma tarefa em segundo plano e excluir uma tarefa em segundo plano.
APIs de gerenciamento de tags	Adicionar ou excluir tags em lotes, listar tags de uma instância e listar tags de um projeto.
Outras APIs	Listar informações de AZ, consultar especificações de produto, consultar dimensões de monitoramento de instância do Kafka e listar janelas de tempo de manutenção.

3 Chamada das APIs

3.1 Fazer uma solicitação de API

Esta seção descreve a estrutura de uma solicitação de API REST e usa a API do IAM para **obtenção de um token de usuário** como um exemplo para demonstrar como chamar uma API. O token obtido pode então ser usado para autenticar a chamada das outras APIs.

URI de solicitação

Um URI de solicitação está no seguinte formato:

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

Embora um URI de solicitação esteja incluído no cabeçalho da solicitação, a maioria das linguagens de programação ou estruturas exigem que o URI de solicitação seja transmitido separadamente.

Tabela 3-1 Parâmetros em um URI

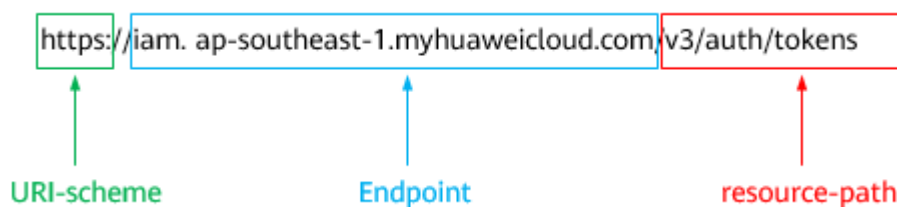
Parâmetro	Descrição
URI-scheme	Protocolo usado para transmitir solicitações. Todas as APIs usam HTTPS.
Endpoint	Nome de domínio ou endereço IP do servidor que suporta o serviço REST. O ponto de extremidade varia entre serviços em diferentes regiões. Ele pode ser obtido em Regiões e pontos de extremidade . Por exemplo, o ponto de extremidade do IAM na região CN-Hong Kong é iam.ap-southeast-1.myhuaweicloud.com .
resource-path	Caminho de acesso de uma API para executar uma operação especificada. Obtenha o caminho a partir do URI de uma API. Por exemplo, o caminho de recurso da API usada para obtain a user token é /v3/auth/tokens .

Parâmetro	Descrição
query-string	Parâmetro de consulta, que é opcional. Certifique-se de que um ponto de interrogação (?) seja incluído antes de cada parâmetro de consulta que esteja no formato de <i>Parameter name=Parameter value</i> . Por exemplo, ?limit=10 indica que um máximo de 10 registros de dados serão exibidos.

Por exemplo, para obter um token do IAM na região **CN-Hong Kong**, obtenha o ponto de extremidade do IAM (**iam.ap-southeast-1.myhuaweicloud.com**) para essa região e o **resource-path** (**/v3/auth/tokens**) no URI da API usada para **obter um token de usuário**. Em seguida, construa o URI da seguinte forma:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figura 3-1 Exemplo de URI



NOTA

Para simplificar a exibição de URI neste documento, cada API é fornecida apenas com um **resource-path** e um método de solicitação. O **URI-scheme** de todas as APIs é **HTTPS**, e os pontos de extremidade de todas as APIs na mesma região são idênticos.

Métodos de solicitação

O protocolo HTTP define os seguintes métodos de solicitação que podem ser usados para enviar uma solicitação ao servidor:

- **GET**: solicita que um servidor retorne os recursos especificados.
- **PUT**: solicita que um servidor atualize os recursos especificados.
- **POST**: solicita que um servidor adicione recursos ou execute operações especiais.
- **DELETE**: solicita que um servidor exclua recursos especificados, por exemplo, objetos.
- **HEAD**: o mesmo que GET, exceto que o servidor deve retornar apenas o cabeçalho da resposta.
- **PATCH**: solicita que um servidor atualize uma parte de um recurso especificado. Se o recurso não existir, um novo recurso pode ser criado usando o método PATCH.

Por exemplo, no caso da API usada para **obter um token de usuário**, o método de solicitação é **POST**. A solicitação é o seguinte:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Cabeçalho da solicitação

Você também pode adicionar campos de cabeçalho adicionais a uma solicitação, como os campos exigidos por um método de URI ou de HTTP especificado. Por exemplo, para

solicitar as informações de autenticação, adicione **Content-Type**, que especifica o tipo de corpo da solicitação.

Tabela 3-2 lista os campos comuns do cabeçalho da solicitação.

Tabela 3-2 Campos comuns de cabeçalho de solicitação

Nome	Descrição	Obrigatório	Exemplo de valor
Host	Informações do servidor de solicitação, que são obtidas do URL de uma API de serviço. O valor está no formato de <i>Hostname:Port number</i> . Se nenhuma porta for especificada, a porta padrão será usada. Para HTTPS, a porta 443 é usada por padrão.	Não Este parâmetro é obrigatório para a autenticação de AK/SK.	code.test.com ou code.test.com:443
Content-Type	Tipo ou formato do corpo da mensagem. O valor padrão application/json é recomendado. Outros valores deste campo serão fornecidos para APIs específicas, se houver.	Sim	application/json
Content-Length	Comprimento do corpo da solicitação. A unidade é byte.	Não	3495
X-Project-Id	ID do projeto. Obtenha o ID do projeto seguindo as instruções em Obtenção de um ID de projeto .	Não Este campo é obrigatório para solicitações que usam autenticação AK/SK no cenário da Dedicated Cloud (DeC) ou cenário de vários projetos.	e9993fc787d94b6c886cbaa340f9c0f4

Nome	Descrição	Obrigatório	Exemplo de valor
X-Auth-Token	Token do usuário. O token de usuário é uma resposta à API usada para obter um token de usuário . Esta API é a única que não requer autenticação. Depois que a solicitação é processada, o valor de X-Subject-Token no cabeçalho da resposta é o valor do token.	Não Obrigatório para autenticação baseada em token.	O seguinte é parte de um exemplo de token: MIIPAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ

 **NOTA**

Além de suportar a autenticação baseada em token, as APIs de serviço em nuvem também suportam a autenticação usando a AK/SK. Durante a autenticação baseada em AK/SK, um SDK é usado para assinar a solicitação, e os campos de cabeçalho **Authorization** (informações de assinatura) e **X-Sdk-Date** (hora em que a solicitação é enviada) são adicionados automaticamente à solicitação.

Para obter mais informações, consulte "Autenticação baseada em AK/SK" em [Autenticação](#).

A API usada para **obter um token de usuário** não requer autenticação. Portanto, apenas o campo **Content-Type** precisa ser adicionado às solicitações para chamar a API. Um exemplo de tais solicitações é o seguinte:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

Corpo de solicitação (opcional)

O corpo de uma solicitação geralmente é enviado em um formato estruturado, conforme especificado no campo de cabeçalho **Content-type**, como JSON ou XML. O corpo da solicitação transfere o conteúdo, exceto o cabeçalho da solicitação.

O corpo da solicitação varia entre as APIs. Algumas APIs não exigem o corpo da solicitação, como as APIs solicitadas usando os métodos GET e DELETE.

No caso da API usada para **obter um token de usuário**, os parâmetros de solicitação e a descrição do parâmetro podem ser obtidos a partir da solicitação da API. O seguinte fornece um exemplo de solicitação com um corpo incluído. Substitua *username*, *domainname*, ******* (senha de logon) e *xxxxxxxxxxxxxxxxxxx* (nome do projeto) pelos valores reais (por exemplo, ap-southeast-1). O nome do projeto pode ser obtido em [Regiões e pontos de extremidade](#).

 **NOTA**

O parâmetro **scope** especifica onde um token entra em vigor. Você pode definir **scope** para uma conta ou um projeto em uma conta. No exemplo a seguir, o token tem efeito somente para os recursos em um projeto especificado. Para obter mais informações sobre essa API, consulte [Obtenção de um token de usuário](#).

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

Se todos os dados exigidos por uma solicitação estiverem disponíveis, você poderá enviar a solicitação para chamar a API por meio de [curl](#), [Postman](#) ou codificação. Na resposta à API usada para obter um token de usuário, **x-subject-token** é o token de usuário desejado. Este token pode ser usado para autenticar a chamada de outras APIs.

3.2 Autenticação

As solicitações para chamar uma API podem ser autenticadas usando um dos seguintes métodos:

- Autenticação baseada em token: as solicitações são autenticadas usando um token.
- Autenticação AK/SK: as solicitações são criptografadas usando pares de AK/SK. A autenticação baseada em AK/SK é recomendada porque é mais segura do que a autenticação baseada em token.

Autenticação baseada em token

NOTA

O período de validade de um token é de 24 horas. Ao usar um token para autenticação, armazene-o em cache para evitar chamadas frequentes à API do IAM usada para obter um token de usuário.

Um token especifica permissões temporárias em um sistema de computador. Durante a autenticação da API usando um token, o token é adicionado para solicitar cabeçalhos para obter permissões para chamar a API. Você pode obter um token [chamando uma API](#).

Um serviço de nuvem pode ser implementado como um serviço de nível de projeto ou serviço global.

- Para um serviço no nível do projeto, você precisa obter um token no nível do projeto. Ao chamar a API, defina **auth.scope** no corpo da solicitação para **project**.
- Para um serviço global, você precisa obter um token global. Ao chamar a API, defina **auth.scope** no corpo da solicitação como **domain**.

Ao chamar a API usada para **obter um token de usuário**, você deve definir **auth.scope** no corpo da solicitação como **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

Depois que um token é obtido, o campo de cabeçalho **X-Auth-Token** deve ser adicionado às solicitações para especificar o token ao chamar outras APIs. Por exemplo, se o token for **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** pode ser adicionado a uma solicitação da seguinte forma:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

Autenticação baseada em AK/SK

NOTA

A autenticação baseada em AK/SK suporta solicitações de API com um corpo não maior que 12 MB. Para solicitações de API com um corpo maior, a autenticação baseada em token é recomendada.

Na autenticação baseada em AK/SK, AK/SK é usada para assinar solicitações e a assinatura é então adicionada aos cabeçalhos da solicitação para autenticação.

- AK: o ID da chave de acesso, que é um identificador exclusivo usado em conjunto com uma chave de acesso secreta para assinar solicitações criptograficamente.
- SK: a chave de acesso secreta usada em conjunto com um AK para assinar solicitações criptograficamente. Ela identifica um remetente da solicitação e impede que a solicitação seja modificada.

Na autenticação baseada em AK/SK, você pode usar uma AK/SK para assinar solicitações com base no algoritmo de assinatura ou usar o SDK de assinatura para assinar solicitações. Para obter detalhes sobre como assinar solicitações e usar o SDK de assinatura, consulte [Guia de assinatura de solicitação de API](#).

AVISO

O SDK de assinatura é usado apenas para solicitações de assinatura e é diferente dos SDKs fornecidos pelos serviços.

3.3 Valores retornados

Código de status

Depois de enviar uma solicitação, você receberá uma resposta, incluindo o código de status, o cabeçalho da resposta e o corpo da resposta.

Um código de status é um grupo de dígitos que variam de 1xx a 5xx. Indica o status de uma resposta. Para obter mais informações, consulte [Código de status](#).

Se o código de status **201** for retornado para a chamada da API para [obtenção de um token de usuário](#), a solicitação é bem-sucedida.

Cabeçalho de resposta

Um cabeçalho de resposta corresponde a um cabeçalho de solicitação, por exemplo, **Content-type**.

[Figura 3-2](#) mostra o cabeçalho de resposta para a API de [obtenção de um token de usuário](#), no qual **x-subject-token** é o token de usuário desejado. Em seguida, você pode usar o token para autenticar a chamada de outras API.

Figura 3-2 Cabeçalho da resposta à solicitação para obter um token de usuário

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → MIYXQYJKoZiIhvcNAQcCoIYVTJCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w00BBwGgghacBIIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6IjI6IiwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkajACgkqO1wi4JIGzrpd18LGXK5bdfq4iqHCYb8P4NaY0NYejcAgzJVeFYtLWT1GSO0zxKZmiQHq82HBqHdgIZO9fuEbL5dMhdavj+33wEI
xHRCe9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jqglFKNPQuFSOU8+uSsttVwRtnfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUBpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block
```

(Opcional) Corpo de resposta

O corpo de uma resposta geralmente é retornado em formato estruturado (como JSON ou XML), conforme especificado no campo de cabeçalho **Content-type**. O corpo da resposta transfere o conteúdo, exceto o cabeçalho da resposta.

Você pode obter um token [chamando uma API](#). Veja a seguir parte do corpo da resposta da API para obter um token de usuário.

```
{
  "token": {
```

```
"expires_at": "2019-02-13T06:52:13.855000Z",
"methods": [
  "password"
],
"catalog": [
  {
    "endpoints": [
      {
        "region_id": "XXXXXX",
        .....

```

Se ocorrer um erro durante a chamada da API, o sistema retornará um código de erro e uma mensagem para você. A seguir, mostra o formato de um corpo de resposta de erro:

```
{
  "error_msg": "The format of message is error",
  "error_code": "AS.0001"
}
```

Nas informações anteriores, **error_code** é um código de erro, e **error_msg** descreve o erro.

4 Primeiros passos

Cenários

Esta seção descreve como chamar uma API para criar uma instância do Kafka e personalizar os recursos de computação e o espaço de armazenamento da instância com base nos requisitos de serviço.

Para obter detalhes sobre como chamar as APIs, consulte [Chamada das APIs](#).

Pré-requisitos

- Ponto de extremidade do IAM obtido de [Regiões e pontos de extremidade](#).
- Ponto de extremidade do Kafka obtido de [Regiões e pontos de extremidade](#).

Criação de uma instância do Kafka

Veja a seguir um exemplo de solicitação para criar uma instância do Kafka:

```
{
  "name": "kafka-demo",
  "engine": "kafka",
  "engine_version": 2.7,
  "broker_num": 3,
  "storage_space": 300,
  "vpc_id": "ead6c5ff-xxx-9ba91820e72c",
  "security_group_id": "aa75ae22-xxx-a9dec8c73220",
  "subnet_id": "3cb6afa2-xxx-05a7f671d6a8",
  "available_zones": [
    "effdcbcxxx6b42f56533"
  ],
  "product_id": "c6.2u4g.cluster",
  "storage_spec_code": "dms.physical.storage.high.v2"
}
```

- **name**: nome da instância.
- **engine**: mecanismo de mensagens. O valor é **kafka**.
- **engine_version**: versão do mecanismo de mensagens.
- **broker_num**: número de proxies.
- **storage_space**: espaço de armazenamento de mensagens em GB. Para obter detalhes sobre o intervalo de valores, consulte [Criação de uma instância](#).
- **vpc_id**: ID da VPC em que a instância do Kafka reside. Obtenha o valor usando a API descrita em [Criação de uma instância](#).

- **security_group_id**: ID do grupo de segurança. Obtenha o valor usando a API descrita em [Criação de uma instância](#).
- **subnet_id**: ID da sub-rede da VPC. Obtenha o valor usando a API descrita em [Criação de uma instância](#).
- **available_zones**: ID da AZ em que a instância reside. O valor não pode ser vazio ou nulo. Obtenha o valor chamando a API descrita em [Consulta das informações da AZ](#).
- **product_id**: ID do produto. Obtenha o valor chamando a API descrita em [Consulta de especificações do produto](#).
- **storage_spec_code**: especificação de I/O de armazenamento. Para obter detalhes sobre o intervalo de valores, consulte [Criação de uma instância](#).

5 APIs V2 (recomendado)

5.1 Gerenciamento do ciclo da vida

5.1.1 Criação de uma instância

Função

Esta API é usada para criar uma instância

. Há suporte para os modos de cobrança de pagamento por uso e anual/mensal.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{engine}/{project_id}/instances

Tabela 5-1 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens. Padrão: kafka
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.

Parâmetros de solicitação

Tabela 5-2 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
name	Sim	String	Nome da instância. Um nome de instância pode conter de 4 a 64 caracteres. Apenas letras, dígitos, sublinhados (_) e hifens (-) são permitidos.
description	Não	String	Descrição de uma instância. A descrição suporta até 1024 caracteres. NOTA A barra invertida (\) e as aspas (") são caracteres especiais para mensagens JSON. Ao usar esses caracteres em um valor de parâmetro, adicione o caractere de escape (\) antes dos caracteres, por exemplo, \ e " .
engine	Sim	String	Mecanismo de mensagens. Valor: kafka .
engine_version	Sim	String	Versão do mecanismo de mensagens. Opções: <ul style="list-style-type: none"> ● 1.1.0 ● 2.7 ● 3.x
broker_num	Sim	Integer	Número de corretores.
storage_space	Sim	Integer	Espaço de armazenamento de mensagens em GB. Intervalo de valores: <ul style="list-style-type: none"> ● 300–300.000 GB para instâncias do Kafka c6.2u4g.cluster ● 300–600.000 GB para instâncias do Kafka c6.4u8g.cluster ● 300–1.500.000 GB para instâncias do Kafka c6.8u16g.cluster ● 300–1.500.000 GB para instâncias do Kafka c6.12u24g.cluster ● 300–1.500.000 GB para instâncias do Kafka c6.16u32g.cluster

Parâmetro	Obrigatório	Tipo	Descrição
access_user	Não	String	<p>Este parâmetro é obrigatório quando ssl_enable é definido como true. Este parâmetro é inválido quando ssl_enable é definido como false.</p> <p>Um nome de usuário deve começar com uma letra e apenas letras, dígitos, hifens (-) e sublinhados (_) são permitidos. Pode conter de 4 a 64 caracteres.</p>
password	Não	String	<p>Este parâmetro é obrigatório quando ssl_enable é definido como true. Este parâmetro é inválido quando ssl_enable é definido como false.</p> <p>Senha da instância.</p> <p>A senha deve atender aos seguintes requisitos de complexidade:</p> <ul style="list-style-type: none"> ● Pode conter de 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais incluem (~!@#\$%^&*()-_+= [{}]:'";<.>/?) e espaços, e não podem começar com um hífen (-).
vpc_id	Sim	String	<p>ID de VPC</p> <p>Para obtê-lo, faça logon no console da VPC e visualize o ID da VPC na página de detalhes da VPC.</p>
security_group_id	Sim	String	<p>ID do grupo de segurança ao qual a instância pertence.</p> <p>Para obtê-lo, faça logon no console da VPC e visualize o ID do grupo de segurança na página de detalhes do grupo de segurança.</p>

Parâmetro	Obrigatório	Tipo	Descrição
subnet_id	Sim	String	<p>Informações da sub-rede.</p> <p>Para obtê-lo, faça logon no console da VPC e clique na sub-rede de destino na página Subnets. Você pode visualizar o ID da rede na página exibida.</p>
available_zones	Sim	Array of strings	<p>ID da AZ onde os corretores de instância residem e que tem recursos disponíveis. Obtenha o ID da AZ consultando a Listagem de informações da AZ.</p> <p>Esse parâmetro não pode ser vazio ou nulo.</p> <p>Uma instância do Kafka pode ser implementada em 1 AZ ou pelo menos 3 AZs. Ao especificar as AZs para corretores, use vírgulas (,) para separar várias AZs.</p>
product_id	Sim	String	<p>ID do produto.</p> <p>Você pode obter um ID de produto na Consulta à lista de especificações do produto.</p>
kafka_manager_user	Não	String	<p>Nome de usuário para efetuar logon no Kafka Manager. O nome de usuário consiste de 4 a 64 caracteres e pode conter letras, dígitos, hifens (-) e sublinhados (_).</p>

Parâmetro	Obrigatório	Tipo	Descrição
kafka_manager_password	Não	String	<p>Senha para efetuar logon no Kafka Manager.</p> <p>A senha deve atender aos seguintes requisitos de complexidade:</p> <ul style="list-style-type: none"> ● Pode conter de 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais incluem (~!@#\$%^&*()-_+= [{}]:",<.>/?) e espaços, e não podem começar com um hífen (-).
maintain_begin	Não	String	Hora em que a janela de manutenção é iniciada. O formato é HH:mm.
maintain_end	Não	String	Hora em que a janela de manutenção termina. O formato é HH:mm.
enable_publicip	Não	Boolean	<p>Se permitir o acesso público. Por padrão, o acesso público está desativado.</p> <ul style="list-style-type: none"> ● true: ativado ● false: desativado
publicip_id	Não	String	<p>ID do EIP vinculado à instância.</p> <p>Use vírgulas (,) para separar vários IDs de EIP.</p> <p>Esse parâmetro é obrigatório se o acesso público estiver ativado (ou seja, enable_publicip estiver definido como true).</p>
ssl_enable	Não	Boolean	<p>Se ativar a criptografia SSL para acesso.</p> <p>Essa configuração é corrigida assim que a instância é criada.</p> <ul style="list-style-type: none"> ● true: ativada ● false: desativada

Parâmetro	Obrigatório	Tipo	Descrição
kafka_security_protocol	Não	String	<p>Protocolo de segurança a ser usado após a ativação do SASL. Este parâmetro é obrigatório se a autenticação SASL estiver ativada (ssl_enable=true).</p> <p>Se esse parâmetro for deixado em branco, a autenticação SASL_SSL será ativada por padrão.</p> <p>Essa configuração é corrigida assim que a instância é criada.</p> <ul style="list-style-type: none"> ● SASL_SSL: os dados são criptografados com certificados SSL para transmissão de alta segurança. ● SASL_PLAINTEXT: os dados são transmitidos em texto não criptografado com autenticação de nome de usuário e senha. Este protocolo usa o mecanismo SCRAM-SHA-512 e oferece alto desempenho.
sasl_enabled_mechanisms	Não	Array of strings	<p>Mecanismo de autenticação a ser usado depois que o SASL for ativado. Este parâmetro é obrigatório se a autenticação SASL estiver ativada (ssl_enable=true).</p> <p>Se este parâmetro for deixado em branco, a autenticação PLAIN será ativada por padrão.</p> <p>Selecione ambos ou um dos seguintes mecanismos para a autenticação SASL. Opções:</p> <ul style="list-style-type: none"> ● PLAIN: verificação simples de nome de usuário e senha. ● SCRAM-SHA-512: verificação da credencial do usuário, que é mais segura do que PLAIN.

Parâmetro	Obrigatório	Tipo	Descrição
retention_policy	Não	String	Ação a ser tomada quando o uso da memória atingir o limite de capacidade do disco. O valor pode ser: <ul style="list-style-type: none"> ● produce_reject: novas mensagens não podem ser criadas. ● time_base: as mensagens mais antigas são excluídas.
ipv6_enable	Não	Boolean	Indica se deve ativar o IPv6. Esse parâmetro está disponível somente quando a VPC oferece suporte a IPv6. Padrão: false
disk_encrypted_enable	Não	Boolean	Indica se a criptografia de disco deve ser ativada.
disk_encrypted_key	Não	String	Key de criptografia de disco. Se a criptografia de disco não estiver ativada, esse parâmetro será deixado em branco.
connector_enable	Não	Boolean	Se deve ativar o despejo de mensagens. Por padrão, o despejo de mensagens está desativado.
enable_auto_topic	Não	Boolean	Se deve ativar a criação automática de tópicos. <ul style="list-style-type: none"> ● true: ativada ● false: desativada Se estiver ativada, um tópico será criado automaticamente com 3 partições e 3 réplicas quando uma mensagem for criada ou recuperada de um tópico que não existe. O valor padrão é false .

Parâmetro	Obrigatório	Tipo	Descrição
storage_spec_code	Sim	String	<p>Especificação de I/O de armazenamento.</p> <p>Intervalo de valores:</p> <ul style="list-style-type: none"> ● dms.physical.storage.high.v2: I/O alta ● dms.physical.storage.ultra.v2: I/O ultra-alta <p>Para obter detalhes sobre como selecionar um tipo de disco, consulte "Tipos de disco e desempenho de disco" na Visão geral de serviço do EVS.</p>
enterprise_project_id	Não	String	ID do projeto empresarial. Esse parâmetro é obrigatório para uma conta de projeto empresarial.
tags	Não	Array of TagEntity objects	Lista de tags.
arch_type	Não	String	<p>Arquitetura da CPU. Atualmente com suporte: x86.</p> <p>Intervalo de valores:</p> <ul style="list-style-type: none"> ● X86
vpc_client_plain	Não	Boolean	Acesso de texto não criptografado dentro da VPC.
bss_param	Não	BssParam object	<p>Parâmetro relacionado ao modo de cobrança anual/mensal.</p> <p>Se esse parâmetro for deixado em branco, o modo de cobrança é pagamento por uso por padrão. Se este parâmetro não for deixado em branco, o modo de cobrança é anual/mensal.</p>

Tabela 5-3 TagEntity

Parâmetro	Obrigatório	Tipo	Descrição
key	Não	String	Chave de tag, que: <ul style="list-style-type: none"> ● Deve ser especificada. ● Deve ser exclusiva para a mesma instância. ● Pode conter de 1 a 128 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>_.:=-@</code> ● Não pode começar ou terminar com um espaço.
value	Não	String	Valor da tag. <ul style="list-style-type: none"> ● Pode conter de 0 a 255 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>_.:=-@</code>

Tabela 5-4 BssParam

Parâmetro	Obrigatório	Tipo	Descrição
is_auto_renew	Não	Boolean	Se a renovação automática está ativada. Opções: <ul style="list-style-type: none"> ● true: a renovação automática está ativada. ● false: a renovação automática não está ativada. Por padrão, a renovação automática está desativada.
charging_mode	Não	String	Modo de cobrança. Este parâmetro especifica um modo de pagamento. Opções: <ul style="list-style-type: none"> ● prePaid: cobrança anual/mensal. ● postPaid: cobrança de pagamento por uso. O valor padrão é postPaid .

Parâmetro	Obrigatório	Tipo	Descrição
is_auto_pay	Não	Boolean	Especifica se o pedido é pago automática ou manualmente. Opções: <ul style="list-style-type: none"> ● true: o pedido será pago automaticamente. ● false: o pedido deve ser pago manualmente. O modo de pagamento padrão é manual.
period_type	Não	String	Tipo de período de assinatura. Opções: <ul style="list-style-type: none"> ● month ● year: Este parâmetro é válido e obrigatório somente quando chargingMode estiver definido como prePaid . **
period_num	Não	Integer	Períodos assinados. Opções: <ul style="list-style-type: none"> ● Se periodType for month, o valor varia de 1 a 9. ● Se periodType for year, o valor varia de 1 a 3. Este parâmetro é válido e obrigatório apenas quando chargingMode está definido como prePaid .

Parâmetros de resposta

Código de status: 200

Tabela 5-5 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
instance_id	String	ID da instância.

Exemplo de solicitações

- Criação de uma instância do Kafka de pagamento por uso cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB
 POST `https://{endpoint}/v2/{engine}/{project_id}/instances`

```
{
  "name" : "kafka-test",
  "description" : "",
  "engine" : "kafka",
  "engine_version" : "2.7",
  "storage_space" : 300,
  "vpc_id" : "*****-9b4a-44c5-a964-*****",
  "subnet_id" : "*****-8fbf-4438-ba71-*****",
  "security_group_id" : "*****-e073-4aad-991f-*****",
  "available_zones" : [ "*****706d4c1fb0eb72f0*****" ],
  "product_id" : "c6.2u4g.cluster",
  "ssl_enable" : true,
  "kafka_security_protocol" : "SASL_SSL",
  "sasl_enabled_mechanisms" : [ "SCRAM-SHA-512" ],
  "storage_spec_code" : "dms.physical.storage.ultra.v2",
  "broker_num" : 3,
  "arch_type" : "X86",
  "enterprise_project_id" : "0",
  "access_user" : "*****",
  "password" : "*****",
  "enable_publicip" : true,
  "tags" : [ {
    "key" : "aaa",
    "value" : "111"
  } ],
  "retention_policy" : "time_base",
  "disk_encrypted_enable" : true,
  "disk_encrypted_key" : "*****-b953-4875-a743-*****",
  "publicip_id" : "*****-88fc-4a8c-86d0-
*****-16af-455d-8d54-*****",
  "vpc_client_plain" : true,
  "enable_auto_topic" : true
}
```

- Criação de uma instância anual/mensal do Kafka cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

POST https://{endpoint}/v2/{engine}/{project_id}/instances

```
{
  "name" : "kafka-test1",
  "description" : "",
  "engine" : "kafka",
  "engine_version" : "2.7",
  "storage_space" : 300,
  "vpc_id" : "*****-9b4a-44c5-a964-*****",
  "subnet_id" : "*****-8fbf-4438-ba71-*****",
  "security_group_id" : "*****-e073-4aad-991f-*****",
  "available_zones" : [ "*****706d4c1fb0eb72f0*****" ],
  "product_id" : "c6.2u4g.cluster",
  "ssl_enable" : true,
  "kafka_security_protocol" : "SASL_SSL",
  "sasl_enabled_mechanisms" : [ "SCRAM-SHA-512" ],
  "storage_spec_code" : "dms.physical.storage.ultra.v2",
  "broker_num" : 3,
  "arch_type" : "X86",
  "enterprise_project_id" : "0",
  "access_user" : "*****",
  "password" : "*****",
  "enable_publicip" : true,
  "tags" : [ {
    "key" : "aaa",
    "value" : "111"
  } ],
  "retention_policy" : "time_base",
  "publicip_id" : "*****-88fc-4a8c-86d0-
*****-16af-455d-8d54-*****",
  "vpc_client_plain" : true,
  "enable_auto_topic" : true,
}
```

```
"bss_param" : {  
  "charging_mode" : "prePaid",  
  "period_type" : "month",  
  "period_num" : 1,  
  "is_auto_pay" : true  
}
```

Exemplo de respostas

Código de status: 200

Instância criada com sucesso.

```
{  
  "instance_id" : "8959ab1c-7n1a-yyb1-a05t-93dfc361b32d"  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- Criação de uma instância do Kafka de pagamento por uso cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateInstanceByEngineSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
        plaintext, which has great security risks. It is recommended that the AK and  
        SK be stored in ciphertext in configuration files or environment variables  
        and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for  
        authentication. Before running this example, set environment variables  
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
  
        CreateInstanceByEngineRequest request = new  
        CreateInstanceByEngineRequest();  
    }  
}
```



```
request.withEngine(CreateInstanceByEngineRequest.EngineEnum.fromValue("{engine}"));
CreateInstanceByEngineReq body = new CreateInstanceByEngineReq();
List<TagEntity> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new TagEntity()
        .withKey("aaa")
        .withValue("111")
);
List<CreateInstanceByEngineReq.SaslEnabledMechanismsEnum>
listbodySaslEnabledMechanisms = new ArrayList<>();

listbodySaslEnabledMechanisms.add(CreateInstanceByEngineReq.SaslEnabledMechanismsEnum.fromValue("SCRAM-SHA-512"));
List<String> listbodyAvailableZones = new ArrayList<>();
listbodyAvailableZones.add("*****706d4c1fb0eb72f0*****");
body.withVpcClientPlain(true);
body.withArchType("X86");
body.withTags(listbodyTags);
body.withEnterpriseProjectId("0");

body.withStorageSpecCode(CreateInstanceByEngineReq.StorageSpecCodeEnum.fromValue("dms.physical.storage.ultra.v2"));
body.withEnableAutoTopic(true);
body.withDiskEncryptedKey("*****-b953-4875-a743-*****");
body.withDiskEncryptedEnable(true);

body.withRetentionPolicy(CreateInstanceByEngineReq.RetentionPolicyEnum.fromValue("time_base"));
body.withSaslEnabledMechanisms(listbodySaslEnabledMechanisms);
body.withKafkaSecurityProtocol("SASL_SSL");
body.withSslEnable(true);
body.withPublicipId("*****-88fc-4a8c-86d0-*****-16af-455d-8d54-*****-3d69-4367-95ab-*****");
body.withEnablePublicip(true);
body.withProductId("c6.2u4g.cluster");
body.withAvailableZones(listbodyAvailableZones);
body.withSubnetId("*****-8fbf-4438-ba71-*****");
body.withSecurityGroupId("*****-e073-4aad-991f-*****");
body.withVpcId("*****-9b4a-44c5-a964-*****");
body.withPassword("*****");
body.withAccessUser("*****");
body.withStorageSpace(300);
body.withBrokerNum(3);
body.withEngineVersion("2.7");

body.withEngine(CreateInstanceByEngineReq.EngineEnum.fromValue("kafka"));
body.withDescription("");
body.withName("kafka-test");
request.withBody(body);
try {
    CreateInstanceByEngineResponse response =
client.createInstanceByEngine(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Criação de uma instância anual/mensal do Kafka cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceByEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateInstanceByEngineRequest request = new
        CreateInstanceByEngineRequest();

        request.withEngine(CreateInstanceByEngineRequest.EngineEnum.fromValue("{engine
        }"));
        CreateInstanceByEngineReq body = new CreateInstanceByEngineReq();
        BssParam bssParambody = new BssParam();

        bssParambody.withChargingMode(BssParam.ChargingModeEnum.fromValue("prePaid"))
            .withIsAutoPay(true)
            .withPeriodType(BssParam.PeriodTypeEnum.fromValue("month"))
            .withPeriodNum(1);
        List<TagEntity> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagEntity()
                .withKey("aaa")
                .withValue("111")
        );
        List<CreateInstanceByEngineReq.SaslEnabledMechanismsEnum>
        listbodySaslEnabledMechanisms = new ArrayList<>();

        listbodySaslEnabledMechanisms.add(CreateInstanceByEngineReq.SaslEnabledMechani
        smsEnum.fromValue("SCRAM-SHA-512"));
        List<String> listbodyAvailableZones = new ArrayList<>();
        listbodyAvailableZones.add("*****706d4c1fb0eb72f0*****");
        body.withBssParam(bssParambody);
        body.withVpcClientPlain(true);
        body.withArchType("X86");
        body.withTags(listbodyTags);
        body.withEnterpriseProjectId("0");
```

```
body.withStorageSpecCode(CreateInstanceByEngineReq.StorageSpecCodeEnum.fromValue("dms.physical.storage.ultra.v2"));
    body.withEnableAutoTopic(true);

body.withRetentionPolicy(CreateInstanceByEngineReq.RetentionPolicyEnum.fromValue("time_base"));
    body.withSaslEnabledMechanisms(listbodySaslEnabledMechanisms);
    body.withKafkaSecurityProtocol("SASL_SSL");
    body.withSslEnable(true);
    body.withPublicipId("*****-88fc-4a8c-86d0-
*****-16af-455d-8d54-*****-3d69-4367-95ab-
*****");
    body.withEnablePublicip(true);
    body.withProductId("c6.2u4g.cluster");
    body.withAvailableZones(listbodyAvailableZones);
    body.withSubnetId("*****-8fbf-4438-ba71-*****");
    body.withSecurityGroupId("*****-e073-4aad-991f-*****");
    body.withVpcId("*****-9b4a-44c5-a964-*****");
    body.withPassword("*****");
    body.withAccessUser("*****");
    body.withStorageSpace(300);
    body.withBrokerNum(3);
    body.withEngineVersion("2.7");

body.withEngine(CreateInstanceByEngineReq.EngineEnum.fromValue("kafka"));
    body.withDescription("");
    body.withName("kafka-test1");
    request.withBody(body);
    try {
        CreateInstanceByEngineResponse response =
client.createInstanceByEngine(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- Criação de uma instância do Kafka de pagamento por uso cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateInstanceByEngineRequest()
    request.engine = "{engine}"
    listTagsbody = [
        TagEntity(
            key="aaa",
            value="111"
        )
    ]
    listSaslEnabledMechanismsbody = [
        "SCRAM-SHA-512"
    ]
    listAvailableZonesbody = [
        "*****706d4c1fb0eb72f0*****"
    ]
    request.body = CreateInstanceByEngineReq(
        vpc_client_plain=True,
        arch_type="X86",
        tags=listTagsbody,
        enterprise_project_id="0",
        storage_spec_code="dms.physical.storage.ultra.v2",
        enable_auto_topic=True,
        disk_encrypted_key="*****-b953-4875-a743-*****",
        disk_encrypted_enable=True,
        retention_policy="time_base",
        sasl_enabled_mechanisms=listSaslEnabledMechanismsbody,
        kafka_security_protocol="SASL_SSL",
        ssl_enable=True,
        publicip_id="*****-88fc-4a8c-86d0-
*****-16af-455d-8d54-*****", *****-3d69-4367-95ab-
*****",
        enable_publicip=True,
        product_id="c6.2u4g.cluster",
        available_zones=listAvailableZonesbody,
        subnet_id="*****-8fbf-4438-ba71-*****",
        security_group_id="*****-e073-4aad-991f-*****",
        vpc_id="*****-9b4a-44c5-a964-*****",
        password="*****",
        access_user="*****",
        storage_space=300,
        broker_num=3,
        engine_version="2.7",
        engine="kafka",
        description="",
        name="kafka-test"
    )
    response = client.create_instance_by_engine(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- Criação de uma instância anual/mensal do Kafka cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

```
# coding: utf-8
```

```

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

```

```

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceByEngineRequest()
        request.engine = "{engine}"
        bssParambody = BssParam(
            charging_mode="prePaid",
            is_auto_pay=True,
            period_type="month",
            period_num=1
        )
        listTagsbody = [
            TagEntity(
                key="aaa",
                value="111"
            )
        ]
        listSaslEnabledMechanismsbody = [
            "SCRAM-SHA-512"
        ]
        listAvailableZonesbody = [
            "*****706d4c1fb0eb72f0*****"
        ]
        request.body = CreateInstanceByEngineReq(
            bss_param=bssParambody,
            vpc_client_plain=True,
            arch_type="X86",
            tags=listTagsbody,
            enterprise_project_id="0",
            storage_spec_code="dms.physical.storage.ultra.v2",
            enable_auto_topic=True,
            retention_policy="time_base",
            sasl_enabled_mechanisms=listSaslEnabledMechanismsbody,
            kafka_security_protocol="SASL_SSL",
            ssl_enable=True,
            publicip_id="*****-88fc-4a8c-86d0-
*****-16af-455d-8d54-*****", *****-3d69-4367-95ab-
*****",
            enable_publicip=True,
            product_id="c6.2u4g.cluster",
            available_zones=listAvailableZonesbody,
            subnet_id="*****-8fbf-4438-ba71-*****",
            security_group_id="*****-e073-4aad-991f-*****",
            vpc_id="*****-9b4a-44c5-a964-*****",
            password="*****",
            access_user="*****",
            storage_space=300,
            broker_num=3,
            engine_version="2.7",
            engine="kafka",
            description="",

```

```
        name="kafka-test1"
    )
    response = client.create_instance_by_engine(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Criação de uma instância do Kafka de pagamento por uso cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceByEngineRequest{}
    request.Engine = model.GetCreateInstanceByEngineRequestEngineEnum().ENGINE
    keyTags := "aaa"
    valueTags := "111"
    var listTagsbody = []model.TagEntity{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    }
    var listSaslEnabledMechanismsbody =
    []model.CreateInstanceByEngineReqSaslEnabledMechanisms{
model.GetCreateInstanceByEngineReqSaslEnabledMechanismsEnum().SCRAM_SHA_512,
    }
    var listAvailableZonesbody = []string{
        "*****706d4c1fb0eb72f0*****",
    }
    vpcClientPlainCreateInstanceByEngineReq := true
    archTypeCreateInstanceByEngineReq := "x86"
```

```

enterpriseProjectIdCreateInstanceByEngineReq:= "0"
enableAutoTopicCreateInstanceByEngineReq:= true
diskEncryptedKeyCreateInstanceByEngineReq:= "*****-b953-4875-a743-
*****"
diskEncryptedEnableCreateInstanceByEngineReq:= true
retentionPolicyCreateInstanceByEngineReq:=
model.GetCreateInstanceByEngineReqRetentionPolicyEnum().TIME_BASE
kafkaSecurityProtocolCreateInstanceByEngineReq:= "SASL_SSL"
sslEnableCreateInstanceByEngineReq:= true
publicipIdCreateInstanceByEngineReq:= "*****-88fc-4a8c-86d0-
*****,*****-16af-455d-8d54-*****,*****-3d69-4367-95ab-
*****"
enablePublicipCreateInstanceByEngineReq:= true
passwordCreateInstanceByEngineReq:= "*****"
accessUserCreateInstanceByEngineReq:= "*****"
descriptionCreateInstanceByEngineReq:= ""
request.Body = &model.CreateInstanceByEngineReq{
    VpcClientPlain: &vpcClientPlainCreateInstanceByEngineReq,
    ArchType: &archTypeCreateInstanceByEngineReq,
    Tags: &listTagsbody,
    EnterpriseProjectId: &enterpriseProjectIdCreateInstanceByEngineReq,
    StorageSpecCode:
model.GetCreateInstanceByEngineReqStorageSpecCodeEnum().DMS_PHYSICAL_STORAGE_U
LTRA,
    EnableAutoTopic: &enableAutoTopicCreateInstanceByEngineReq,
    DiskEncryptedKey: &diskEncryptedKeyCreateInstanceByEngineReq,
    DiskEncryptedEnable: &diskEncryptedEnableCreateInstanceByEngineReq,
    RetentionPolicy: &retentionPolicyCreateInstanceByEngineReq,
    SaslEnabledMechanisms: &listSaslEnabledMechanismsbody,
    KafkaSecurityProtocol:
&kafkaSecurityProtocolCreateInstanceByEngineReq,
    SslEnable: &sslEnableCreateInstanceByEngineReq,
    PublicipId: &publicipIdCreateInstanceByEngineReq,
    EnablePublicip: &enablePublicipCreateInstanceByEngineReq,
    ProductId: "c6.2u4g.cluster",
    AvailableZones: listAvailableZonesbody,
    SubnetId: "*****-8fbf-4438-ba71-*****",
    SecurityGroupId: "*****-e073-4aad-991f-*****",
    VpcId: "*****-9b4a-44c5-a964-*****",
    Password: &passwordCreateInstanceByEngineReq,
    AccessUser: &accessUserCreateInstanceByEngineReq,
    StorageSpace: int32(300),
    BrokerNum: int32(3),
    EngineVersion: "2.7",
    Engine: model.GetCreateInstanceByEngineReqEngineEnum().KAFKA,
    Description: &descriptionCreateInstanceByEngineReq,
    Name: "kafka-test",
}
response, err := client.CreateInstanceByEngine(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

- Criação de uma instância anual/mensal do Kafka cuja versão é 2.7, as especificações são 2 vCPUs | 4 GB x 3 e o espaço de armazenamento é de 300 GB

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
region"
)

func main() {

```

```
// The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateInstanceByEngineRequest{}
request.Engine = model.GetCreateInstanceByEngineRequestEngineEnum().ENGINE
chargingModeBssParam:= model.GetBssParamChargingModeEnum().PRE_PAID
isAutoPayBssParam:= true
periodTypeBssParam:= model.GetBssParamPeriodTypeEnum().MONTH
periodNumBssParam:= int32(1)
bssParambody := &model.BssParam{
    ChargingMode: &chargingModeBssParam,
    IsAutoPay: &isAutoPayBssParam,
    PeriodType: &periodTypeBssParam,
    PeriodNum: &periodNumBssParam,
}
keyTags:= "aaa"
valueTags:= "111"
var listTagsbody = []model.TagEntity{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
}
var listSaslEnabledMechanismsbody =
[]model.CreateInstanceByEngineReqSaslEnabledMechanisms{
model.GetCreateInstanceByEngineReqSaslEnabledMechanismsEnum().SCRAM_SHA_512,
}
var listAvailableZonesbody = []string{
    "*****706d4c1fb0eb72f0*****",
}
vpcClientPlainCreateInstanceByEngineReq:= true
archTypeCreateInstanceByEngineReq:= "X86"
enterpriseProjectIdCreateInstanceByEngineReq:= "0"
enableAutoTopicCreateInstanceByEngineReq:= true
retentionPolicyCreateInstanceByEngineReq:=
model.GetCreateInstanceByEngineReqRetentionPolicyEnum().TIME_BASE
kafkaSecurityProtocolCreateInstanceByEngineReq:= "SASL_SSL"
sslEnableCreateInstanceByEngineReq:= true
publicipIdCreateInstanceByEngineReq:= "*****-88fc-4a8c-86d0-
*****-16af-455d-8d54-*****", *****-3d69-4367-95ab-
*****"
enablePublicipCreateInstanceByEngineReq:= true
passwordCreateInstanceByEngineReq:= "*****"
accessUserCreateInstanceByEngineReq:= "*****"
descriptionCreateInstanceByEngineReq:= ""
request.Body = &model.CreateInstanceByEngineReq{
    BssParam: bssParambody,
    VpcClientPlain: &vpcClientPlainCreateInstanceByEngineReq,
```



```

        ArchType: &archTypeCreateInstanceByEngineReq,
        Tags: &listTagsbody,
        EnterpriseProjectId: &enterpriseProjectIdCreateInstanceByEngineReq,
        StorageSpecCode:
model.GetCreateInstanceByEngineReqStorageSpecCodeEnum().DMS_PHYSICAL_STORAGE_U
LTRA,
        EnableAutoTopic: &enableAutoTopicCreateInstanceByEngineReq,
        RetentionPolicy: &retentionPolicyCreateInstanceByEngineReq,
        SaslEnabledMechanisms: &listSaslEnabledMechanismsbody,
        KafkaSecurityProtocol:
&kafkaSecurityProtocolCreateInstanceByEngineReq,
        SslEnable: &sslEnableCreateInstanceByEngineReq,
        PublicipId: &publicipIdCreateInstanceByEngineReq,
        EnablePublicip: &enablePublicipCreateInstanceByEngineReq,
        ProductId: "c6.2u4g.cluster",
        AvailableZones: listAvailableZonesbody,
        SubnetId: "*****-8fbf-4438-ba71-*****",
        SecurityGroupId: "*****-e073-4aad-991f-*****",
        VpcId: "*****-9b4a-44c5-a964-*****",
        Password: &passwordCreateInstanceByEngineReq,
        AccessUser: &accessUserCreateInstanceByEngineReq,
        StorageSpace: int32(300),
        BrokerNum: int32(3),
        EngineVersion: "2.7",
        Engine: model.GetCreateInstanceByEngineReqEngineEnum().KAFKA,
        Description: &descriptionCreateInstanceByEngineReq,
        Name: "kafka-test1",
    }
    response, err := client.CreateInstanceByEngine(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Instância criada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.2 Listagem de todas as instâncias

Função

Essa API é usada para consultar as instâncias de uma conta pelas condições especificadas.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances

Tabela 5-6 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.

Tabela 5-7 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens. Valor: kafka . Padrão: kafka
name	Não	String	Nome da instância.
instance_id	Não	String	ID da instância.
status	Não	String	Status da instância. Para obter detalhes, consulte Status da instância.
include_failure	Não	String	Indica se deve ser retornado o número de instâncias que não foram criadas. Se o valor for true , será retornado o número de instâncias que não foram criadas. Se o valor for false , o número não será retornado.
exact_match_name	Não	String	Se deve procurar a instância que corresponde precisamente a um nome de instância especificado. O valor padrão é <i>false*</i> , indicando que uma pesquisa difusa é executada com base em um nome de instância especificado. Se o valor for true , a instância que corresponde precisamente a um nome de instância especificado será consultada.

Parâmetro	Obrigatório	Tipo	Descrição
enterprise_project_id	Não	String	ID do projeto empresarial.
offset	Não	String	Deslocamento, que é a posição onde a consulta começa. O valor deve ser maior ou igual a 0.
limit	Não	String	Número máximo de instâncias retornadas na consulta atual. O valor padrão é 10 . O valor varia de 1 a 50.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-8 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
instances	Array of ShowInstanceResp objects	Lista de instâncias.
instance_num	Integer	Número de instâncias.

Tabela 5-9 ShowInstanceResp

Parâmetro	Tipo	Descrição
name	String	Nome da instância.
engine	String	Mecanismo de mensagens.
engine_version	String	Versão.
description	String	Descrição da instância
specification	String	Especificações de instância.
storage_space	Integer	Espaço de armazenamento de mensagens em GB.
partition_num	String	Número de partições em uma instância de Kafka.
used_storage_space	Integer	Espaço de armazenamento de mensagens usado em GB.

Parâmetro	Tipo	Descrição
connect_addresses	String	Endereço IP de uma instância.
port	Integer	Porta de uma instância.
status	String	Status da instância. Para obter detalhes, consulte Status da instância.
instance_id	String	ID da instância.
resource_spec_code	String	Especificações de recursos. <ul style="list-style-type: none">● dms.instance.kafka.cluster.c3.mini: instância do Kafka com largura de banda de 100 MB/s● dms.instance.kafka.cluster.c3.small.2: instância do Kafka com largura de banda de 300 MB/s● dms.instance.kafka.cluster.c3.middle.2: instância do Kafka com largura de banda de 600 MB/s● dms.instance.kafka.cluster.c3.high.2: instância do Kafka com largura de banda de 1200 MB/s
charging_mode	Integer	Modo de cobrança. Opções: 1 : pagamento por uso; 0 : anual/mensal.
vpc_id	String	ID de VPC
vpc_name	String	Nome da VPC.
created_at	String	Hora em que a instância foi criada. A hora está no formato de carimbo de data/hora, ou seja, o deslocamento em milissegundos de 01/01/1970 00:00:00 UTC até a hora especificada.
subnet_name	String	Nome da sub-rede.
subnet_cidr	String	Bloco CIDR da sub-rede.
user_id	String	ID do usuário.
user_name	String	Nome de usuário.
access_user	String	Nome de usuário para acessar a instância.
order_id	String	ID do pedido. Este parâmetro tem um valor apenas quando o modo de cobrança é anual/mensal.
maintain_begin	String	Hora em que a janela de tempo de manutenção começa. O formato é HH:mm:ss.
maintain_end	String	Hora em que a janela de tempo de manutenção termina. O formato é HH:mm:ss.

Parâmetro	Tipo	Descrição
enable_publicip	Boolean	Se o acesso público está ativado para a instância. <ul style="list-style-type: none"> ● true: ativado ● false: desativado
management_connect_address	String	Endereço de conexão do Kafka Manager da instância do Kafka.
ssl_enable	Boolean	Se a autenticação de segurança está ativada. <ul style="list-style-type: none"> ● true: ativada ● false: desativada
broker_ssl_enable	Boolean	Indica se deve ser ativada a transmissão de réplicas criptografadas entre corretores. <ul style="list-style-type: none"> ● true: ativada ● false: desativada
kafka_security_protocol	String	Protocolo de segurança a ser usado após a ativação do SASL. <ul style="list-style-type: none"> ● SASL_SSL: os dados são criptografados com certificados SSL para transmissão de alta segurança. ● SASL_PLAINTEXT: os dados são transmitidos em texto não criptografado com autenticação de nome de usuário e senha. Este protocolo usa o mecanismo SCRAM-SHA-512 e oferece alto desempenho.
sasl_enabled_mechanisms	Array of strings	Mecanismo de autenticação usado após a ativação do SASL. <ul style="list-style-type: none"> ● PLAIN: verificação simples de nome de usuário e senha. ● SCRAM-SHA-512: verificação da credencial do usuário, que é mais segura do que PLAIN.
ssl_two_way_enable	Boolean	Indica se a autenticação bidirecional deve ser ativada.
cert_replaced	Boolean	Se o certificado pode ser substituído.
public_management_connect_address	String	Endereço para acessar o Kafka Manager em redes públicas.
enterprise_project_id	String	ID do projeto empresarial.

Parâmetro	Tipo	Descrição
is_logical_volume	Boolean	Se a instância é uma nova instância. Esse parâmetro é usado para distinguir instâncias anteriores de instâncias novas durante a expansão da capacidade da instância. <ul style="list-style-type: none">● true: nova instância, que permite a expansão dinâmica da capacidade do disco sem reiniciar a instância.● false: instância anterior.
extend_times	Integer	Número de tempos de expansão do disco. Se o valor exceder 20, a expansão do disco não será mais permitida.
enable_auto_topic	Boolean	Se a criação automática de tópicos está ativada. <ul style="list-style-type: none">● true: ativada● false: desativada
type	String	Tipos de instâncias. O valor pode ser cluster .
product_id	String	ID do produto.
security_group_id	String	ID do grupo de segurança.
security_group_name	String	Nome do grupo de segurança.
subnet_id	String	ID da sub-rede.
available_zones	Array of strings	AZ ao qual os corretores de instância pertencem. O ID da AZ é retornado.
available_zone_names	Array of strings	Nome da AZ à qual o nó da instância pertence. O nome da AZ é retornado.
total_storage_space	Integer	Espaço de armazenamento de mensagens em GB.
public_connect_address	String	Endereço de acesso público da instância. Esse parâmetro está disponível somente quando o acesso público está ativado para a instância.
storage_resource_id	String	ID do recurso de armazenamento.
storage_spec_code	String	Especificações de I/O.
service_type	String	Tipo de serviço.
storage_type	String	Classe de armazenamento.
retention_policy	String	Política de retenção de mensagens.

Parâmetro	Tipo	Descrição
kafka_public_status	String	Se o acesso público está ativado para o Kafka.
public_bandwidth	Integer	Largura de banda de acesso à rede pública.
kafka_manager_enable	Boolean	Se o Kafka Manager está ativado.
kafka_manager_user	String	Nome de usuário para efetuar logon no Kafka Manager.
enable_log_collection	Boolean	Se a coleta de logs está ativada.
new_auth_cert	Boolean	Indica se deve ativar um novo certificado.
cross_vpc_info	String	Informações de acesso entre VPCs.
ipv6_enable	Boolean	Se o IPv6 está ativado.
ipv6_connect_addresses	Array of strings	Endereço de conexão IPv6.
connector_enable	Boolean	Se o despejo está ativado. O despejo não é suportado para o novo tipo de especificação.
connector_node_num	Integer	Número de conectores.
connector_id	String	ID da tarefa de despejo.
rest_enable	Boolean	Se o Kafka REST está ativado.
rest_connect_address	String	Endereço de conexão do Kafka REST.
public_bandwidth	Integer	Largura de banda de acesso à rede pública. Versão a ser excluída.
message_query_inst_enable	Boolean	Se a consulta de mensagem está ativada.
vpc_client_plain	Boolean	Se o acesso de texto não criptografado dentro da VPC está ativado.
support_features	String	Lista de recursos suportados pela instância de Kafka.
trace_enable	Boolean	Se o rastreamento de mensagens está ativado.
agent_enable	Boolean	Indica se o proxy está ativado.
pod_connect_address	String	Endereço de conexão no lado do locatário.
disk_encrypted	Boolean	Se a criptografia de disco está ativada.

Parâmetro	Tipo	Descrição
disk_encrypted_key	String	Key de criptografia de disco. Se a criptografia de disco não estiver ativada, esse parâmetro será deixado em branco.
kafka_private_connect_address	String	Endereço de conexão privada de uma instância do Kafka.
ces_version	String	Versão do Cloud Eye.
public_access_enabled	String	Hora em que o acesso público foi ativado para uma instância. O valor pode ser true , actived , closed ou false .
node_num	Integer	Quantidade de nó.
enable_acl	Boolean	Indica se o controle de acesso está ativado.
new_spec_billing_enable	Boolean	Se a cobrança com base em novas especificações está ativada.
broker_num	Integer	Quantidade do corretor.
tags	Array of TagEntity objects	Lista de tags.
dr_enable	Boolean	Indica se a DR está ativada.

Tabela 5-10 TagEntity

Parâmetro	Tipo	Descrição
key	String	Chave de tag, que: <ul style="list-style-type: none"> ● Deve ser especificada. ● Deve ser exclusiva para a mesma instância. ● Pode conter de 1 a 128 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code> ● Não pode começar ou terminar com um espaço.
value	String	Valor da tag. <ul style="list-style-type: none"> ● Pode conter de 0 a 255 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code>

Exemplo de solicitações

Consulta da lista de instâncias


```
GET https://{endpoint}/v2/{project_id}/instances
```

Exemplo de respostas

Código de status: 200

Lista de instâncias consultada.

- A lista de instâncias é consultada. Um exemplo de resultado retornado é o seguinte.

```
{
  "instances" : [ {
    "name" : "kafka-2085975099",
    "engine" : "kafka",
    "port" : 9092,
    "status" : "RUNNING",
    "type" : "cluster",
    "specification" : "100MB",
    "engine_version" : "1.1.0",
    "connect_address" : "192.168.0.100,192.168.0.61,192.168.0.72",
    "instance_id" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "resource_spec_code" : "dms.instance.kafka.cluster.c3.mini",
    "charging_mode" : 1,
    "vpc_id" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "vpc_name" : "dms-test",
    "created_at" : "1585618587087",
    "product_id" : "00300-30308-0--0",
    "security_group_id" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "security_group_name" : "Sys-default",
    "subnet_id" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "available_zones" : [ "38b0f7a602344246bcb0da47b5d548e7" ],
    "available_zone_names" : [ "AZ1" ],
    "user_id" : "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "user_name" : "paas_dms",
    "kafka_manager_enable" : false,
    "kafka_manager_user" : "root",
    "maintain_begin" : "02:00:00",
    "maintain_end" : "06:00:00",
    "enable_log_collection" : false,
    "new_auth_cert" : false,
    "storage_space" : 492,
    "total_storage_space" : 600,
    "used_storage_space" : 25,
    "partition_num" : "300",
    "enable_publicip" : false,
    "ssl_enable" : false,
    "broker_ssl_enable" : false,
    "cert_replaced" : false,
    "management_connect_address" : "https://192.168.0.100:9999",
    "cross_vpc_info" : "{\\"192.168.0.61\\":{\\"advertised_ip
\\":\\"192.168.0.61\\",\\"port\\":9011,\\"port_id\\":\\"xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx\\"},\\"192.168.0.72\\":{\\"advertised_ip\\":\\"192.168.0.72\\",\\"port
\\":9011,\\"port_id\\":\\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
\\"},\\"192.168.0.100\\":{\\"advertised_ip\\":\\"192.168.0.100\\",\\"port
\\":9011,\\"port_id\\":\\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\\"}}",
    "storage_resource_id" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "storage_spec_code" : "dms.physical.storage.ultra",
    "service_type" : "advanced",
    "storage_type" : "hec",
    "enterprise_project_id" : "0",
    "is_logical_volume" : true,
    "extend_times" : 0,
    "retention_policy" : "produce_reject",
    "ipv6_enable" : false,
    "ipv6_connect_addresses" : [ ],
    "connector_enable" : false,
    "connector_node_num" : 0,
    "connector_id" : "",
    "rest_enable" : false,
  }
  ]
}
```

```
"rest_connect_address" : "",
"kafka_public_status" : "closed",
"public_bandwidth" : 0,
"message_query_inst_enable" : true,
"vpc_client_plain" : false,
"support_features" :
"kafka.new.pod.port,feature.physerver.kafka.topic.modify,feature.physerver.kaf
ka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.p
hyserver.kafka.publlic.dynamic,roma_app_enable,features.log.collection,auto_top
ic_switch,feature.physerver.kafka.user.manager",
"trace_enable" : false,
"agent_enable" : false,
"pod_connect_address" :
"100.86.75.15:9080,100.86.142.77:9080,100.86.250.167:9080",
"disk_encrypted" : false,
"kafka_private_connect_address" :
"192.168.0.61:9092,192.168.0.100:9092,192.168.0.72:9092",
"enable_auto_topic" : false,
"new_spec_billing_enable" : false,
"ces_version" : "linux"
} ],
"instance_num" : 1
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInstancesRequest request = new ListInstancesRequest();
        try {
            ListInstancesResponse response = client.listInstances(request);
        }
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstancesRequest()
        response = client.list_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
```

```
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListInstancesRequest{}
response, err := client.ListInstances(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Lista de instâncias consultada.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.3 Consulta de uma instância

Função

Essa API é usada para consultar os detalhes sobre uma instância especificada.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}

Tabela 5-11 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-12 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
name	String	Nome da instância.
engine	String	Mecanismo de mensagens.
engine_version	String	Versão.
description	String	Descrição da instância
specification	String	Especificações de instância.
storage_space	Integer	Espaço de armazenamento de mensagens em GB.
partition_num	String	Número de partições em uma instância de Kafka.
used_storage_space	Integer	Espaço de armazenamento de mensagens usado em GB.
connect_addresses	String	Endereço IP de uma instância.
port	Integer	Porta de uma instância.
status	String	Status da instância. Para obter detalhes, consulte Status da instância.
instance_id	String	ID da instância.

Parâmetro	Tipo	Descrição
resource_spec_code	String	Especificações de recursos. <ul style="list-style-type: none">● dms.instance.kafka.cluster.c3.mini: instância do Kafka com largura de banda de 100 MB/s● dms.instance.kafka.cluster.c3.small.2: instância do Kafka com largura de banda de 300 MB/s● dms.instance.kafka.cluster.c3.middle.2: instância do Kafka com largura de banda de 600 MB/s● dms.instance.kafka.cluster.c3.high.2: instância do Kafka com largura de banda de 1200 MB/s
charging_mode	Integer	Modo de cobrança. Opções: 1: pagamento por uso; 0: anual/mensal.
vpc_id	String	ID de VPC
vpc_name	String	Nome da VPC.
created_at	String	Hora em que a instância foi criada. A hora está no formato de carimbo de data/hora, ou seja, o deslocamento em milissegundos de 01/01/1970 00:00:00 UTC até a hora especificada.
subnet_name	String	Nome da sub-rede.
subnet_cidr	String	Bloco CIDR da sub-rede.
user_id	String	ID do usuário.
user_name	String	Nome de usuário.
access_user	String	Nome de usuário para acessar a instância.
order_id	String	ID do pedido. Este parâmetro tem um valor apenas quando o modo de cobrança é anual/mensal.
maintain_begin	String	Hora em que a janela de tempo de manutenção começa. O formato é HH:mm:ss.
maintain_end	String	Hora em que a janela de tempo de manutenção termina. O formato é HH:mm:ss.
enable_publicip	Boolean	Se o acesso público está ativado para a instância. <ul style="list-style-type: none">● true: ativado● false: desativado
management_connect_address	String	Endereço de conexão do Kafka Manager da instância do Kafka.
ssl_enable	Boolean	Se a autenticação de segurança está ativada. <ul style="list-style-type: none">● true: ativada● false: desativada

Parâmetro	Tipo	Descrição
broker_ssl_enable	Boolean	Indica se deve ser ativada a transmissão de réplicas criptografadas entre corretores. <ul style="list-style-type: none"> ● true: ativada ● false: desativada
kafka_security_protocol	String	Protocolo de segurança a ser usado após a ativação do SASL. <ul style="list-style-type: none"> ● SASL_SSL: os dados são criptografados com certificados SSL para transmissão de alta segurança. ● SASL_PLAINTEXT: os dados são transmitidos em texto não criptografado com autenticação de nome de usuário e senha. Este protocolo usa o mecanismo SCRAM-SHA-512 e oferece alto desempenho.
sasl_enabled_mechanisms	Array of strings	Mecanismo de autenticação usado após a ativação do SASL. <ul style="list-style-type: none"> ● PLAIN: verificação simples de nome de usuário e senha. ● SCRAM-SHA-512: verificação da credencial do usuário, que é mais segura do que PLAIN.
ssl_two_way_enable	Boolean	Indica se a autenticação bidirecional deve ser ativada.
cert_replaced	Boolean	Se o certificado pode ser substituído.
public_management_connect_address	String	Endereço para acessar o Kafka Manager em redes públicas.
enterprise_project_id	String	ID do projeto empresarial.
is_logical_volume	Boolean	Se a instância é uma nova instância. Esse parâmetro é usado para distinguir instâncias anteriores de instâncias novas durante a expansão da capacidade da instância. <ul style="list-style-type: none"> ● true: nova instância, que permite a expansão dinâmica da capacidade do disco sem reiniciar a instância. ● false: instância anterior.
extend_times	Integer	Número de tempos de expansão do disco. Se o valor exceder 20, a expansão do disco não será mais permitida.

Parâmetro	Tipo	Descrição
enable_auto_topic	Boolean	Se a criação automática de tópicos está ativada. <ul style="list-style-type: none"> ● true: ativada ● false: desativada
type	String	Tipos de instâncias. O valor pode ser cluster .
product_id	String	ID do produto.
security_group_id	String	ID do grupo de segurança.
security_group_name	String	Nome do grupo de segurança.
subnet_id	String	ID da sub-rede.
available_zones	Array of strings	AZ ao qual os corretores de instância pertencem. O ID da AZ é retornado.
available_zone_names	Array of strings	Nome da AZ à qual o nó da instância pertence. O nome da AZ é retornado.
total_storage_space	Integer	Espaço de armazenamento de mensagens em GB.
public_connect_address	String	Endereço de acesso público da instância. Esse parâmetro está disponível somente quando o acesso público está ativado para a instância.
storage_resource_id	String	ID do recurso de armazenamento.
storage_spec_code	String	Especificações de I/O.
service_type	String	Tipo de serviço.
storage_type	String	Classe de armazenamento.
retention_policy	String	Política de retenção de mensagens.
kafka_public_status	String	Se o acesso público está ativado para o Kafka.
public_bandwidth	Integer	Largura de banda de acesso à rede pública.
kafka_manager_enable	Boolean	Se o Kafka Manager está ativado.
kafka_manager_user	String	Nome de usuário para efetuar logon no Kafka Manager.
enable_log_collection	Boolean	Se a coleta de logs está ativada.

Parâmetro	Tipo	Descrição
new_auth_cert	Boolean	Indica se deve ativar um novo certificado.
cross_vpc_info	String	Informações de acesso entre VPCs.
ipv6_enable	Boolean	Se o IPv6 está ativado.
ipv6_connect_addresses	Array of strings	Endereço de conexão IPv6.
connector_enable	Boolean	Se o despejo está ativado. O despejo não é suportado para o novo tipo de especificação.
connector_node_num	Integer	Número de conectores.
connector_id	String	ID da tarefa de despejo.
rest_enable	Boolean	Se o Kafka REST está ativado.
rest_connect_address	String	Endereço de conexão do Kafka REST.
public_bandwidth	Integer	Largura de banda de acesso à rede pública. Versão a ser excluída.
message_query_inst_enable	Boolean	Se a consulta de mensagem está ativada.
vpc_client_plain	Boolean	Se o acesso de texto não criptografado dentro da VPC está ativado.
support_features	String	Lista de recursos suportados pela instância de Kafka.
trace_enable	Boolean	Se o rastreamento de mensagens está ativado.
agent_enable	Boolean	Indica se o proxy está ativado.
pod_connect_address	String	Endereço de conexão no lado do locatário.
disk_encrypted	Boolean	Se a criptografia de disco está ativada.
disk_encrypted_key	String	Key de criptografia de disco. Se a criptografia de disco não estiver ativada, esse parâmetro será deixado em branco.
kafka_private_connect_address	String	Endereço de conexão privada de uma instância do Kafka.
ces_version	String	Versão do Cloud Eye.
public_access_enabled	String	Hora em que o acesso público foi ativado para uma instância. O valor pode ser true , actived , closed ou false .
node_num	Integer	Quantidade de nó.

Parâmetro	Tipo	Descrição
enable_acl	Boolean	Indica se o controle de acesso está ativado.
new_spec_billing_enable	Boolean	Se a cobrança com base em novas especificações está ativada.
broker_num	Integer	Quantidade do corretor.
tags	Array of TagEntity objects	Lista de tags.
dr_enable	Boolean	Indica se a DR está ativada.

Tabela 5-13 TagEntity

Parâmetro	Tipo	Descrição
key	String	Chave de tag, que: <ul style="list-style-type: none"> ● Deve ser especificada. ● Deve ser exclusiva para a mesma instância. ● Pode conter de 1 a 128 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code> ● Não pode começar ou terminar com um espaço.
value	String	Valor da tag. <ul style="list-style-type: none"> ● Pode conter de 0 a 255 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code>

Exemplo de solicitações

Consulta de uma instância

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

Exemplo de respostas

Código de status: 200

Instância especificada consultada.

- A instância é consultada. Um exemplo de resultado retornado é o seguinte.

```
{
  "name" : "kafka-2085975099",
  "engine" : "kafka",
  "port" : 9092,
  "status" : "RUNNING",
  "type" : "cluster",
  "specification" : "100MB",
```



```
"new_spec_billing_enable" : false,  
"ces_version" : "linux"  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ShowInstanceSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
plaintext, which has great security risks. It is recommended that the AK and SK  
be stored in ciphertext in configuration files or environment variables and  
decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for  
authentication. Before running this example, set environment variables  
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowInstanceRequest request = new ShowInstanceRequest();  
        request.withInstanceId("{instance_id}");  
        try {  
            ShowInstanceResponse response = client.showInstance(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```

        Build()

        request := &model.ShowInstanceRequest{}
        request.InstanceId = "{instance_id}"
        response, err := client.ShowInstance(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Instância especificada consultada.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.4 Exclusão de uma instância

Função

Essa API é usada para excluir uma instância para liberar todos os recursos ocupados por ela.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

DELETE /v2/{project_id}/instances/{instance_id}

Tabela 5-14 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Exclusão de uma instância

```
DELETE https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        DeleteInstanceRequest request = new DeleteInstanceRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeleteInstanceResponse response = client.deleteInstance(request);
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteInstanceRequest()
        request.instance_id = "{instance_id}"
        response = client.delete_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
```



```
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteInstanceRequest{
    request.InstanceId = "{instance_id}"
}
response, err := client.DeleteInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A instância especificada foi excluída com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.5 Modificação das informações da instância

Função

Essa API é usada para modificar as informações da instância.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}

Tabela 5-15 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-16 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
name	Não	String	Nome da instância. Um nome de instância começa com uma letra, consiste de 4 a 64 caracteres e pode conter apenas letras, dígitos, sublinhados (_) e hifens (-).
description	Não	String	Descrição de uma instância. A descrição suporta até 1024 caracteres. NOTA A barra invertida (\) e as aspas (") são caracteres especiais para mensagens JSON. Ao usar esses caracteres em um valor de parâmetro, adicione o caractere de escape (\) antes dos caracteres, por exemplo, \ e \".
maintain_begin	Não	String	Hora em que a janela de manutenção é iniciada. O formato é HH:mm:ss.
maintain_end	Não	String	Hora em que a janela de manutenção termina. O formato é HH:mm:ss.
security_group_id	Não	String	ID do grupo de segurança. Para obtê-lo, faça logon no console da VPC e visualize o ID do grupo de segurança na página de detalhes do grupo de segurança.

Parâmetro	Obrigatório	Tipo	Descrição
retention_policy	Não	String	Política de limite de capacidade. Opções: <ul style="list-style-type: none"> ● produce_reject: novas mensagens não podem ser criadas. ● time_base: as mensagens mais antigas são excluídas.
enterprise_project_id	Não	String	Projeto empresarial

Parâmetros de resposta

Nenhum

Exemplo de solicitações

- Modificação do nome e da descrição de uma instância.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}
{
  "name" : "kafka001",
  "description" : "kafka description"
}
```

- Modificação do nome, da descrição e da janela de tempo de manutenção de uma instância.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}
{
  "name" : "dms002",
  "description" : "instance description",
  "maintain_begin" : "02:00:00",
  "maintain_end" : "06:00:00"
}
```

- Alteração da política de limite de capacidade.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}
{
  "retention_policy" : "time_base"
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- Modificação do nome e da descrição de uma instância.

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceReq body = new UpdateInstanceReq();
        body.withDescription("kafka description");
        body.withName("kafka001");
        request.withBody(body);
        try {
            UpdateInstanceResponse response = client.updateInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

- **Modificação do nome, da descrição e da janela de tempo de manutenção de uma instância.**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceReq body = new UpdateInstanceReq();
        body.withMaintainEnd("06:00:00");
        body.withMaintainBegin("02:00:00");
        body.withDescription("instance description");
        body.withName("dms002");
        request.withBody(body);
        try {
            UpdateInstanceResponse response = client.updateInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- **Alteração da política de limite de capacidade.**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
```

```

CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

UpdateInstanceRequest request = new UpdateInstanceRequest();
request.withInstanceId("{instance_id}");
UpdateInstanceReq body = new UpdateInstanceReq();

body.withRetentionPolicy(UpdateInstanceReq.RetentionPolicyEnum.fromValue("time_base"));
request.withBody(body);
try {
    UpdateInstanceResponse response = client.updateInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

Python

- Modificação do nome e da descrição de uma instância.

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceRequest()

```

```

request.instance_id = "{instance_id}"
request.body = UpdateInstanceReq(
    description="kafka description",
    name="kafka001"
)
response = client.update_instance(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- **Modificação do nome, da descrição e da janela de tempo de manutenção de uma instância.**

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateInstanceReq(
            maintain_end="06:00:00",
            maintain_begin="02:00:00",
            description="instance description",
            name="dms002"
        )
        response = client.update_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

- **Alteração da política de limite de capacidade.**

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables

```

```
and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateInstanceRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdateInstanceReq(
        retention_policy="time_base"
    )
    response = client.update_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Modificação do nome e da descrição de uma instância.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{}
    request.InstanceId = "{instance_id}"
```



```
descriptionUpdateInstanceReq:= "kafka description"
nameUpdateInstanceReq:= "kafka001"
request.Body = &model.UpdateInstanceReq{
    Description: &descriptionUpdateInstanceReq,
    Name: &nameUpdateInstanceReq,
}
response, err := client.UpdateInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- **Modificação do nome, da descrição e da janela de tempo de manutenção de uma instância.**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{}
    request.InstanceId = "{instance_id}"
    maintainEndUpdateInstanceReq:= "06:00:00"
    maintainBeginUpdateInstanceReq:= "02:00:00"
    descriptionUpdateInstanceReq:= "instance description"
    nameUpdateInstanceReq:= "dms002"
    request.Body = &model.UpdateInstanceReq{
        MaintainEnd: &maintainEndUpdateInstanceReq,
        MaintainBegin: &maintainBeginUpdateInstanceReq,
        Description: &descriptionUpdateInstanceReq,
        Name: &nameUpdateInstanceReq,
    }
    response, err := client.UpdateInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- **Alteração da política de limite de capacidade.**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and
    // SK be stored in ciphertext in configuration files or environment variables
    // and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{}
    request.InstanceId = "{instance_id}"
    retentionPolicyUpdateInstanceReq :=
    model.GetUpdateInstanceReqRetentionPolicyEnum().TIME_BASE
    request.Body = &model.UpdateInstanceReq{
        RetentionPolicy: &retentionPolicyUpdateInstanceReq,
    }
    response, err := client.UpdateInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	As informações da instância foram modificadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.6 Reinicialização ou exclusão de instâncias em lote

Função

Essa API é usada para reiniciar ou excluir instâncias em lote.

Quando uma instância estiver sendo reiniciada, as solicitações de recuperação e criação de mensagens do cliente serão rejeitadas.

A exclusão de uma instância excluirá os dados na instância sem nenhum backup. Tenha cuidado ao realizar esta operação.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/action

Tabela 5-17 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.

Parâmetros de solicitação

Tabela 5-18 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
instances	Não	Array of strings	Lista dos IDs de instância.
action	Sim	String	Operação a ser executada nas instâncias. O valor pode ser restart ou delete .
all_failure	Não	String	O valor kafka indica que todas as instâncias de Kafka que não forem criadas serão excluídas.

Parâmetros de resposta

Código de status: 200

Tabela 5-19 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
results	Array of results objects	Resultado da modificação da instância.

Tabela 5-20 results

Parâmetro	Tipo	Descrição
result	String	Resultado da operação. <ul style="list-style-type: none"> ● success: a operação foi bem-sucedida. ● failed: a operação falhou.
instance	String	ID da instância.

Exemplo de solicitações

- Reinicialização de instâncias em lotes.

```
POST https://{endpoint}/v2/{project_id}/instances/action
{
  "action" : "restart",
  "instances" : [ "54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-
dbad-4d79-9610-7163e6f8b640" ]
}
```

- Exclusão de instâncias em lotes.

```
POST https://{endpoint}/v2/{project_id}/instances/action
{
  "action" : "delete",
  "instances" : [ "54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-
dbad-4d79-9610-7163e6f8b640" ]
}
```

- Exclusão de todas as instâncias que não foram criadas.

```
POST https://{endpoint}/v2/{project_id}/instances/action
{
  "action" : "delete",
  "all_failure" : "kafka"
}
```

Exemplo de respostas

Código de status: 200

As instâncias são reiniciadas ou excluídas com sucesso.

```
{
  "results" : [ {
    "result" : "success",
    "instance" : "019cacb7-4ff0-4d3c-9f33-f5f7b7fdc0e6"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- Reinicialização de instâncias em lotes.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchRestartOrDeleteInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and
        // SK be stored in ciphertext in configuration files or environment variables
        // and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchRestartOrDeleteInstancesRequest request = new
        BatchRestartOrDeleteInstancesRequest();
        BatchRestartOrDeleteInstanceReq body = new
        BatchRestartOrDeleteInstanceReq();
        List<String> listbodyInstances = new ArrayList<>();
        listbodyInstances.add("54602a9d-5e22-4239-9123-77e350df4a34");
        listbodyInstances.add("7166cdea-dbad-4d79-9610-7163e6f8b640");

        body.withAction(BatchRestartOrDeleteInstanceReq.ActionEnum.fromValue("restart"
        ));
        body.withInstances(listbodyInstances);
        request.withBody(body);
        try {
            BatchRestartOrDeleteInstancesResponse response =
        client.batchRestartOrDeleteInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```

```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- **Exclusão de instâncias em lotes.**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchRestartOrDeleteInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchRestartOrDeleteInstancesRequest request = new
        BatchRestartOrDeleteInstancesRequest();
        BatchRestartOrDeleteInstanceReq body = new
        BatchRestartOrDeleteInstanceReq();
        List<String> listbodyInstances = new ArrayList<>();
        listbodyInstances.add("54602a9d-5e22-4239-9123-77e350df4a34");
        listbodyInstances.add("7166cdea-dbad-4d79-9610-7163e6f8b640");

        body.withAction(BatchRestartOrDeleteInstanceReq.ActionEnum.fromValue("delete")
        );
        body.withInstances(listbodyInstances);
        request.withBody(body);
        try {
            BatchRestartOrDeleteInstancesResponse response =
        client.batchRestartOrDeleteInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- **Exclusão de todas as instâncias que não foram criadas.**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class BatchRestartOrDeleteInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and
        // SK be stored in ciphertext in configuration files or environment variables
        // and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchRestartOrDeleteInstancesRequest request = new
        BatchRestartOrDeleteInstancesRequest();
        BatchRestartOrDeleteInstanceReq body = new
        BatchRestartOrDeleteInstanceReq();

        body.withAllFailure(BatchRestartOrDeleteInstanceReq.AllFailureEnum.fromValue("
        kafka"));

        body.withAction(BatchRestartOrDeleteInstanceReq.ActionEnum.fromValue("delete")
        );

        request.withBody(body);
        try {
            BatchRestartOrDeleteInstancesResponse response =
            client.batchRestartOrDeleteInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Reinicialização de instâncias em lotes.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchRestartOrDeleteInstancesRequest()
        listInstancesbody = [
            "54602a9d-5e22-4239-9123-77e350df4a34",
            "7166cdea-dbad-4d79-9610-7163e6f8b640"
        ]
        request.body = BatchRestartOrDeleteInstanceReq(
            action="restart",
            instances=listInstancesbody
        )
        response = client.batch_restart_or_delete_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Exclusão de instâncias em lotes.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
```



```
.with_credentials(credentials) \  
.with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = BatchRestartOrDeleteInstancesRequest()  
    listInstancesbody = [  
        "54602a9d-5e22-4239-9123-77e350df4a34",  
        "7166cdea-dbad-4d79-9610-7163e6f8b640"  
    ]  
    request.body = BatchRestartOrDeleteInstanceReq(  
        action="delete",  
        instances=listInstancesbody  
    )  
    response = client.batch_restart_or_delete_instances(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

- Exclusão de todas as instâncias que não foram criadas.

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsddkafka.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in  
    plaintext, which has great security risks. It is recommended that the AK and  
    SK be stored in ciphertext in configuration files or environment variables  
    and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for  
    authentication. Before running this example, set environment variables  
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId) \  
  
    client = KafkaClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = BatchRestartOrDeleteInstancesRequest()  
        request.body = BatchRestartOrDeleteInstanceReq(  
            all_failure="kafka",  
            action="delete"  
        )  
        response = client.batch_restart_or_delete_instances(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

- Reinicialização de instâncias em lotes.

```
package main  
  
import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchRestartOrDeleteInstancesRequest{}
    var listInstancesbody = []string{
        "54602a9d-5e22-4239-9123-77e350df4a34",
        "7166cdea-dbad-4d79-9610-7163e6f8b640",
    }
    request.Body = &model.BatchRestartOrDeleteInstanceReq{
        Action: model.GetBatchRestartOrDeleteInstanceReqActionEnum().RESTART,
        Instances: &listInstancesbody,
    }
    response, err := client.BatchRestartOrDeleteInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Exclusão de instâncias em lotes.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
    region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchRestartOrDeleteInstancesRequest{}
var listInstancesbody = []string{
    "54602a9d-5e22-4239-9123-77e350df4a34",
    "7166cdea-dbad-4d79-9610-7163e6f8b640",
}
request.Body = &model.BatchRestartOrDeleteInstanceReq{
    Action: model.GetBatchRestartOrDeleteInstanceReqActionEnum().DELETE,
    Instances: &listInstancesbody,
}
response, err := client.BatchRestartOrDeleteInstances(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Exclusão de todas as instâncias que não foram criadas.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchRestartOrDeleteInstancesRequest{}
    allFailureBatchRestartOrDeleteInstanceReq:=
```

```

model.GetBatchRestartOrDeleteInstanceReqAllFailureEnum().KAFKA
request.Body = &model.BatchRestartOrDeleteInstanceReq{
    AllFailure: &allFailureBatchRestartOrDeleteInstanceReq,
    Action: model.GetBatchRestartOrDeleteInstanceReqActionEnum().DELETE,
}
response, err := client.BatchRestartOrDeleteInstances(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As instâncias são reiniciadas ou excluídas com sucesso.
204	Todas as instâncias do Kafka que não foram criadas são excluídas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.7 Obtenção de configurações de instância

Função

Essa API é usada para obter configurações de instâncias.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/configs

Tabela 5-21 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.

Parâmetro	Obrigatório	Tipo	Descrição
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-22 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
kafka_configs	Array of InstanceConfig objects	Configurações de Kafka.

Tabela 5-23 InstanceConfig

Parâmetro	Tipo	Descrição
name	String	Nome da configuração.
valid_values	String	Valor válido.
default_value	String	Valor padrão.
config_type	String	Tipo de configuração. O valor pode ser static ou dynamic .
value	String	Valor atual.
value_type	String	Tipo de valor.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/configs
```

Exemplo de respostas

Código de status: 200

Configuração obtida.

```
{
  "kafka_configs" : [ {
    "name" : "min.insync.replicas",
    "valid_values" : "1~3",
    "default_value" : "1",
```

```
"config_type" : "dynamic",
"value" : "1",
"value_type" : "integer"
}, {
  "name" : "message.max.bytes",
  "valid_values" : "0~10485760",
  "default_value" : "10485760",
  "config_type" : "dynamic",
  "value" : "10485760",
  "value_type" : "integer"
}, {
  "name" : "auto.create.groups.enable",
  "valid_values" : "true,false",
  "default_value" : "true",
  "config_type" : "dynamic",
  "value" : "true",
  "value_type" : "enum"
}, {
  "name" : "connections.max.idle.ms",
  "valid_values" : "5000~600000",
  "default_value" : "600000",
  "config_type" : "static",
  "value" : "600000",
  "value_type" : "integer"
}, {
  "name" : "log.retention.hours",
  "valid_values" : "1~168",
  "default_value" : "72",
  "config_type" : "static",
  "value" : "72",
  "value_type" : "integer"
}, {
  "name" : "max.connections.per.ip",
  "valid_values" : "100~20000",
  "default_value" : "1000",
  "config_type" : "dynamic",
  "value" : "1000",
  "value_type" : "integer"
}, {
  "name" : "group.max.session.timeout.ms",
  "valid_values" : "6000~1800000",
  "default_value" : "1800000",
  "config_type" : "static",
  "value" : "1800000",
  "value_type" : "integer"
}, {
  "name" : "unclean.leader.election.enable",
  "valid_values" : "true,false",
  "default_value" : "false",
  "config_type" : "dynamic",
  "value" : "false",
  "value_type" : "enum"
}, {
  "name" : "default.replication.factor",
  "valid_values" : "1~3",
  "default_value" : "3",
  "config_type" : "static",
  "value" : "3",
  "value_type" : "integer"
}, {
  "name" : "offsets.retention.minutes",
  "valid_values" : "1440~30240",
  "default_value" : "20160",
  "config_type" : "dynamic",
  "value" : "20160",
  "value_type" : "integer"
}, {
  "name" : "num.partitions",
  "valid_values" : "1~200",
```

```
    "default_value" : "3",
    "config_type" : "static",
    "value" : "3",
    "value_type" : "integer"
  }, {
    "name" : "group.min.session.timeout.ms",
    "valid_values" : "6000~300000",
    "default_value" : "6000",
    "config_type" : "static",
    "value" : "6000",
    "value_type" : "integer"
  }, {
    "name" : "allow.everyone.if.no.acl.found",
    "valid_values" : "true,false",
    "default_value" : "true",
    "config_type" : "static",
    "value" : "true",
    "value_type" : "enum"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceConfigsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowInstanceConfigsRequest request = new ShowInstanceConfigsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowInstanceConfigsResponse response =
client.showInstanceConfigs(request);
            System.out.println(response.toString());
        }
```

```
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceConfigsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_configs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
```



```
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowInstanceConfigsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowInstanceConfigs(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia [Código de exemplo no API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Configuração obtida.

Códigos de erro

Consulte [Códigos de erro](#).

5.1.8 Modificação das configurações de instância

Função

Essa API é usada para modificar configurações de instâncias.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/configs

Tabela 5-24 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-25 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
kafka_configs	Não	Array of ModifyInstanceConfig objects	Configurações a serem modificadas.

Tabela 5-26 ModifyInstanceConfig

Parâmetro	Obrigatório	Tipo	Descrição
name	Não	String	Nomes das configurações a serem modificadas.
value	Não	String	Novo valor da configuração modificada.

Parâmetros de resposta

Código de status: 200

Tabela 5-27 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa de modificação de configuração.
dynamic_config	Integer	Número de parâmetros de configuração dinâmica a serem modificados.
static_config	Integer	Número de parâmetros de configuração estática a serem modificados.

Exemplo de solicitações

Modificação do limite de tempo limite de conexão ociosa e o intervalo de exclusão do log.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/configs

{
  "kafka_configs" : [ {
    "name" : "connections.max.idle.ms",
    "value" : "500000"
  }, {
    "name" : "log.retention.hours",
    "value" : "66"
  } ]
}
```

Exemplo de respostas

Código de status: 200

Configuração modificada.

```
{
  "job_id" : "8abfa7b38ba79a20018ba9afc550576a",
  "dynamic_config" : 0,
  "static_config" : 2
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Modificação do limite de tempo limite de conexão ociosa e o intervalo de exclusão do log.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ModifyInstanceConfigsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and SK
        // be stored in ciphertext in configuration files or environment variables and
        // decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

ModifyInstanceConfigsRequest request = new ModifyInstanceConfigsRequest();
request.withInstanceId("{instance_id}");
ModifyInstanceConfigsReq body = new ModifyInstanceConfigsReq();
List<ModifyInstanceConfig> listbodyKafkaConfigs = new ArrayList<>();
listbodyKafkaConfigs.add(
    new ModifyInstanceConfig()
        .withName("connections.max.idle.ms")
        .withValue("500000")
);
listbodyKafkaConfigs.add(
    new ModifyInstanceConfig()
        .withName("log.retention.hours")
        .withValue("66")
);
body.withKafkaConfigs(listbodyKafkaConfigs);
request.withBody(body);
try {
    ModifyInstanceConfigsResponse response =
client.modifyInstanceConfigs(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Modificação do limite de tempo limite de conexão ociosa e o intervalo de exclusão do log.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
```

```
client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ModifyInstanceConfigsRequest()
    request.instance_id = "{instance_id}"
    listKafkaConfigsbody = [
        ModifyInstanceConfig(
            name="connections.max.idle.ms",
            value="500000"
        ),
        ModifyInstanceConfig(
            name="log.retention.hours",
            value="66"
        )
    ]
    request.body = ModifyInstanceConfigsReq(
        kafka_configs=listKafkaConfigsbody
    )
    response = client.modify_instance_configs(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Modificação do limite de tempo limite de conexão ociosa e o intervalo de exclusão do log.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ModifyInstanceConfigsRequest{}
```

```
request.InstanceId = "{instance_id}"
nameKafkaConfigs:= "connections.max.idle.ms"
valueKafkaConfigs:= "500000"
nameKafkaConfigs1:= "log.retention.hours"
valueKafkaConfigs1:= "66"
var listKafkaConfigsbody = []model.ModifyInstanceConfig{
    {
        Name: &nameKafkaConfigs,
        Value: &valueKafkaConfigs,
    },
    {
        Name: &nameKafkaConfigs1,
        Value: &valueKafkaConfigs1,
    },
}
request.Body = &model.ModifyInstanceConfigsReq{
    KafkaConfigs: &listKafkaConfigsbody,
}
response, err := client.ModifyInstanceConfigs(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Configuração modificada.

Códigos de erro

Consulte [Códigos de erro](#).

5.2 Gerenciamento de instâncias

5.2.1 Redefinição da senha

Função

Essa API é usada para redefinir a senha de uma instância com SSL ativado.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/password

Tabela 5-28 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-29 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
new_password	Sim	String	A senha pode conter de 8 a 32 caracteres e deve conter pelo menos três tipos dos seguintes caracteres: <ul style="list-style-type: none">● Letras maiúsculas● Letras minúsculas● Dígitos● Caracteres especiais `~!@#\$%^&*()-_+=+ [{}];:'''<>/? e espaços, e não pode começar com um hífen (-).

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Redefinição de uma senha

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/password
{
  "new_password" : "*****"
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Redefinição de uma senha

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetPasswordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ResetPasswordRequest request = new ResetPasswordRequest();
        request.withInstanceId("{instance_id}");
        ResetPasswordReq body = new ResetPasswordReq();
        body.withNewPassword("*****");
        request.withBody(body);
        try {
            ResetPasswordResponse response = client.resetPassword(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Redefinição de uma senha


```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetPasswordRequest()
        request.instance_id = "{instance_id}"
        request.body = ResetPasswordReq(
            new_password="*****"
        )
        response = client.reset_password(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Redefinição de uma senha

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```
WithProjectId(projectId).
Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResetPasswordRequest{
request.InstanceId = "{instance_id}"
request.Body = &model.ResetPasswordReq{
    NewPassword: "*****",
}
response, err := client.ResetPassword(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A senha foi redefinida com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.2 Redefinição da senha do Kafka Manager

Função

Essa API é usada para redefinir a senha do Kafka Manager.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/kafka-manager-password

Tabela 5-30 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-31 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
new_password	Não	String	A senha pode conter de 8 a 32 caracteres e deve conter pelo menos três tipos dos seguintes caracteres: <ul style="list-style-type: none"> ● Letras maiúsculas ● Letras minúsculas ● Dígitos ● Caracteres especiais `~!@#\$%^&*()-_+=\ [{}];:'''<>/? e espaços, e não podem começar com um hífen (-).

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Redefinição da senha do Kafka Manager

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/kafka-manager-password
{
  "new_password" : "*****"
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Redefinição da senha do Kafka Manager

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetManagerPasswordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ResetManagerPasswordRequest request = new ResetManagerPasswordRequest();
        request.withInstanceId("{instance_id}");
        ResetManagerPasswordReq body = new ResetManagerPasswordReq();
        body.withNewPassword("*****");
        request.withBody(body);
        try {
            ResetManagerPasswordResponse response =
client.resetManagerPassword(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Redefinição da senha do Kafka Manager

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__ ('os').getenv("CLOUD_SDK_AK")
    sk = __import__ ('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetManagerPasswordRequest()
        request.instance_id = "{instance_id}"
        request.body = ResetManagerPasswordReq(
            new_password="*****"
        )
        response = client.reset_manager_password(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Redefinição da senha do Kafka Manager

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```
client := kafka.NewKafkaClient(  
    kafka.KafkaClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ResetManagerPasswordRequest{  
    request.InstanceId = "{instance_id}"  
    newPasswordResetManagerPasswordReq:= "*****"  
    request.Body = &model.ResetManagerPasswordReq{  
        NewPassword: &newPasswordResetManagerPasswordReq,  
    }  
    response, err := client.ResetManagerPassword(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A senha foi redefinida com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.3 Reinicialização do Kafka Manager

Função

Esta API é usada para reiniciar o Kafka Manager.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/restart-kafka-manager

Tabela 5-32 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-33 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
result	String	Resultado da execução.
instance_id	String	ID da instância.

Exemplo de solicitações

Reinicialização do Kafka Manager

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/restart-kafka-manager
```

Exemplo de respostas

Código de status: 200

O Kafka Manager foi reiniciado com sucesso.

```
{  
  "result" : "success",  
  "instance_id" : "605cd78c-92dc-4335-8bae-43677f31fd6c"  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class RestartManagerSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        RestartManagerRequest request = new RestartManagerRequest();
        request.withInstanceId("{instance_id}");
        try {
            RestartManagerResponse response = client.restartManager(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```



```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = RestartManagerRequest()
    request.instance_id = "{instance_id}"
    response = client.restart_manager(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestartManagerRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.RestartManager(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	O Kafka Manager foi reiniciado com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.4 Configuração da criação automática dos tópicos

Função

Essa API é usada para ativar ou desativar a criação automática de tópicos.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/autopic

Tabela 5-34 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-35 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
enable_auto_to_pic	Sim	Boolean	Se deve ativar a criação automática de tópicos.

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Ativação da criação automática de tópicos.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/autotopic
{
  "enable_auto_topic" : true
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Ativação da criação automática de tópicos.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceAutoCreateTopicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();
```

```
        UpdateInstanceAutoCreateTopicRequest request = new
UpdateInstanceAutoCreateTopicRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceAutoCreateTopicReq body = new
UpdateInstanceAutoCreateTopicReq();
        body.withEnableAutoTopic(true);
        request.withBody(body);
        try {
            UpdateInstanceAutoCreateTopicResponse response =
client.updateInstanceAutoCreateTopic(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Ativação da criação automática de tópicos.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceAutoCreateTopicRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateInstanceAutoCreateTopicReq(
            enable_auto_topic=True
        )
        response = client.update_instance_auto_create_topic(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Ativação da criação automática de tópicos.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceAutoCreateTopicRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.UpdateInstanceAutoCreateTopicReq{
        EnableAutoTopic: true,
    }
    response, err := client.UpdateInstanceAutoCreateTopic(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A função foi ativada ou desativada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.5 Modificação do endereço IP privado para acesso entre VPCs

Função

Essa API é usada para modificar o endereço IP privado para acesso entre VPCs.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/crossvpc/modify

Tabela 5-36 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-37 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
advertised_ip_contents	Sim	Map<String,String>	Par de chave-valor advertised_ip_contents definido pelo usuário. A chave é o endereço IP dos ouvintes. O valor é o endereço IP ou o nome de domínio de advertised.listeners. NOTA Preencha os itens que não são modificados durante a alteração de endereço IP.

Parâmetros de resposta

Código de status: 200

Tabela 5-38 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
success	Boolean	Resultado da modificação de acesso entre VPCs.
results	Array of results objects	Detalhes do resultado da modificação de acesso entre VPCs.

Tabela 5-39 results

Parâmetro	Tipo	Descrição
advertised_ip	String	Endereço IP ou nome de domínio de advertised.listeners.
success	Boolean	Status da modificação de acesso entre VPCs.
ip	String	Endereço IP dos ouvintes.

Exemplo de solicitações

Modificação do endereço IP privado para acesso entre VPCs.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/crossvpc/modify
{
  "advertised_ip_contents" : {
    "192.168.245.246" : "192.168.245.247",
    "192.168.197.36" : "192.168.197.38",
    "192.168.190.11" : "192.168.190.11"
  }
}
```

Exemplo de respostas

Código de status: 200

O endereço IP privado para acesso entre VPCs foi modificado com sucesso.

```
{
  "success" : true,
  "results" : [ {
    "advertised_ip" : "192.168.197.36",
    "success" : true,
    "ip" : "192.168.197.36"
  }, {
    "advertised_ip" : "192.168.190.11",
    "success" : true,
    "ip" : "192.168.190.11"
  }, {
    "advertised_ip" : "192.168.245.255",
    "success" : true,
    "ip" : "192.168.245.246"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Modificação do endereço IP privado para acesso entre VPCs.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateInstanceCrossVpcIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateInstanceCrossVpcIpRequest request = new
UpdateInstanceCrossVpcIpRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceCrossVpcIpReq body = new UpdateInstanceCrossVpcIpReq();
        Map<String, String> listbodyAdvertisedIpContents = new HashMap<>();
        listbodyAdvertisedIpContents.put("192.168.245.246", "192.168.245.247");
        listbodyAdvertisedIpContents.put("192.168.197.36", "192.168.197.38");
        listbodyAdvertisedIpContents.put("192.168.190.11", "192.168.190.11");
        body.withAdvertisedIpContents(listbodyAdvertisedIpContents);
        request.withBody(body);
        try {
            UpdateInstanceCrossVpcIpResponse response =
client.updateInstanceCrossVpcIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```



```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Modificação do endereço IP privado para acesso entre VPCs.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceCrossVpcIpRequest()
        request.instance_id = "{instance_id}"
        listAdvertisedIpContentsbody = {
            "192.168.245.246": "192.168.245.247",
            "192.168.197.36": "192.168.197.38",
            "192.168.190.11": "192.168.190.11"
        }
        request.body = UpdateInstanceCrossVpcIpReq(
            advertised_ip_contents=listAdvertisedIpContentsbody
        )
        response = client.update_instance_cross_vpc_ip(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Modificação do endereço IP privado para acesso entre VPCs.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceCrossVpcIpRequest{}
    request.InstanceId = "{instance_id}"
    var listAdvertisedIpContentsbody = map[string]string{
        "192.168.245.246": "192.168.245.247",
        "192.168.197.36": "192.168.197.38",
        "192.168.190.11": "192.168.190.11",
    }
    request.Body = &model.UpdateInstanceCrossVpcIpReq{
        AdvertisedIpContents: listAdvertisedIpContentsbody,
    }
    response, err := client.UpdateInstanceCrossVpcIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	O endereço IP privado para acesso entre VPCs foi modificado com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.6 Consulta de metadados do cluster do Kafka

Função

Essa API é usada para consultar metadados de cluster do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/cluster

Tabela 5-40 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-41 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
cluster	cluster object	Informações básicas do cluster.

Tabela 5-42 cluster

Parâmetro	Tipo	Descrição
controller	String	ID do controlador.
brokers	Array of brokers objects	Lista de corretores.
topics_count	Integer	Número de tópicos.
partitions_count	Integer	Número de partições.

Parâmetro	Tipo	Descrição
online_partitions_count	Integer	Número de partições on-line.
replicas_count	Integer	Número de réplicas.
isr_replicas_count	Integer	Número total de réplicas sincronizadas (ISRs).
consumers_count	Integer	Número de grupos de consumidores.

Tabela 5-43 brokers

Parâmetro	Tipo	Descrição
host	String	Endereço IP do corretor.
port	Integer	Número da porta.
broker_id	String	ID do corretor.
is_controller	Boolean	Se o corretor é um controlador.
version	String	Versão do servidor.
register_time	Long	Tempo de registro do corretor, que é um carimbo de data/hora Unix.
is_health	Boolean	Se os corretores de Kafka podem ser conectados.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/cluster
```

Exemplo de respostas

Código de status: 200

Os metadados do cluster do Kafka foram consultados com sucesso.

```
{
  "cluster" : {
    "controller" : "2",
    "brokers" : [ {
      "host" : "192.168.0.159",
      "port" : 9093,
      "broker_id" : "0",
      "is_controller" : false,
      "version" : "1.1.0",
      "register_time" : 1588754647872,
      "is_health" : true
    }, {
      "host" : "192.168.0.48",
      "port" : 9093,
      "broker_id" : "1",
```

```
    "is_controller" : false,
    "version" : "1.1.0",
    "register_time" : 1588754647653,
    "is_health" : true
  }, {
    "host" : "192.168.0.212",
    "port" : 9093,
    "broker_id" : "2",
    "is_controller" : true,
    "version" : "1.1.0",
    "register_time" : 1588754647284,
    "is_health" : true
  } ],
  "topics_count" : 3,
  "partitions_count" : 9,
  "online_partitions_count" : 9,
  "replicas_count" : 27,
  "isr_replicas_count" : 27,
  "consumers_count" : 0
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();
        ShowClusterRequest request = new ShowClusterRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowClusterResponse response = client.showCluster(request);
            System.out.println(response.toString());
        }
```

```
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowClusterRequest()
        request.instance_id = "{instance_id}"
        response = client.show_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
```

```
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowClusterRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Os metadados do cluster do Kafka foram consultados com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.7 Consulta de detalhes do grupo de consumidores

Função

Essa API é usada para consultar detalhes do grupo de consumidores.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/groups/{group}

Tabela 5-44 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
group	Sim	String	Nome do grupo de consumidores.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-45 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
group	group object	Informações do grupo de consumidores.

Tabela 5-46 group

Parâmetro	Tipo	Descrição
group_id	String	Nome do grupo de consumidores.
state	String	Status do grupo de consumidores. O valor pode ser: <ul style="list-style-type: none"> ● Dead: o grupo de consumidores não tem membros nem metadados. ● Empty: o grupo de consumidores tem metadados, mas não tem membros. ● PreparingRebalance: o grupo de consumidores deve ser reequilibrado. ● CompletingRebalance: todos os membros se juntaram ao grupo. ● Stable: os membros do grupo de consumidores podem consumir mensagens normalmente.
coordinator_id	Integer	ID do coordenador.

Parâmetro	Tipo	Descrição
members	Array of members objects	Lista de consumidores.
group_message_offsets	Array of group_message_offsets objects	Deslocamento do consumidor.
assignment_strategy	String	Política de atribuição de partição.

Tabela 5-47 members

Parâmetro	Tipo	Descrição
host	String	Endereço do consumidor.
assignment	Array of assignment objects	Detalhes sobre a partição atribuída ao consumidor.
member_id	String	ID do consumidor.
client_id	String	ID do cliente.

Tabela 5-48 assignment

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.
partitions	Array of integers	Lista de partições.

Tabela 5-49 group_message_offsets

Parâmetro	Tipo	Descrição
partition	Integer	Número da partição.
lag	Long	Número de mensagens restantes que podem ser recuperadas, ou seja, o número de mensagens acumuladas.
topic	String	Nome do tópico.

Parâmetro	Tipo	Descrição
message_current_offset	Long	Deslocamento do consumidor.
message_log_end_offset	Long	Deslocamento de fim de log (LEO).

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/groups/{group}
```

Exemplo de respostas

Código de status: 200

Os detalhes do grupo de consumidores são consultados com sucesso.

```
{
  "group" : {
    "members" : [ {
      "host" : "/172.31.1.102",
      "assignment" : [ {
        "topic" : "test",
        "partitions" : [ 0, 1, 2 ]
      } ],
      "member_id" : "consumer-1-6b8ee551-d499-47d4-9beb-ba1527496785",
      "client_id" : "consumer-1"
    } ],
    "state" : "STABLE",
    "group_id" : "test-consumer-group",
    "coordinator_id" : 2,
    "group_message_offsets" : [ {
      "partition" : 0,
      "lag" : 31396,
      "topic" : "test",
      "message_current_offset" : 935,
      "message_log_end_offset" : 32331
    }, {
      "partition" : 0,
      "lag" : 0,
      "topic" : "aaaa",
      "message_current_offset" : 0,
      "message_log_end_offset" : 0
    }, {
      "partition" : 1,
      "lag" : 31279,
      "topic" : "test",
      "message_current_offset" : 1058,
      "message_log_end_offset" : 32337
    }, {
      "partition" : 1,
      "lag" : 0,
      "topic" : "aaaa",
      "message_current_offset" : 0,
      "message_log_end_offset" : 0
    }, {
      "partition" : 2,
      "lag" : 31603,
      "topic" : "test",
      "message_current_offset" : 739,
      "message_log_end_offset" : 32342
    } ],
  }
}
```

```
"assignment_strategy" : "range"
}
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowGroupsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowGroupsRequest request = new ShowGroupsRequest();
        request.withInstanceId("{instance_id}");
        request.withGroup("{group}");
        try {
            ShowGroupsResponse response = client.showGroups(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowGroupsRequest()
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        response = client.show_groups(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```
client := kafka.NewKafkaClient(  
    kafka.KafkaClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ShowGroupsRequest{  
    request.InstanceId = "{instance_id}"  
    request.Group = "{group}"  
}  
response, err := client.ShowGroups(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Os detalhes do grupo de consumidores são consultados com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.8 Consulta de detalhes do coordenador de uma instância do Kafka

Função

Essa API é usada para consultar detalhes do coordenador de uma instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/coordinators

Tabela 5-50 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-51 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
coordinators	Array of coordinators objects	Lista dos coordenadores de todos os grupos de consumidores.

Tabela 5-52 coordinators

Parâmetro	Tipo	Descrição
group_id	String	ID do grupo de consumidores.
id	Integer	ID do corretor do coordenador.
host	String	Endereço do coordenador.
port	Integer	Número da porta.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/coordinators
```

Exemplo de respostas

Código de status: 200

Os detalhes do coordenador da instância do Kafka são consultados com sucesso.

```
{
  "coordinators" : [ {
    "group_id" : "XXXX",
```

```
    "id" : 2,
    "host" : "172.31.1.15",
    "port" : 9091
  }, {
    "group_id" : "XXXX",
    "id" : 2,
    "host" : "172.31.1.15",
    "port" : 9092
  }, {
    "group_id" : "XXXX",
    "id" : 2,
    "host" : "172.31.1.15",
    "port" : 9092
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowCoordinatorsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCoordinatorsRequest request = new ShowCoordinatorsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowCoordinatorsResponse response = client.showCoordinators(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCoordinatorsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_coordinators(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```



```
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowCoordinatorsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowCoordinators(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Os detalhes do coordenador da instância do Kafka são consultados com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.9 Reatribuição de réplicas de um tópico para uma instância do Kafka

Função

Essa API é usada para reatribuir réplicas de um tópico para uma instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/replicas-reassignment

Tabela 5-53 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico.

Parâmetros de solicitação

Tabela 5-54 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
partitions	Não	Array of partitions objects	Atribuição de réplicas da partição após a alteração.

Tabela 5-55 partitions

Parâmetro	Obrigatório	Tipo	Descrição
partition	Não	Integer	ID da partição.
replicas	Não	Array of integers	ID do corretor onde se espera que a réplica resida. O primeiro número inteiro na matriz representa o ID do corretor de réplica líder. Todas as partições devem ter o mesmo número de réplicas. O número de réplicas não pode ser maior que o número de corretores.

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Reatribuição de partições de tópicos, com réplicas da partição 1 nos corretores 1 e 2 e a réplica líder no corretor 1

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/replicas-reassignment

{
  "partitions" : [ {
    "partition" : 1,
    "replicas" : [ 1, 2 ]
  }, {
    "partition" : 0,
    "replicas" : [ 0, 1 ]
  } ]
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Reatribuição de partições de tópicos, com réplicas da partição 1 nos corretores 1 e 2 e a réplica líder no corretor 1

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateTopicReplicaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
UpdateTopicReplicaRequest request = new UpdateTopicReplicaRequest();
request.withInstanceId("{instance_id}");
request.withTopic("{topic}");
ResetReplicaReq body = new ResetReplicaReq();
List<Integer> listPartitionsReplicas = new ArrayList<>();
listPartitionsReplicas.add(0);
listPartitionsReplicas.add(1);
List<Integer> listPartitionsReplicas1 = new ArrayList<>();
listPartitionsReplicas1.add(1);
listPartitionsReplicas1.add(2);
List<ResetReplicaReqPartitions> listbodyPartitions = new ArrayList<>();
listbodyPartitions.add(
    new ResetReplicaReqPartitions()
        .withPartition(1)
        .withReplicas(listPartitionsReplicas1)
);
listbodyPartitions.add(
    new ResetReplicaReqPartitions()
        .withPartition(0)
        .withReplicas(listPartitionsReplicas)
);
body.withPartitions(listbodyPartitions);
request.withBody(body);
try {
    UpdateTopicReplicaResponse response =
client.updateTopicReplica(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Reatribuição de partições de tópicos, com réplicas da partição 1 nos corretores 1 e 2 e a réplica líder no corretor 1

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudskkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
```

```
client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateTopicReplicaRequest()
    request.instance_id = "{instance_id}"
    request.topic = "{topic}"
    listReplicasPartitions = [
        0,
        1
    ]
    listReplicasPartitions1 = [
        1,
        2
    ]
    listPartitionsbody = [
        ResetReplicaReqPartitions(
            partition=1,
            replicas=listReplicasPartitions1
        ),
        ResetReplicaReqPartitions(
            partition=0,
            replicas=listReplicasPartitions
        )
    ]
    request.body = ResetReplicaReq(
        partitions=listPartitionsbody
    )
    response = client.update_topic_replica(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Reatribuição de partições de tópicos, com réplicas da partição 1 nos corretores 1 e 2 e a réplica líder no corretor 1

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```

        WithProjectId(projectId).
        Build()

        client := kafka.NewKafkaClient(
            kafka.KafkaClientBuilder().
                WithRegion(region.ValueOf("<YOUR REGION>")).
                WithCredential(auth).
                Build())

        request := &model.UpdateTopicReplicaRequest{}
        request.InstanceId = "{instance_id}"
        request.Topic = "{topic}"
        var listReplicasPartitions = []int32{
            int32(0),
            int32(1),
        }
        var listReplicasPartitions1 = []int32{
            int32(1),
            int32(2),
        }
        partitionPartitions:= int32(1)
        partitionPartitions1:= int32(0)
        var listPartitionsbody = []model.ResetReplicaReqPartitions{
            {
                Partition: &partitionPartitions,
                Replicas: &listReplicasPartitions1,
            },
            {
                Partition: &partitionPartitions1,
                Replicas: &listReplicasPartitions,
            },
        }
        request.Body = &model.ResetReplicaReq{
            Partitions: &listPartitionsbody,
        }
        response, err := client.UpdateTopicReplica(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	As réplicas são reatribuídas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.10 Consulta do status de uso de disco dos tópicos

Função

Essa API é usada para consultar o uso do disco do corretor de tópicos.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/topics/diskusage

Tabela 5-56 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-57 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
minSize	Não	String	Consulta de partições pelo espaço em disco usado. Opções: 1 KB, 1 MB e 1 GB. Valor padrão: 1 GB.
top	Não	String	Consulta de partições por uso máximo de disco.
percentage	Não	String	Consulta de partições pela porcentagem do espaço em disco utilizado.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-58 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
broker_list	Array of DiskusageEntity objects	Lista de corretores.

Tabela 5-59 DiskusageEntity

Parâmetro	Tipo	Descrição
broker_name	String	Nome do corretor.
data_disk_size	String	Capacidade do disco.
data_disk_use	String	Espaço em disco usado.
data_disk_free	String	Espaço em disco restante.
data_disk_use_percentage	String	Rótulo da mensagem.
status	String	Rótulo da mensagem.
topic_list	Array of DiskusageTopicEntity objects	Lista de uso de disco dos tópicos.

Tabela 5-60 DiskusageTopicEntity

Parâmetro	Tipo	Descrição
size	String	Uso do disco.
topic_name	String	Nome do tópico.
topic_partition	String	Partição.
percentage	Double	Porcentagem de espaço em disco usado.

Exemplo de solicitações

Consulta do status de uso do disco dos tópicos

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics/diskusage
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.


```
{
  "broker_list" : [ {
    "broker_name" : "broker-0",
    "data_disk_size" : "66G",
    "data_disk_use" : "53M",
    "data_disk_free" : "63G",
    "data_disk_use_percentage" : "1",
    "status" : "Success get info",
    "topic_list" : [ {
      "size" : "12K",
      "topic_name" : "topic-test",
      "topic_partition" : "2",
      "percentage" : 1.7339533025568183E-5
    }, {
      "size" : "12K",
      "topic_name" : "__consumer_offsets",
      "topic_partition" : "4",
      "percentage" : 1.7339533025568183E-5
    }, {
      "size" : "12K",
      "topic_name" : "__consumer_offsets",
      "topic_partition" : "3",
      "percentage" : 1.7339533025568183E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "__trace",
      "topic_partition" : "6",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "__trace",
      "topic_partition" : "4",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "__trace",
      "topic_partition" : "2",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "__trace",
      "topic_partition" : "0",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "topic-test",
      "topic_partition" : "0",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "topic-1568537362",
      "topic_partition" : "2",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "__consumer_offsets",
      "topic_partition" : "7",
      "percentage" : 1.1559688683712121E-5
    } ]
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaTopicPartitionDiskusageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ShowKafkaTopicPartitionDiskusageRequest request = new
        ShowKafkaTopicPartitionDiskusageRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowKafkaTopicPartitionDiskusageResponse response =
            client.showKafkaTopicPartitionDiskusage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowKafkaTopicPartitionDiskusageRequest()
    request.instance_id = "{instance_id}"
    response = client.show_kafka_topic_partition_diskusage(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKafkaTopicPartitionDiskusageRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowKafkaTopicPartitionDiskusage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```

} else {
    fmt.Println(err)
}
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.11 Consulta de todos os grupos de consumidores

Função

Essa API é usada para consultar todos os grupos de consumidores.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/groups

Tabela 5-61 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-62 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
offset	Não	String	Deslocamento, que é a posição onde a consulta começa. O valor deve ser maior ou igual a 0.
limit	Não	String	Número máximo de IDs de grupos de consumidores retornados na consulta atual. O valor padrão é 10 . O valor varia de 1 a 50.
group	Não	String	Filtrar nomes de grupos de consumidores que contenham palavras-chave específicas.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-63 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
groups	Array of GroupInfoSimple objects	Todos os grupos de consumidores.
total	Integer	Número total de grupos de consumidores.

Tabela 5-64 GroupInfoSimple

Parâmetro	Tipo	Descrição
createdAt	Long	Hora de criação.
group_id	String	ID do grupo de consumidores.

Parâmetro	Tipo	Descrição
state	String	Status do grupo de consumidores. O valor pode ser: <ul style="list-style-type: none"> ● Dead: o grupo de consumidores não tem membros ou metadados. ● Empty: o grupo de consumidores tem metadados, mas não tem membros. ● PreparingRebalance: o grupo de consumidores deve ser reequilibrado. ● CompletingRebalance: todos os membros se juntaram ao grupo. ● Stable: os membros do grupo de consumidores podem consumir mensagens.
coordinator_id	Integer	ID do coordenador.
group_desc	String	Descrição do grupo de consumidores.
lag	Long	Número de mensagens acumuladas.

Exemplo de solicitações

Consulta da lista de grupos de consumidores

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/groups?
offset={offset}&limit={limit}&group={group}
```

Exemplo de respostas

Código de status: 200

Todos os grupos de consumidores são consultados com sucesso.

```
{
  "groups" : [ {
    "createdAt" : 1691401194847,
    "group_id" : "consumer-1",
    "state" : "EMPTY",
    "coordinator_id" : 1,
    "lag" : 0,
    "group_desc" : null
  }, {
    "createdAt" : 1691401194960,
    "group_id" : "consumer-2",
    "state" : "STABLE",
    "coordinator_id" : 2,
    "lag" : 0,
    "group_desc" : null
  }, {
    "createdAt" : 1691401207309,
    "group_id" : "consumer-3",
    "state" : "STABLE",
    "coordinator_id" : 3,
    "lag" : 0,
    "group_desc" : null
  } ],
  "total" : 3
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListInstanceConsumerGroupsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and SK
        // be stored in ciphertext in configuration files or environment variables and
        // decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ListInstanceConsumerGroupsRequest request = new
        ListInstanceConsumerGroupsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListInstanceConsumerGroupsResponse response =
            client.listInstanceConsumerGroups(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceConsumerGroupsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_instance_consumer_groups(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
request := &model.ListInstanceConsumerGroupsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListInstanceConsumerGroups(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Todos os grupos de consumidores são consultados com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.12 Exclusão de grupos de consumidores de uma instância do Kafka em lotes

Função

Essa API é usada para excluir vários grupos de consumidores de uma instância do Kafka em lotes.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/groups/batch-delete

Tabela 5-65 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-66 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
group_ids	Sim	Array of strings	IDs de todos os grupos de consumidores a serem excluídos.

Parâmetros de resposta

Código de status: 200

Tabela 5-67 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
failed_groups	Array of failed_groups objects	Lista de grupos de consumidores que não foram excluídos.
total	Integer	Número de registros que não foram excluídos.

Tabela 5-68 failed_groups

Parâmetro	Tipo	Descrição
group_id	String	ID dos grupos de consumidores que não foram excluídos.
error_message	String	Causa da falha de exclusão.

Exemplo de solicitações

Exclusão em lote de grupos de consumidores

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/groups/batch-delete
{
  "group_ids" : [ "get-sync-group0", "get-sync-group1" ]
}
```

Exemplo de respostas

Código de status: 200

Os grupos de consumidores foram excluídos com sucesso.

```
{
  "failed_groups" : [ {
    "group_id" : "test-1",
```

```
    "error_message" : "UNKNOW"
  }, {
    "group_id" : "test-2",
    "error_message" : "UNKNOW"
  } ],
  "total" : 2
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Exclusão em lote de grupos de consumidores

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();

        BatchDeleteGroupRequest request = new BatchDeleteGroupRequest();
        request.withInstanceId("{instance_id}");
        BatchDeleteGroupReq body = new BatchDeleteGroupReq();
        List<String> listbodyGroupIds = new ArrayList<>();
        listbodyGroupIds.add("get-sync-group0");
        listbodyGroupIds.add("get-sync-group1");
        body.withGroupIds(listbodyGroupIds);
        request.withBody(body);
        try {
            BatchDeleteGroupResponse response = client.batchDeleteGroup(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

Exclusão em lote de grupos de consumidores

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR_REGION>")) \
        .build()

    try:
        request = BatchDeleteGroupRequest()
        request.instance_id = "{instance_id}"
        listGroupIdsbody = [
            "get-sync-group0",
            "get-sync-group1"
        ]
        request.body = BatchDeleteGroupReq(
            group_ids=listGroupIdsbody
        )
        response = client.batch_delete_group(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Exclusão em lote de grupos de consumidores

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteGroupRequest{}
    request.InstanceId = "{instance_id}"
    var listGroupIdsbody = []string{
        "get-sync-group0",
        "get-sync-group1",
    }
    request.Body = &model.BatchDeleteGroupReq{
        GroupIds: listGroupIdsbody,
    }
    response, err := client.BatchDeleteGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Os grupos de consumidores foram excluídos com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.13 Criação de um grupo de consumidores

Função

Essa API é usada para criar um grupo de consumidores.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/group

Tabela 5-69 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-70 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
group_name	Sim	String	Nome do grupo de consumidores.
group_desc	Não	String	Descrição do grupo de consumidores. Mínimo: 0 Máximo: 200

Parâmetros de resposta

Código de status: 400

Tabela 5-71 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Exemplo de solicitações

Criação de um grupo de consumidores chamado test

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/group
{
  "group_name" : "test"
}
```

Exemplo de respostas

Código de status: 200

A criação foi bem-sucedida.

```
success
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Criação de um grupo de consumidores chamado test

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateKafkaConsumerGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();

        CreateKafkaConsumerGroupRequest request = new
        CreateKafkaConsumerGroupRequest();
        request.withInstanceId("{instance_id}");
        CreateGroupReq body = new CreateGroupReq();
        body.withGroupName("test");
    }
}
```

```
        request.withBody(body);
    }
    try {
        CreateKafkaConsumerGroupResponse response =
client.createKafkaConsumerGroup(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Criação de um grupo de consumidores chamado test

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateKafkaConsumerGroupRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateGroupReq(
            group_name="test"
        )
        response = client.create_kafka_consumer_group(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Criação de um grupo de consumidores chamado test

```
package main
```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateKafkaConsumerGroupRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.CreateGroupReq{
        GroupName: "test",
    }
    response, err := client.CreateKafkaConsumerGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A criação foi bem-sucedida.
400	Falha na criação.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.14 Edição de um grupo de consumidores

Função

Essa API é usada para editar um grupo de consumidores.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}

Tabela 5-72 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens.
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
group	Sim	String	Grupo de consumidores.

Parâmetros de solicitação

Tabela 5-73 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
group_name	Sim	String	Nome do grupo de consumidores.
group_desc	Não	String	Descrição do grupo de consumidores. Mínimo: 0 Máximo: 200

Parâmetros de resposta

Código de status: **400**

Tabela 5-74 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Exemplo de solicitações

Edite um grupo de consumidores. Digite "desc" na descrição.

```
PUT https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/groups/test
{
  "group_name" : "test",
  "group_desc" : "desc"
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Edite um grupo de consumidores. Digite "desc" na descrição.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceConsumerGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateInstanceConsumerGroupRequest request = new
UpdateInstanceConsumerGroupRequest();
request.withEngine("{engine}");
request.withInstanceId("{instance_id}");
request.withGroup("{group}");
CreateGroupReq body = new CreateGroupReq();
body.withGroupDesc("desc");
body.withGroupName("test");
request.withBody(body);
try {
    UpdateInstanceConsumerGroupResponse response =
client.updateInstanceConsumerGroup(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Edite um grupo de consumidores. Digite "desc" na descrição.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceConsumerGroupRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        request.body = CreateGroupReq(
            group_desc="desc",
```

```
        group_name="test"
    )
    response = client.update_instance_consumer_group(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Edite um grupo de consumidores. Digite "desc" na descrição.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceConsumerGroupRequest{}
    request.Engine = "{engine}"
    request.InstanceId = "{instance_id}"
    request.Group = "{group}"
    groupDescCreateGroupReq := "desc"
    request.Body = &model.CreateGroupReq{
        GroupDesc: &groupDescCreateGroupReq,
        GroupName: "test",
    }
    response, err := client.UpdateInstanceConsumerGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Editado.
400	Falha na edição.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.15 Inicialização da reatribuição de partição para uma instância do Kafka

Função

Essa API é usada para enviar uma tarefa de rebalanceamento de partição a uma instância do Kafka ou calcular o tempo estimado de rebalanceamento.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/kafka/{project_id}/instances/{instance_id}/reassign

Tabela 5-75 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-76 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
reassignments	Sim	Array of PartitionReassignEntity objects	Plano de reatribuição.
throttle	Não	Integer	Limite de reatribuição.
is_schedule	Não	Boolean	Indica se a tarefa está agendada. Se não, is_schedule e execute_at podem ser deixados em branco. Se sim, is_schedule é true e execute_at deve ser especificado.
execute_at	Não	Long	Horário de agendamento. O valor é um carimbo de data/hora UNIX, em ms.
time_estimate	Não	Boolean	Defina true para executar tarefas de estimativa de tempo e false para executar tarefas de rebalanceamento. Padrão: false

Tabela 5-77 PartitionReassignEntity

Parâmetro	Obrigatório	Tipo	Descrição
topic	Sim	String	Nome do tópico.
brokers	Não	Array of integers	Lista de corretores aos quais as partições são reatribuídas. Este parâmetro é obrigatório na atribuição automática.
replication_factor	Não	Integer	Fator de replicação, que pode ser especificado na atribuição automática.
assignment	Não	Array of TopicAssignment objects	Plano de atribuição especificado manualmente. O parâmetro brokers e este parâmetro não podem estar vazios ao mesmo tempo.

Tabela 5-78 TopicAssignment

Parâmetro	Obrigatório	Tipo	Descrição
partition	Não	Integer	Número da partição na atribuição manual.
partition_brokers	Não	Array of integers	Lista de corretores a serem atribuídos a uma partição na atribuição manual.

Parâmetros de resposta

Código de status: 200

Tabela 5-79 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa. Somente job_id é retornado para uma tarefa de rebalanceamento.
reassignment_time	Integer	Tempo estimado, em segundos. Somente reassignment_time é retornado para uma tarefa de estimativa de tempo.

Exemplo de solicitações

```
POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/reassign
{
  "reassignments" : [ {
    "topic" : "topic-1513476102",
    "brokers" : [ 0, 1, 2 ],
    "replication_factor" : 3,
    "assignment" : [ {
      "partition" : 0,
      "partition_brokers" : [ 0, 1, 2 ]
    }, {
      "partition" : 1,
      "partition_brokers" : [ 1, 2, 0 ]
    }, {
      "partition" : 2,
      "partition_brokers" : [ 2, 0, 1 ]
    } ]
  }, {
    "topic" : "topic-1513558717",
    "brokers" : [ 0, 1, 4 ],
    "replication_factor" : 3,
    "assignment" : [ {
      "partition" : 0,
      "partition_brokers" : [ 0, 1, 2 ]
    }, {
      "partition" : 1,
      "partition_brokers" : [ 1, 2, 0 ]
    }, {
      "partition" : 2,
      "partition_brokers" : [ 2, 0, 1 ]
    } ]
  } ]
}
```



```
} ],  
  "throttle" : 10000000,  
  "time_estimate" : false  
}
```

Exemplo de respostas

Código de status: 200

Tarefa de rebalanceamento enviada ou o tempo estimado é retornado.

```
{  
  "job_id" : "8a2c259182ab0e9d0182ab1882560009",  
  "reassignment_time" : 10  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateReassignmentTaskSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
        plaintext, which has great security risks. It is recommended that the AK and SK  
        be stored in ciphertext in configuration files or environment variables and  
        decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for  
        authentication. Before running this example, set environment variables  
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateReassignmentTaskRequest request = new  
        CreateReassignmentTaskRequest();  
        request.withInstanceId("{instance_id}");  
        PartitionReassignRequest body = new PartitionReassignRequest();  
        List<Integer> listAssignmentPartitionBrokers = new ArrayList<>();  
        listAssignmentPartitionBrokers.add(2);  
        listAssignmentPartitionBrokers.add(0);  
        listAssignmentPartitionBrokers.add(1);  
    }  
}
```

```
List<Integer> listAssignmentPartitionBrokers1 = new ArrayList<>();
listAssignmentPartitionBrokers1.add(1);
listAssignmentPartitionBrokers1.add(2);
listAssignmentPartitionBrokers1.add(0);
List<Integer> listAssignmentPartitionBrokers2 = new ArrayList<>();
listAssignmentPartitionBrokers2.add(0);
listAssignmentPartitionBrokers2.add(1);
listAssignmentPartitionBrokers2.add(2);
List<TopicAssignment> listReassignmentsAssignment = new ArrayList<>();
listReassignmentsAssignment.add(
    new TopicAssignment()
        .withPartition(0)
        .withPartitionBrokers(listAssignmentPartitionBrokers2)
);
listReassignmentsAssignment.add(
    new TopicAssignment()
        .withPartition(1)
        .withPartitionBrokers(listAssignmentPartitionBrokers1)
);
listReassignmentsAssignment.add(
    new TopicAssignment()
        .withPartition(2)
        .withPartitionBrokers(listAssignmentPartitionBrokers)
);
List<Integer> listReassignmentsBrokers = new ArrayList<>();
listReassignmentsBrokers.add(0);
listReassignmentsBrokers.add(1);
listReassignmentsBrokers.add(4);
List<Integer> listAssignmentPartitionBrokers3 = new ArrayList<>();
listAssignmentPartitionBrokers3.add(2);
listAssignmentPartitionBrokers3.add(0);
listAssignmentPartitionBrokers3.add(1);
List<Integer> listAssignmentPartitionBrokers4 = new ArrayList<>();
listAssignmentPartitionBrokers4.add(1);
listAssignmentPartitionBrokers4.add(2);
listAssignmentPartitionBrokers4.add(0);
List<Integer> listAssignmentPartitionBrokers5 = new ArrayList<>();
listAssignmentPartitionBrokers5.add(0);
listAssignmentPartitionBrokers5.add(1);
listAssignmentPartitionBrokers5.add(2);
List<TopicAssignment> listReassignmentsAssignment1 = new ArrayList<>();
listReassignmentsAssignment1.add(
    new TopicAssignment()
        .withPartition(0)
        .withPartitionBrokers(listAssignmentPartitionBrokers5)
);
listReassignmentsAssignment1.add(
    new TopicAssignment()
        .withPartition(1)
        .withPartitionBrokers(listAssignmentPartitionBrokers4)
);
listReassignmentsAssignment1.add(
    new TopicAssignment()
        .withPartition(2)
        .withPartitionBrokers(listAssignmentPartitionBrokers3)
);
List<Integer> listReassignmentsBrokers1 = new ArrayList<>();
listReassignmentsBrokers1.add(0);
listReassignmentsBrokers1.add(1);
listReassignmentsBrokers1.add(2);
List<PartitionReassignEntity> listbodyReassignments = new ArrayList<>();
listbodyReassignments.add(
    new PartitionReassignEntity()
        .withTopic("topic-1513476102")
        .withBrokers(listReassignmentsBrokers1)
        .withReplicationFactor(3)
        .withAssignment(listReassignmentsAssignment1)
);
listbodyReassignments.add(
```

```
        new PartitionReassignEntity()
            .withTopic("topic-1513558717")
            .withBrokers(listReassignmentsBrokers)
            .withReplicationFactor(3)
            .withAssignment(listReassignmentsAssignment)
        );
        body.withTimeEstimate(false);
        body.withThrottle(10000000);
        body.withReassignments(listbodyReassignments);
        request.withBody(body);
        try {
            CreateReassignmentTaskResponse response =
client.createReassignmentTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateReassignmentTaskRequest()
        request.instance_id = "{instance_id}"
        listPartitionBrokersAssignment = [
            2,
            0,
            1
        ]
        listPartitionBrokersAssignment1 = [
            1,
            2,
            0
        ]
    
```

```
listPartitionBrokersAssignment2 = [
    0,
    1,
    2
]
listAssignmentReassignments = [
    TopicAssignment(
        partition=0,
        partition_brokers=listPartitionBrokersAssignment2
    ),
    TopicAssignment(
        partition=1,
        partition_brokers=listPartitionBrokersAssignment1
    ),
    TopicAssignment(
        partition=2,
        partition_brokers=listPartitionBrokersAssignment
    )
]
listBrokersReassignments = [
    0,
    1,
    4
]
listPartitionBrokersAssignment3 = [
    2,
    0,
    1
]
listPartitionBrokersAssignment4 = [
    1,
    2,
    0
]
listPartitionBrokersAssignment5 = [
    0,
    1,
    2
]
listAssignmentReassignments1 = [
    TopicAssignment(
        partition=0,
        partition_brokers=listPartitionBrokersAssignment5
    ),
    TopicAssignment(
        partition=1,
        partition_brokers=listPartitionBrokersAssignment4
    ),
    TopicAssignment(
        partition=2,
        partition_brokers=listPartitionBrokersAssignment3
    )
]
listBrokersReassignments1 = [
    0,
    1,
    2
]
listReassignmentsbody = [
    PartitionReassignEntity(
        topic="topic-1513476102",
        brokers=listBrokersReassignments1,
        replication_factor=3,
        assignment=listAssignmentReassignments1
    ),
    PartitionReassignEntity(
        topic="topic-1513558717",
        brokers=listBrokersReassignments,
        replication_factor=3,

```

```
        assignment=listAssignmentReassignments
    )
]
request.body = PartitionReassignRequest(
    time_estimate=False,
    throttle=10000000,
    reassignments=listReassignmentsbody
)
response = client.create_reassignment_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateReassignmentTaskRequest{}
    request.InstanceId = "{instance_id}"
    var listPartitionBrokersAssignment = []int32{
        int32(2),
        int32(0),
        int32(1),
    }
    var listPartitionBrokersAssignment1 = []int32{
        int32(1),
        int32(2),
        int32(0),
    }
    var listPartitionBrokersAssignment2 = []int32{
        int32(0),
        int32(1),
        int32(2),
    }
}
```

```
partitionAssignment:= int32(0)
partitionAssignment1:= int32(1)
partitionAssignment2:= int32(2)
var listAssignmentReassignments = []model.TopicAssignment{
    {
        Partition: &partitionAssignment,
        PartitionBrokers: &listPartitionBrokersAssignment2,
    },
    {
        Partition: &partitionAssignment1,
        PartitionBrokers: &listPartitionBrokersAssignment1,
    },
    {
        Partition: &partitionAssignment2,
        PartitionBrokers: &listPartitionBrokersAssignment,
    },
}
var listBrokersReassignments = []int32{
    int32(0),
    int32(1),
    int32(4),
}
var listPartitionBrokersAssignment3 = []int32{
    int32(2),
    int32(0),
    int32(1),
}
var listPartitionBrokersAssignment4 = []int32{
    int32(1),
    int32(2),
    int32(0),
}
var listPartitionBrokersAssignment5 = []int32{
    int32(0),
    int32(1),
    int32(2),
}
partitionAssignment3:= int32(0)
partitionAssignment4:= int32(1)
partitionAssignment5:= int32(2)
var listAssignmentReassignments1 = []model.TopicAssignment{
    {
        Partition: &partitionAssignment3,
        PartitionBrokers: &listPartitionBrokersAssignment5,
    },
    {
        Partition: &partitionAssignment4,
        PartitionBrokers: &listPartitionBrokersAssignment4,
    },
    {
        Partition: &partitionAssignment5,
        PartitionBrokers: &listPartitionBrokersAssignment3,
    },
}
var listBrokersReassignments1 = []int32{
    int32(0),
    int32(1),
    int32(2),
}
replicationFactorReassignments:= int32(3)
replicationFactorReassignments1:= int32(3)
var listReassignmentsbody = []model.PartitionReassignEntity{
    {
        Topic: "topic-1513476102",
        Brokers: &listBrokersReassignments1,
        ReplicationFactor: &replicationFactorReassignments,
        Assignment: &listAssignmentReassignments1,
    },
    {
```

```
        Topic: "topic-1513558717",
        Brokers: &listBrokersReassignments,
        ReplicationFactor: &replicationFactorReassignments1,
        Assignment: &listAssignmentReassignments,
    },
}
timeEstimatePartitionReassignRequest:= false
throttlePartitionReassignRequest:= int32(10000000)
request.Body = &model.PartitionReassignRequest{
    TimeEstimate: &timeEstimatePartitionReassignRequest,
    Throttle: &throttlePartitionReassignRequest,
    Reassignments: listReassignmentsbody,
}
response, err := client.CreateReassignmentTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Tarefa de rebalanceamento enviada ou o tempo estimado é retornado.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.16 Desativação do Kafka Manager

Função

Esta API é usada para desativar o Kafka Manager. Depois que o Kafka Manager for desativado, as APIs de gerenciamento relacionadas ficarão indisponíveis.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

DELETE /v2/{project_id}/kafka/instances/{instance_id}/management

Tabela 5-80 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto. Mínimo: 1 Máximo: 64
instance_id	Sim	String	ID da instância. Mínimo: 1 Máximo: 64

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

```
DELETE https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/management
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CloseKafkaManagerSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
```



```
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CloseKafkaManagerRequest request = new CloseKafkaManagerRequest();
request.withInstanceId("{instance_id}");
try {
    CloseKafkaManagerResponse response =
client.closeKafkaManager(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CloseKafkaManagerRequest()
        request.instance_id = "{instance_id}"
        response = client.close_kafka_manager(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CloseKafkaManagerRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.CloseKafkaManager(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.17 Exclusão de uma cota de usuário ou cliente

Função

Essa API é usada para enviar uma tarefa de excluir uma cota de usuário ou cliente de uma instância do Kafka. Um `job_id` será retornado se a tarefa for bem-sucedida.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

DELETE /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Tabela 5-81 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
instance_id	Sim	String	ID da instância.
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.

Parâmetros de solicitação

Tabela 5-82 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
user	Não	String	Nome de usuário. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.
client	Não	String	ID do cliente. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.

Parâmetro	Obrigatório	Tipo	Descrição
user-default	Não	Boolean	Indica se as configurações padrão do usuário devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os usuários. Nesse caso, o nome de usuário não pode ser definido. ● Não: aplicar a cota a usuários específicos. Nesse caso, defina o nome de usuário.
client-default	Não	Boolean	Indica se as configurações padrão do cliente devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os clientes. Nesse caso, o ID do cliente não pode ser definido. ● Não: aplicar a cota a clientes específicos. Nesse caso, defina o ID do cliente.

Parâmetros de resposta

Código de status: 200

Tabela 5-83 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa para exclusão de cotas.

Exemplo de solicitações

Exclusão de uma cota de usuário ou cliente

```
DELETE https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota
{
  "user" : "",
  "client" : "",
  "user-default" : false,
  "client-default" : true
}
```

Exemplo de respostas

Código de status: 200

Cotas de usuário ou cliente excluídas.

```
{
  "job_id" : "ff8080828bdc0f64018bdcadfd8f00d7"
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Exclusão de uma cota de usuário ou cliente

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteKafkaUserClientQuotaTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteKafkaUserClientQuotaTaskRequest request = new
        DeleteKafkaUserClientQuotaTaskRequest();
        request.withInstanceId("{instance_id}");
        DeleteKafkaUserClientQuotaTaskReq body = new
        DeleteKafkaUserClientQuotaTaskReq();
        body.withClientDefault(true);
        body.withUserDefault(false);
        body.withClient("");
        body.withUser("");
        request.withBody(body);
        try {
            DeleteKafkaUserClientQuotaTaskResponse response =
            client.deleteKafkaUserClientQuotaTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Exclusão de uma cota de usuário ou cliente

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteKafkaUserClientQuotaTaskRequest()
        request.instance_id = "{instance_id}"
        request.body = DeleteKafkaUserClientQuotaTaskReq(
            client_default=True,
            user_default=False,
            client="",
            user=""
        )
        response = client.delete_kafka_user_client_quota_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Exclusão de uma cota de usuário ou cliente

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteKafkaUserClientQuotaTaskRequest{}
    request.InstanceId = "{instance_id}"
    clientDefaultDeleteKafkaUserClientQuotaTaskReq:= true
    userDefaultDeleteKafkaUserClientQuotaTaskReq:= false
    clientDeleteKafkaUserClientQuotaTaskReq:= ""
    userDeleteKafkaUserClientQuotaTaskReq:= ""
    request.Body = &model.DeleteKafkaUserClientQuotaTaskReq{
        ClientDefault: &clientDefaultDeleteKafkaUserClientQuotaTaskReq,
        UserDefault: &userDefaultDeleteKafkaUserClientQuotaTaskReq,
        Client: &clientDeleteKafkaUserClientQuotaTaskReq,
        User: &userDeleteKafkaUserClientQuotaTaskReq,
    }
    response, err := client.DeleteKafkaUserClientQuotaTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Cotas de usuário ou cliente excluídas.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.18 Consulta de cotas de usuários ou clientes

Função

Essa API é usada para consultar cotas de uma instância do Kafka. Uma lista de cotas é retornada se a consulta for bem-sucedida.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Tabela 5-84 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-85 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
offset	Não	Integer	Deslocamento. Os registros após esse deslocamento serão consultados. Mínimo: 0 Máximo: 10000
limit	Não	Integer	Número máximo de registros que podem ser retornados. Mínimo: 1 Máximo: 50

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-86 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
quotas	Array of Quota objects	Cotas de clientes.
count	Integer	Número de cotas de usuário ou cliente.

Tabela 5-87 Cota

Parâmetro	Tipo	Descrição
user	String	Nome de usuário. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.
client	String	ID do cliente. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.
user-default	Boolean	Indica se as configurações padrão do usuário devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os usuários. Nesse caso, o nome de usuário não pode ser definido. ● Não: aplicar a cota a usuários específicos. Nesse caso, defina o nome de usuário.
client-default	Boolean	Indica se as configurações padrão do cliente devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os clientes. Nesse caso, o ID do cliente não pode ser definido. ● Não: aplicar a cota a clientes específicos. Nesse caso, defina o ID do cliente.
producer-byte-rate	Long	Limite da taxa de produção. A unidade é byte/s.
consumer-byte-rate	Long	Limite da taxa de consumo. A unidade é byte/s. NOTA O limite da taxa de produção ou o limite da taxa de consumo deve ser especificado.

Exemplo de solicitações

Nenhum

Exemplo de respostas

Código de status: 200

Cotas de cliente consultadas.

```
{
  "quotas" : [ {
    "user" : "",
    "client" : "",
    "user-default" : false,
    "client-default" : true,
    "producer-byte-rate" : 2097152,
    "consumer-byte-rate" : 2097152
  } ],
  "count" : 1
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaUserClientQuotaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowKafkaUserClientQuotaRequest request = new
        ShowKafkaUserClientQuotaRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowKafkaUserClientQuotaResponse response =
            client.showKafkaUserClientQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowKafkaUserClientQuotaRequest()
        request.instance_id = "{instance_id}"
        response = client.show_kafka_user_client_quota(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
```

```
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowKafkaUserClientQuotaRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowKafkaUserClientQuota(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Cotas de cliente consultadas.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.19 Criação de cotas de usuário ou cliente

Função

Essa API é usada para enviar uma tarefa para criar cotas de usuário ou cliente para uma instância do Kafka. Um `job_id` será retornado se a tarefa for bem-sucedida.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Tabela 5-88 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
instance_id	Sim	String	ID da instância.
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.

Parâmetros de solicitação

Tabela 5-89 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
user	Não	String	Nome de usuário. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.
client	Não	String	ID do cliente. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.
user-default	Não	Boolean	Indica se as configurações padrão do usuário devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os usuários. Nesse caso, o nome de usuário não pode ser definido. ● Não: aplicar a cota a usuários específicos. Nesse caso, defina o nome de usuário.
client-default	Não	Boolean	Indica se as configurações padrão do cliente devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os clientes. Nesse caso, o ID do cliente não pode ser definido. ● Não: aplicar a cota a clientes específicos. Nesse caso, defina o ID do cliente.

Parâmetro	Obrigatório	Tipo	Descrição
producer-byte-rate	Não	Long	Limite da taxa de produção. A unidade é byte/s.
consumer-byte-rate	Não	Long	Limite da taxa de consumo. A unidade é byte/s. NOTA O limite da taxa de produção ou o limite da taxa de consumo deve ser especificado.

Parâmetros de resposta

Código de status: 200

Tabela 5-90 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa para criar cotas.

Exemplo de solicitações

Criação de cotas de usuários ou clientes

```
POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota
{
  "user" : "",
  "client" : "",
  "user-default" : false,
  "client-default" : true,
  "producer-byte-rate" : 3145728,
  "consumer-byte-rate" : 2097152
}
```

Exemplo de respostas

Código de status: 200

Cotas de usuário ou cliente criadas.

```
{
  "job_id" : "ff8080828bdc0f64018bdcadfd8f00d7"
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Criação de cotas de usuários ou clientes

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateKafkaUserClientQuotaTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and SK
        // be stored in ciphertext in configuration files or environment variables and
        // decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateKafkaUserClientQuotaTaskRequest request = new
        CreateKafkaUserClientQuotaTaskRequest();
        request.withInstanceId("{instance_id}");
        CreateKafkaUserClientQuotaTaskReq body = new
        CreateKafkaUserClientQuotaTaskReq();
        body.withConsumerByteRate(2097152L);
        body.withProducerByteRate(3145728L);
        body.withClientDefault(true);
        body.withUserDefault(false);
        body.withClient("");
        body.withUser("");
        request.withBody(body);
        try {
            CreateKafkaUserClientQuotaTaskResponse response =
            client.createKafkaUserClientQuotaTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Criação de cotas de usuários ou clientes

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateKafkaUserClientQuotaTaskRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateKafkaUserClientQuotaTaskReq(
            consumer_byte_rate=2097152,
            producer_byte_rate=3145728,
            client_default=True,
            user_default=False,
            client="",
            user=""
        )
        response = client.create_kafka_user_client_quota_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Criação de cotas de usuários ou clientes

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```



```
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateKafkaUserClientQuotaTaskRequest{
    request.InstanceId = "{instance_id}"
    consumerByteRateCreateKafkaUserClientQuotaTaskReq:= int64(2097152)
    producerByteRateCreateKafkaUserClientQuotaTaskReq:= int64(3145728)
    clientDefaultCreateKafkaUserClientQuotaTaskReq:= true
    userDefaultCreateKafkaUserClientQuotaTaskReq:= false
    clientCreateKafkaUserClientQuotaTaskReq:= ""
    userCreateKafkaUserClientQuotaTaskReq:= ""
    request.Body = &model.CreateKafkaUserClientQuotaTaskReq{
        ConsumerByteRate: &consumerByteRateCreateKafkaUserClientQuotaTaskReq,
        ProducerByteRate: &producerByteRateCreateKafkaUserClientQuotaTaskReq,
        ClientDefault: &clientDefaultCreateKafkaUserClientQuotaTaskReq,
        UserDefault: &userDefaultCreateKafkaUserClientQuotaTaskReq,
        Client: &clientCreateKafkaUserClientQuotaTaskReq,
        User: &userCreateKafkaUserClientQuotaTaskReq,
    }
}
response, err := client.CreateKafkaUserClientQuotaTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Cotas de usuário ou cliente criadas.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.20 Modificação de cotas de usuários ou clientes

Função

Essa API é usada para enviar uma tarefa para modificar cotas de usuário ou cliente de uma instância do Kafka. Um `job_id` será retornado se a tarefa for bem-sucedida.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT `/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota`

Tabela 5-91 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
<code>instance_id</code>	Sim	String	ID da instância.
<code>project_id</code>	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.

Parâmetros de solicitação

Tabela 5-92 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
<code>user</code>	Não	String	Nome de usuário. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.
<code>client</code>	Não	String	ID do cliente. O nome de usuário ou o ID do cliente deve ser especificado se as cotas não forem aplicadas a todos os usuários ou clientes.

Parâmetro	Obrigatório	Tipo	Descrição
user-default	Não	Boolean	Indica se as configurações padrão do usuário devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os usuários. Nesse caso, o nome de usuário não pode ser definido. ● Não: aplicar a cota a usuários específicos. Nesse caso, defina o nome de usuário.
client-default	Não	Boolean	Indica se as configurações padrão do cliente devem ser usadas. <ul style="list-style-type: none"> ● Sim: aplicar a cota a todos os clientes. Nesse caso, o ID do cliente não pode ser definido. ● Não: aplicar a cota a clientes específicos. Nesse caso, defina o ID do cliente.
producer-byte-rate	Não	Long	Limite da taxa de produção. A unidade é byte/s.
consumer-byte-rate	Não	Long	Limite da taxa de consumo. A unidade é byte/s. NOTA O limite da taxa de produção ou o limite da taxa de consumo deve ser especificado.

Parâmetros de resposta

Código de status: 200

Tabela 5-93 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa para modificar as cotas.

Exemplo de solicitações

Modificação de cotas de usuários ou clientes

```
PUT https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota
{
  "user" : "",
  "client" : "",
}
```

```
"user-default" : false,  
"client-default" : true,  
"producer-byte-rate" : 3145728,  
"consumer-byte-rate" : 2097152  
}
```

Exemplo de respostas

Código de status: 200

Cotas de usuário ou cliente modificadas.

```
{  
  "job_id" : "8abfa7b38ba79a20018ba9afc550576a"  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Modificação de cotas de usuários ou clientes

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class UpdateKafkaUserClientQuotaTaskSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
plaintext, which has great security risks. It is recommended that the AK and SK  
be stored in ciphertext in configuration files or environment variables and  
decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for  
authentication. Before running this example, set environment variables  
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateKafkaUserClientQuotaTaskRequest request = new  
UpdateKafkaUserClientQuotaTaskRequest();  
        request.withInstanceId("{instance_id}");  
        UpdateKafkaUserClientQuotaTaskReq body = new  
UpdateKafkaUserClientQuotaTaskReq();  
        body.withConsumerByteRate(2097152L);  
        body.withProducerByteRate(3145728L);
```

```
        body.withClientDefault(true);
        body.withUserDefault(false);
        body.withClient("");
        body.withUser("");
        request.withBody(body);
        try {
            UpdateKafkaUserClientQuotaTaskResponse response =
client.updateKafkaUserClientQuotaTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Modificação de cotas de usuários ou clientes

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateKafkaUserClientQuotaTaskRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateKafkaUserClientQuotaTaskReq(
            consumer_byte_rate=2097152,
            producer_byte_rate=3145728,
            client_default=True,
            user_default=False,
            client="",
            user=""
        )
        response = client.update_kafka_user_client_quota_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

Go

Modificação de cotas de usuários ou clientes

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateKafkaUserClientQuotaTaskRequest{}
    request.InstanceId = "{instance_id}"
    consumerByteRateUpdateKafkaUserClientQuotaTaskReq:= int64(2097152)
    producerByteRateUpdateKafkaUserClientQuotaTaskReq:= int64(3145728)
    clientDefaultUpdateKafkaUserClientQuotaTaskReq:= true
    userDefaultUpdateKafkaUserClientQuotaTaskReq:= false
    clientUpdateKafkaUserClientQuotaTaskReq:= ""
    userUpdateKafkaUserClientQuotaTaskReq:= ""
    request.Body = &model.UpdateKafkaUserClientQuotaTaskReq{
        ConsumerByteRate: &consumerByteRateUpdateKafkaUserClientQuotaTaskReq,
        ProducerByteRate: &producerByteRateUpdateKafkaUserClientQuotaTaskReq,
        ClientDefault: &clientDefaultUpdateKafkaUserClientQuotaTaskReq,
        UserDefault: &userDefaultUpdateKafkaUserClientQuotaTaskReq,
        Client: &clientUpdateKafkaUserClientQuotaTaskReq,
        User: &userUpdateKafkaUserClientQuotaTaskReq,
    }
    response, err := client.UpdateKafkaUserClientQuotaTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Cotas de usuário ou cliente modificadas.

Códigos de erro

Consulte [Códigos de erro](#).

5.2.21 Redefinição do deslocamento do grupo de consumidores para a posição especificada

Função

As instâncias do Kafka não oferecem suporte à redefinição on-line do deslocamento do consumidor. Antes de redefinir, pare o cliente para o qual o deslocamento deve ser redefinido. Depois que um cliente é interrompido, o servidor considera o cliente off-line somente após o período especificado em `ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG` (1000 ms por padrão).

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/kafka/{project_id}/instances/{instance_id}/groups/{group}/reset-message-offset

Tabela 5-94 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto.
instance_id	Sim	String	ID da instância.
group	Sim	String	Nome do grupo de consumidores.

Parâmetros de solicitação

Tabela 5-95 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
topic	Não	String	Nome do tópico.
partition	Sim	Integer	Número da partição. O valor padrão é -1 , indicando que todas as partições são redefinidas.
message_offset	Não	Long	<p>Redefinição do deslocamento do grupo de consumidores para a posição especificada.</p> <ul style="list-style-type: none"> ● Se essa posição for anterior ao deslocamento mais anterior atual, o deslocamento será redefinido para o deslocamento mais anterior. ● Se esse deslocamento for posterior ao maior deslocamento atual, o deslocamento será redefinido para o deslocamento mais recente. <p>Deve ser especificado message_offset ou timestamp.</p>
timestamp	Não	Long	<p>Hora especificada para a qual o deslocamento deve ser redefinido. O valor é um carimbo de data/hora UNIX, em milissegundos.</p> <ul style="list-style-type: none"> ● Se essa hora for anterior ao registro de data e hora mais anterior atual, o deslocamento será redefinido para o carimbo de data/hora mais anterior. ● Se esse tempo for posterior ao maior carimbo de data/hora atual, o deslocamento será redefinido para o carimbo de data/hora mais recente. <p>Deve ser especificado message_offset ou timestamp.</p>

Parâmetros de resposta

Nenhum

Exemplo de solicitações

- Redefinição do deslocamento do grupo de consumidores para a posição especificada

```
POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/groups/{group}/reset-message-offset

{
  "topic" : "test",
  "partition" : 0,
  "message_offset" : 10
}
```

- Redefinição do deslocamento do grupo de consumidores para o horário especificado

```
POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/groups/{group}/reset-message-offset

{
  "topic" : "test",
  "partition" : 0,
  "timestamp" : 1571812144000
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- Redefinição do deslocamento do grupo de consumidores para a posição especificada

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetWithEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    ResetMessageOffsetWithEngineRequest request = new
ResetMessageOffsetWithEngineRequest();
    request.withInstanceId("{instance_id}");
    request.withGroup("{group}");
    ResetMessageOffsetReq body = new ResetMessageOffsetReq();
    body.withMessageOffset(10L);
    body.withPartition(0);
    body.withTopic("test");
    request.withBody(body);
    try {
        ResetMessageOffsetWithEngineResponse response =
client.resetMessageOffsetWithEngine(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- Redefinição do deslocamento do grupo de consumidores para o horário especificado

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetWithEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ResetMessageOffsetWithEngineRequest request = new
ResetMessageOffsetWithEngineRequest();
        request.withInstanceId("{instance_id}");
        request.withGroup("{group}");
```

```
ResetMessageOffsetReq body = new ResetMessageOffsetReq();
body.withTimestamp(1571812144000L);
body.withPartition(0);
body.withTopic("test");
request.withBody(body);
try {
    ResetMessageOffsetWithEngineResponse response =
client.resetMessageOffsetWithEngine(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Redefinição do deslocamento do grupo de consumidores para a posição especificada

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetMessageOffsetWithEngineRequest()
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        request.body = ResetMessageOffsetReq(
            message_offset=10,
            partition=0,
            topic="test"
        )
        response = client.reset_message_offset_with_engine(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Redefinição do deslocamento do grupo de consumidores para o horário especificado

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and
    # SK be stored in ciphertext in configuration files or environment variables
    # and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR_REGION>")) \
        .build()

    try:
        request = ResetMessageOffsetWithEngineRequest()
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        request.body = ResetMessageOffsetReq(
            timestamp=1571812144000,
            partition=0,
            topic="test"
        )
        response = client.reset_message_offset_with_engine(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Redefinição do deslocamento do grupo de consumidores para a posição especificada

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and
    // SK be stored in ciphertext in configuration files or environment variables
    // and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResetMessageOffsetWithEngineRequest{
    request.InstanceId = "{instance_id}"
    request.Group = "{group}"
    messageOffsetResetMessageOffsetReq:= int64(10)
    topicResetMessageOffsetReq:= "test"
    request.Body = &model.ResetMessageOffsetReq{
        MessageOffset: &messageOffsetResetMessageOffsetReq,
        Partition: int32(0),
        Topic: &topicResetMessageOffsetReq,
    }
}
response, err := client.ResetMessageOffsetWithEngine(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- **Redefinição do deslocamento do grupo de consumidores para o horário especificado**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetMessageOffsetWithEngineRequest{
        request.InstanceId = "{instance_id}"
```

```
request.Group = "{group}"
timestampResetMessageOffsetReq:= int64(1571812144000)
topicResetMessageOffsetReq:= "test"
request.Body = &model.ResetMessageOffsetReq{
    Timestamp: &timestampResetMessageOffsetReq,
    Partition: int32(0),
    Topic: &topicResetMessageOffsetReq,
}
response, err := client.ResetMessageOffsetWithEngine(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Bem-sucedido

Códigos de erro

Consulte [Códigos de erro](#).

5.3 Smart Connect

5.3.1 Ativação do Smart Connect (instância de pagamento por uso)

Função

Essa API é usada para ativar o Smart Connect para que você possa criar um conector.

Atualmente, essa API só pode ser usada para criar conectores para instâncias de pagamento por uso.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/connector

Tabela 5-96 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-97 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
specification	Não	String	Largura de banda para a implementação do Smart Connect, ou seja, a quantidade máxima de dados transferidos por unidade de tempo. Use a largura de banda da instância atual. Este parâmetro é obrigatório apenas para flavors de instância anteriores. Opções: <ul style="list-style-type: none"> ● 100 MB ● 300 MB ● 600 MB ● 1200 MB
node_cnt	Não	String	Número de conectores. Mínimo: 2. O valor padrão é 2 se não for especificado.
spec_code	Não	String	Código de especificação do conector. Este parâmetro é obrigatório apenas para flavors de instância anteriores.

Parâmetros de resposta

Código de status: 200

Tabela 5-98 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa.
connector_id	String	ID de despejo da instância.

Exemplo de solicitações

- Para ativar o Smart Connect para instâncias de pagamento por uso usando novos flavors, defina o número de conectores como 2.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector

{
  "node_cnt" : 2
}
```

- Para ativar o Smart Connect para instâncias de pagamento por uso usando flavors anteriores, defina o tamanho dos conectores como 100 MB e o número deles como 2.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector

{
  "specification" : "100MB",
  "node_cnt" : 2,
  "spec_code" : "kafka.c3.mini.connector"
}
```

Exemplo de respostas

Código de status: 200

Bem-sucedido.

```
{
  "job_id" : "7c3ec20c-11de-4df9-acc0-7ef1dea25dfe",
  "connector_id" : "55b78880-9077-4c74-ad5a-6868555f76a4"
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- Para ativar o Smart Connect para instâncias de pagamento por uso usando novos flavors, defina o número de conectores como 2.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```



```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

CreateConnectorRequest request = new CreateConnectorRequest();
request.withInstanceId("{instance_id}");
CreateConnectorReq body = new CreateConnectorReq();
body.withNodeCnt("2");
request.withBody(body);
try {
    CreateConnectorResponse response =
client.createConnector(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Para ativar o Smart Connect para instâncias de pagamento por uso usando flavors anteriores, defina o tamanho dos conectores como 100 MB e o número deles como 2.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateConnectorRequest request = new CreateConnectorRequest();
```

```
        request.withInstanceId("{instance_id}");
        CreateConnectorReq body = new CreateConnectorReq();
        body.withSpecCode("kafka.c3.mini.connector");
        body.withNodeCnt("2");

body.withSpecification(CreateConnectorReq.SpecificationEnum.fromValue("100MB")
);
        request.withBody(body);
        try {
            CreateConnectorResponse response =
client.createConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Para ativar o Smart Connect para instâncias de pagamento por uso usando novos flavors, defina o número de conectores como 2.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateConnectorReq(
            node_cnt="2"
        )
        response = client.create_connector(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Para ativar o Smart Connect para instâncias de pagamento por uso usando flavors anteriores, defina o tamanho dos conectores como 100 MB e o número deles como 2.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateConnectorReq(
            spec_code="kafka.c3.mini.connector",
            node_cnt="2",
            specification="100MB"
        )
        response = client.create_connector(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Para ativar o Smart Connect para instâncias de pagamento por uso usando novos flavors, defina o número de conectores como 2.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateConnectorRequest{
    request.InstanceId = "{instance_id}"
    nodeCntCreateConnectorReq:= "2"
    request.Body = &model.CreateConnectorReq{
        NodeCnt: &nodeCntCreateConnectorReq,
    }
}
response, err := client.CreateConnector(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Para ativar o Smart Connect para instâncias de pagamento por uso usando flavors anteriores, defina o tamanho dos conectores como 100 MB e o número deles como 2.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorRequest{
        request.InstanceId = "{instance_id}"
        specCodeCreateConnectorReq:= "kafka.c3.mini.connector"
        nodeCntCreateConnectorReq:= "2"
    }
```

```
specificationCreateConnectorReq:=
model.GetCreateConnectorReqSpecificationEnum().E_100_MB
request.Body = &model.CreateConnectorReq{
    SpecCode: &specCodeCreateConnectorReq,
    NodeCnt: &nodeCntCreateConnectorReq,
    Specification: &specificationCreateConnectorReq,
}
response, err := client.CreateConnector(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.2 Desativação do Smart Connect (instância de pagamento por uso)

Função

Essa API é usada para desativar o Smart Connect para uma instância de pagamento por uso.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/delete-connector

Tabela 5-99 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-100 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID do trabalho para executar de forma assíncrona a tarefa de exclusão.

Exemplo de solicitações

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/delete-connector
```

Exemplo de respostas

Código de status: 200

A tarefa de desativação do Smart Connect foi enviada com sucesso.

```
{  
  "job_id" : "d366178c-29ea-4d5c-a344-fa2ece4a1836"  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;
```

```
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        DeleteConnectorRequest request = new DeleteConnectorRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeleteConnectorResponse response = client.deleteConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
```

```
.with_credentials(credentials) \
.with_region(KafkaRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DeleteConnectorRequest()
    request.instance_id = "{instance_id}"
    response = client.delete_connector(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteConnectorRequest{
        request.InstanceId = "{instance_id}"
    }
    response, err := client.DeleteConnector(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A tarefa de desativação do Smart Connect foi enviada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.3 Criação de uma tarefa do Smart Connect

Função

Essa API é usada para criar uma tarefa do Smart Connect.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/connector/tasks

Tabela 5-101 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-102 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
task_name	Não	String	Nome da tarefa do Smart Connect.
start_later	Não	Boolean	Indica se a tarefa deve ser iniciada mais tarde. false : para criar uma tarefa e iniciá-la imediatamente; true : para criar uma tarefa e iniciá-la manualmente mais tarde na lista de tarefas.

Parâmetro	Obrigatório	Tipo	Descrição
topics	Não	String	Tópico de uma tarefa do Smart Connect.
topics_regex	Não	String	Expressão regular do tópico de uma tarefa do Smart Connect.
source_type	Não	String	Tipo de origem de uma tarefa do Smart Connect.
source_task	Não	SmartConnectTaskReqSourceConfig object	Configuração de origem de uma tarefa do Smart Connect.
sink_type	Não	String	Tipo de destino de uma tarefa do Smart Connect.
sink_task	Não	SmartConnectTaskReqSinkConfig object	Tipo de destino de uma tarefa do Smart Connect.

Tabela 5-103 SmartConnectTaskReqSourceConfig

Parâmetro	Obrigatório	Tipo	Descrição
redis_address	Não	String	Endereço da instância do Redis. (Obrigatório somente quando o tipo de origem for Redis.)
redis_type	Não	String	Tipo de instância do Redis. (Obrigatório somente quando o tipo de origem for Redis.)
dc_instance_id	Não	String	ID da instância do DCS. (Obrigatório somente quando o tipo de origem for Redis.)
redis_password	Não	String	Senha do Redis. (Obrigatório somente quando o tipo de origem for Redis.)
sync_mode	Não	String	Tipo de sincronização: RDB_ONLY indica sincronização completa; CUSTOM_OFFSET indica sincronização completa e incremental. (Obrigatório somente quando o tipo de origem for Redis.)
full_sync_wait_ms	Não	Integer	Intervalo de novas tentativas de sincronização total, em ms. (Obrigatório somente quando o tipo de origem for Redis.)

Parâmetro	Obrigatório	Tipo	Descrição
full_sync_max_retry	Não	Integer	Máximo de novas tentativas de sincronização total. (Obrigatório somente quando o tipo de origem for Redis.)
ratelimit	Não	Integer	Limite de taxa, em KB/s. -1: não há limite de taxa. (Obrigatório somente quando o tipo de origem for Redis.)
current_cluster_name	Não	String	Nome da instância atual do Kafka. (Obrigatório somente quando o tipo de origem for Kafka.)
cluster_name	Não	String	Nome da instância do Kafka de destino. (Obrigatório somente quando o tipo de origem for Kafka.)
user_name	Não	String	Nome de usuário definido quando SASL_SSL foi ativado para uma instância do Kafka de destino ou quando um usuário do SASL_SSL foi criado. (Obrigatório somente quando o tipo de origem for Kafka e o modo de autenticação do Kafka de destino for SASL_SSL.)
password	Não	String	Senha definida quando o SASL_SSL foi ativado para uma instância de destino do Kafka ou quando um usuário do SASL_SSL foi criado. (Obrigatório somente quando o tipo de origem for Kafka e o modo de autenticação do Kafka de destino for SASL_SSL.)
sasl_mechanism	Não	String	Modo de autenticação do Kafka de destino. (Obrigatório somente quando o tipo de origem é Kafka e o modo de autenticação é SASL_SSL.)
instance_id	Não	String	ID de instância do Kafka de destino. (Obrigatório somente quando o tipo de origem for Kafka. Especifique o instance_id ou bootstrap_servers .)

Parâmetro	Obrigatório	Tipo	Descrição
bootstrap_servers	Não	String	Endereço de instância de Kafka de destino. (Obrigatório somente quando o tipo de origem for Kafka. Especifique o instance_id ou bootstrap_servers .)
security_protocol	Não	String	Autenticação do Kafka de destino. (Obrigatório somente quando o tipo de origem é Kafka). Existem dois modos de autenticação: <ul style="list-style-type: none"> ● SASL_SSL: SASL_SSL está ativado. ● PLAINTEXT: SASL_SSL está desativado.
direction	Não	String	Direção de sincronização: pull replica dados da instância do Kafka de destino para a atual; push replica dados da instância do Kafka de origem para a de destino; two-way replica dados da instância do Kafka de origem e de destino entre si. (Obrigatório somente quando o tipo de origem for Kafka.)
sync_consumer_offsets_enabled	Não	Boolean	Indica se o progresso do consumo deve ser sincronizado. (Obrigatório somente quando o tipo de origem for Kafka.)
replication_factor	Não	Integer	Número de réplicas de tópico quando um tópico é criado automaticamente na instância de par. O valor deste parâmetro não pode exceder o número de corretores na instância de par. Esse parâmetro substitui o parâmetro default.replication.factor configurado na instância de destino. (Obrigatório somente quando o tipo de origem for Kafka.)

Parâmetro	Obrigatório	Tipo	Descrição
task_num	Não	Integer	Número de tarefas de replicação de dados. O valor padrão é 2 . É aconselhável usar o valor padrão. Se a direção de sincronização for definida como bidirecional, o número real de tarefas será o dobro do número de tarefas configuradas aqui. (Obrigatório somente quando o tipo de origem for Kafka.)
rename_topic_enabled	Não	Boolean	Indica se um tópico deve ser renomeado. Se sim, adicione o alias da instância do Kafka de origem antes do nome do tópico de destino para formar um novo nome do tópico de destino. (Obrigatório somente quando o tipo de origem for Kafka.)
provenance_header_enabled	Não	Boolean	O tópico de destino recebe as mensagens replicadas. O cabeçalho da mensagem contém a origem da mensagem. Se você selecionar Both para Sync Direction , ative Add Source Header para impedir a replicação infinita. (Obrigatório somente quando o tipo de origem for Kafka.)
consumer_strategy	Não	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Obrigatório somente quando o tipo de origem for Kafka.)
compression_type	Não	String	Algoritmo de compactação a ser usado para copiar mensagens. (Obrigatório somente quando o tipo de origem for Kafka.) <ul style="list-style-type: none"> ● none ● gzip ● snappy ● lz4 ● zstd

Parâmetro	Obrigatório	Tipo	Descrição
topics_mapping	Não	String	Mapeamento de tópico, que é usado para personalizar o nome do tópico de destino. Rename Topic e Topic Mapping não podem ser configurados ao mesmo tempo. Formato de mapeamento de tópicos: <i>source topic:target topic</i> . Use vírgulas (,) para separar vários mapeamentos de tópicos, por exemplo, topic-sc-1:topic-sc-2,topic-sc-3:topic-sc-4 . (Obrigatório somente quando o tipo de origem for Kafka.)

Tabela 5-104 SmartConnectTaskReqSinkConfig

Parâmetro	Obrigatório	Tipo	Descrição
redis_address	Não	String	Endereço da instância do Redis. (Obrigatório somente quando o tipo de destino for Redis.)
redis_type	Não	String	Tipo de instância do Redis. (Obrigatório somente quando o tipo de destino for Redis.)
dcs_instance_id	Não	String	ID da instância do DCS. (Obrigatório somente quando o tipo de destino for Redis.)
redis_password	Não	String	Senha do Redis. (Obrigatório somente quando o tipo de destino for Redis.)
consumer_strategy	Não	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Obrigatório somente quando o tipo de destino for OBS.)
destination_file_type	Não	String	Formato do arquivo de despejo. Somente TEXT é suportado. (Obrigatório somente quando o tipo de destino for OBS.)
deliver_time_interval	Não	Integer	Intervalos de despejo de dados, em segundos. O intervalo padrão é 300s. (Obrigatório somente quando o tipo de destino for OBS.)

Parâmetro	Obrigatório	Tipo	Descrição
access_key	Não	String	AK: ID da chave de acesso. (Obrigatório somente quando o tipo de destino for OBS.)
secret_key	Não	String	SK: chave de acesso secreta usada junto com o ID da chave de acesso. (Obrigatório somente quando o tipo de destino for OBS.)
obs_bucket_name	Não	String	Endereço de despejo, que é o bucket do OBS usado para armazenar os dados do tópico. (Obrigatório somente quando o tipo de destino for OBS.)
obs_path	Não	String	Diretório de despejo, que é o diretório para armazenar arquivos de tópicos despejados no OBS. Use barras (/) para separar os níveis de diretório. (Obrigatório somente quando o tipo de destino for OBS.)
partition_format	Não	String	Formato do diretório de hora. (Obrigatório somente quando o tipo de destino for OBS.) <ul style="list-style-type: none"> ● yyyy: ano. ● yyyy/MM: ano e mês. ● yyyy/MM/dd: ano, mês e dia. ● yyyy/MM/dd/HH: ano, mês, dia e hora. ● yyyy/MM/dd/HH/mm: ano, mês, dia, hora e minuto.
record_delimiter	Não	String	Quebra de linha, que é usada para separar os dados do usuário que são gravados no arquivo de despejo. (Obrigatório somente quando o tipo de destino for OBS.) Intervalo de valores: <ul style="list-style-type: none"> ● Comma (,) ● Semicolon (;) ● Vertical bar () ● Newline (\n) ● NULL
store_keys	Não	Boolean	Especifica se as chaves devem ser despejadas. (Obrigatório somente quando o tipo de destino for OBS.)

Parâmetros de resposta

Código de status: 200

Tabela 5-105 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
task_name	String	Nome da tarefa do Smart Connect.
topics	String	Tópico de uma tarefa do Smart Connect.
topics_regex	String	Expressão regular do tópico de uma tarefa do Smart Connect.
source_type	String	Tipo de origem de uma tarefa do Smart Connect.
source_task	SmartConnectTaskRespSourceConfig object	Configuração de origem de uma tarefa do Smart Connect.
sink_type	String	Tipo de destino de uma tarefa do Smart Connect.
sink_task	SmartConnectTaskRespSinkConfig object	Tipo de destino de uma tarefa do Smart Connect.
id	String	ID de uma tarefa do Smart Connect.
status	String	Status da tarefa do Smart Connect.
create_time	Long	Hora em que a tarefa do Smart Connect foi criada.

Tabela 5-106 SmartConnectTaskRespSourceConfig

Parâmetro	Tipo	Descrição
redis_address	String	Endereço da instância do Redis. (Exibido somente quando o tipo de origem é Redis.)
redis_type	String	Tipo de instância do Redis. (Exibido somente quando o tipo de origem é Redis.)
dcs_instance_id	String	ID da instância do DCS. (Exibido somente quando o tipo de origem é Redis.)
sync_mode	String	Tipo de sincronização: RDB_ONLY indica sincronização completa; CUSTOM_OFFSET indica sincronização completa e incremental. (Exibido somente quando o tipo de origem é Redis.)

Parâmetro	Tipo	Descrição
full_sync_wait_ms	Integer	Intervalo de novas tentativas de sincronização total, em ms. (Exibido somente quando o tipo de origem é Redis.)
full_sync_max_retry	Integer	Máximo de novas tentativas de sincronização total. (Exibido somente quando o tipo de origem é Redis.)
ratelimit	Integer	Limite de taxa, em KB/s. -1 : desativar. (Exibido somente quando o tipo de origem é Redis.)
current_cluster_name	String	Nome da instância atual do Kafka. (Exibido somente quando o tipo de origem é Kafka.)
cluster_name	String	Nome da instância do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
user_name	String	Nome de usuário da instância de Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
sasl_mechanism	String	Modo de autenticação do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
instance_id	String	ID de instância do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
bootstrap_servers	String	Endereço de instância de Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
security_protocol	String	Autenticação do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
direction	String	Direção de sincronização. (Exibido somente quando o tipo de origem é Kafka.)
sync_consumer_offsets_enabled	Boolean	Indica se o progresso do consumo deve ser sincronizado. (Exibido somente quando o tipo de origem é Kafka.)
replication_factor	Integer	Número de réplicas. (Exibido somente quando o tipo de origem é Kafka.)
task_num	Integer	Número de tarefas. (Exibido somente quando o tipo de origem é Kafka.)
rename_topic_enabled	Boolean	Indica se um tópico deve ser renomeado. (Exibido somente quando o tipo de origem é Kafka.)
provenance_header_enabled	Boolean	Indica se o cabeçalho de origem deve ser adicionado. (Exibido somente quando o tipo de origem é Kafka.)
consumer_strategy	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Exibido somente quando o tipo de origem é Kafka.)
compression_type	String	Algoritmo de compactação. (Exibido somente quando o tipo de origem é Kafka.)

Parâmetro	Tipo	Descrição
topics_mapping	String	Mapeamento de tópicos. (Exibido somente quando o tipo de origem é Kafka.)

Tabela 5-107 SmartConnectTaskRespSinkConfig

Parâmetro	Tipo	Descrição
redis_address	String	Endereço da instância do Redis. (Exibido somente quando o tipo de destino é Redis.)
redis_type	String	Tipo de instância do Redis. (Exibido somente quando o tipo de destino é Redis.)
dcs_instance_id	String	ID da instância do DCS. (Exibido somente quando o tipo de destino é Redis.)
target_db	Integer	Banco de dados de destino. O valor padrão é -1 . (Exibido somente quando o tipo de destino é Redis.)
consumer_strategy	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Exibido somente quando o tipo de destino é OBS.)
destination_file_type	String	Formato do arquivo de despejo. Somente TEXT é suportado. (Exibido somente quando o tipo de destino é OBS.)
deliver_time_interval	Integer	Período(s) de despejo. (Exibido somente quando o tipo de destino é OBS.)
obs_bucket_name	String	Endereço de despejo. (Exibido somente quando o tipo de destino é OBS.)
obs_path	String	Diretório de despejo. (Exibido somente quando o tipo de destino é OBS.)
partition_format	String	Formato do diretório de hora. (Exibido somente quando o tipo de destino é OBS.)
record_delimiter	String	Quebra de linha. (Exibido somente quando o tipo de destino é OBS.)
store_keys	Boolean	Key de armazenamento. (Exibido somente quando o tipo de destino é OBS.)
obs_part_size	Integer	Tamanho (em bytes) de cada arquivo a ser carregado. O valor padrão é 5242880 . (Exibido somente quando o tipo de destino é OBS.)
flush_size	Integer	flush_size. (Exibido somente quando o tipo de destino é OBS.)

Parâmetro	Tipo	Descrição
timezone	String	Fuso horário. (Exibido somente quando o tipo de destino é OBS.)
connector_class	String	Classe de connector. O valor padrão é com.huawei.dms.connector.obs.OBSSinkConnector . (Exibido somente quando o tipo de destino é OBS.)
storage_class	String	Classe de storage. O valor padrão é com.huawei.dms.connector.obs.storage.OBSStorage . (Exibido somente quando o tipo de destino é OBS.)
format_class	String	Classe de format. O valor padrão é com.huawei.dms.connector.obs.format.bytearray.ByteArrayFormat . (Exibido somente quando o tipo de destino é OBS.)
schema_generator_class	String	Classe de schema_generator. O valor padrão é io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator . (Exibido somente quando o tipo de destino é OBS.)
partitioners	String	Classe de partitioner. O valor padrão é io.confluent.connect.storage.partitioners.TimeBasedPartitioner . (Exibido somente quando o tipo de destino é OBS.)
value_converter	String	value_converter. O valor padrão é org.apache.kafka.connect.converters.ByteArrayConverter . (Exibido somente quando o tipo de destino é OBS.)
key_converter	String	key_converter. O valor padrão é org.apache.kafka.connect.converters.ByteArrayConverter . (Exibido somente quando o tipo de destino é OBS.)
kv_delimiter	String	kv_delimiter. O valor padrão é . . (Exibido somente quando o tipo de destino é OBS.)

Exemplo de solicitações

- Criação de uma tarefa de despejo que começa imediatamente.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks
```

```
{
  "task_name" : "smart-connect-1",
  "start_later" : false,
  "source_type" : "NONE",
  "topics_regex" : "topic-obs*",
  "sink_type" : "OBS_SINK",
  "sink_task" : {
    "consumer_strategy" : "earliest",
    "destination_file_type" : "TEXT",
    "deliver_time_interval" : 300,
  }
}
```

```

"access_key" : "*****",
"secret_key" : "*****",
"obs_bucket_name" : "obs_bucket",
"obs_path" : "obsTransfer-1810125534",
"partition_format" : "yyyy/MM/dd/HH/mm",
"record_delimiter" : "\\n",
"store_keys" : false
}
}

```

- Criação de uma tarefa de replicação de dados do Kafka que é iniciada mais tarde.

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks

```

{
  "task_name" : "smart-connect-2",
  "start_later" : true,
  "source_type" : "KAFKA_REPLICATOR_SOURCE",
  "source_task" : {
    "current_cluster_name" : "A",
    "cluster_name" : "B",
    "user_name" : "user1",
    "password" : "*****",
    "sasl_mechanism" : "SCRAM-SHA-512",
    "instance_id" : "b54c9dd8-*****-*****",
    "direction" : "two-way",
    "sync_consumer_offsets_enabled" : false,
    "replication_factor" : 3,
    "task_num" : 2,
    "rename_topic_enabled" : false,
    "provenance_header_enabled" : true,
    "consumer_strategy" : "latest",
    "compression_type" : "snappy",
    "topics_mapping" : "topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4"
  }
}

```

- Criação de uma tarefa de replicação de dados do Redis que começa imediatamente: sincronização total, 10 tempos máximos de repetição, intervalo de repetição de 10.000 ms e limite de largura de banda de 10 KB/s.

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks

```

{
  "task_name" : "smart-connect-3",
  "start_later" : false,
  "source_type" : "REDIS_REPLICATOR_SOURCE",
  "source_task" : {
    "redis_address" :
"192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379",
    "redis_type" : "cluster",
    "redis_password" : "*****",
    "sync_mode" : "RDB_ONLY",
    "full_sync_max_retry" : 10,
    "full_sync_wait_ms" : 10000,
    "ratelimit" : 10
  },
  "topics" : "topic-sc-3",
  "sink_type" : "REDIS_REPLICATOR_SINK",
  "sink_task" : {
    "redis_address" : "192.168.119.51:6379",
    "redis_type" : "standalone",
    "redis_password" : "*****"
  }
}

```

Exemplo de respostas

Código de status: 200

Bem-sucedido.

```
{
  "task_name" : "smart-connect-121248117",
  "topics" : "topic-sc",
  "source_task" : {
    "redis_address" : "192.168.91.179:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "949190a2-598a-4afd-99a8-dad3caele7cd",
    "sync_mode" : "RDB_ONLY",
    "full_sync_wait_ms" : 13000,
    "full_sync_max_retry" : 4,
    "ratelimit" : -1
  },
  "source_type" : "REDIS_REPLICATOR_SOURCE",
  "sink_task" : {
    "redis_address" : "192.168.119.51:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "9b981368-a8e3-416a-87d9-1581a968b41b",
    "target_db" : -1
  },
  "sink_type" : "REDIS_REPLICATOR_SINK",
  "id" : "8a205bbd-7181-4b5e-9bd6-37274ce84577",
  "status" : "RUNNING",
  "create_time" : 1708427753133
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- Criação de uma tarefa de despejo que começa imediatamente.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
```

```
        .build();
        CreateConnectorTaskRequest request = new CreateConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        CreateSmartConnectTaskReq body = new CreateSmartConnectTaskReq();
        SmartConnectTaskReqSinkConfig sinkTaskbody = new
SmartConnectTaskReqSinkConfig();
        sinkTaskbody.withConsumerStrategy("earliest")
            .withDestinationFileType("TEXT")
            .withDeliverTimeInterval(300)
            .withAccessKey("*****")
            .withSecretKey("*****")
            .withObsBucketName("obs_bucket")
            .withObsPath("obsTransfer-1810125534")
            .withPartitionFormat("yyyy/MM/dd/HH/mm")
            .withRecordDelimiter("\n")
            .withStoreKeys(false);
        body.withSinkTask(sinkTaskbody);

body.withSinkType(CreateSmartConnectTaskReq.SinkTypeEnum.fromValue("OBS_SINK")
);

body.withSourceType(CreateSmartConnectTaskReq.SourceTypeEnum.fromValue("NONE")
);

        body.withTopicsRegex("topic-obs*");
        body.withStartLater(false);
        body.withTaskName("smart-connect-1");
        request.withBody(body);
        try {
            CreateConnectorTaskResponse response =
client.createConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Criação de uma tarefa de replicação de dados do Kafka que é iniciada mais tarde.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

CreateConnectorTaskRequest request = new CreateConnectorTaskRequest();
request.withInstanceId("{instance_id}");
CreateSmartConnectTaskReq body = new CreateSmartConnectTaskReq();
SmartConnectTaskReqSourceConfig sourceTaskbody = new
SmartConnectTaskReqSourceConfig();
sourceTaskbody.withCurrentClusterName("A")
    .withClusterName("B")
    .withUserName("user1")
    .withPassword("*****")
    .withSaslMechanism("SCRAM-SHA-512")
    .withInstanceId("b54c9dd8-*****-*****")
    .withDirection("two-way")
    .withSyncConsumerOffsetsEnabled(false)
    .withReplicationFactor(3)
    .withTaskNum(2)
    .withRenameTopicEnabled(false)
    .withProvenanceHeaderEnabled(true)
    .withConsumerStrategy("latest")
    .withCompressionType("snappy")
    .withTopicsMapping("topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4");
body.withSourceTask(sourceTaskbody);

body.withSourceType(CreateSmartConnectTaskReq.SourceTypeEnum.fromValue("KAFKA_
REPLICATOR_SOURCE"));
body.withStartLater(true);
body.withTaskName("smart-connect-2");
request.withBody(body);
try {
    CreateConnectorTaskResponse response =
client.createConnectorTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Criação de uma tarefa de replicação de dados do Redis que começa imediatamente: sincronização total, 10 tempos máximos de repetição, intervalo de repetição de 10.000 ms e limite de largura de banda de 10 KB/s.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class CreateConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and
        SK be stored in ciphertext in configuration files or environment variables
        and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateConnectorTaskRequest request = new CreateConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        CreateSmartConnectTaskReq body = new CreateSmartConnectTaskReq();
        SmartConnectTaskReqSinkConfig sinkTaskbody = new
SmartConnectTaskReqSinkConfig();
        sinkTaskbody.withRedisAddress("192.168.119.51:6379")
            .withRedisType("standalone")
            .withRedisPassword("*****");
        SmartConnectTaskReqSourceConfig sourceTaskbody = new
SmartConnectTaskReqSourceConfig();
        sourceTaskbody.withRedisAddress("192.168.99.249:6379,192.168.120.127:6379,192.
168.116.219:6379")
            .withRedisType("cluster")
            .withRedisPassword("*****")
            .withSyncMode("RDB_ONLY")
            .withFullSyncWaitMs(10000)
            .withFullSyncMaxRetry(10)
            .withRatelimit(10);
        body.withSinkTask(sinkTaskbody);

        body.withSinkType(CreateSmartConnectTaskReq.SinkTypeEnum.fromValue("REDIS_REPL
ICATOR_SINK"));
        body.withSourceTask(sourceTaskbody);

        body.withSourceType(CreateSmartConnectTaskReq.SourceTypeEnum.fromValue("REDIS_
REPLICATOR_SOURCE"));
        body.withTopics("topic-sc-3");
        body.withStartLater(false);
        body.withTaskName("smart-connect-3");
        request.withBody(body);
        try {
            CreateConnectorTaskResponse response =
client.createConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```



```
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- Criação de uma tarefa de despejo que começa imediatamente.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and
    # SK be stored in ciphertext in configuration files or environment variables
    # and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        sinkTaskbody = SmartConnectTaskReqSinkConfig(
            consumer_strategy="earliest",
            destination_file_type="TEXT",
            deliver_time_interval=300,
            access_key="*****",
            secret_key="*****",
            obs_bucket_name="obs_bucket",
            obs_path="obsTransfer-1810125534",
            partition_format="yyyy/MM/dd/HH/mm",
            record_delimiter="\n",
            store_keys=False
        )
        request.body = CreateSmartConnectTaskReq(
            sink_task=sinkTaskbody,
            sink_type="OBS_SINK",
            source_type="NONE",
            topics_regex="topic-obs*",
            start_later=False,
            task_name="smart-connect-1"
        )
        response = client.create_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Criação de uma tarefa de replicação de dados do Kafka que é iniciada mais tarde.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        sourceTaskbody = SmartConnectTaskReqSourceConfig(
            current_cluster_name="A",
            cluster_name="B",
            user_name="user1",
            password="*****",
            sasl_mechanism="SCRAM-SHA-512",
            instance_id="b54c9dd8-*****-*****",
            direction="two-way",
            sync_consumer_offsets_enabled=False,
            replication_factor=3,
            task_num=2,
            rename_topic_enabled=False,
            provenance_header_enabled=True,
            consumer_strategy="latest",
            compression_type="snappy",
            topics_mapping="topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4"
        )
        request.body = CreateSmartConnectTaskReq(
            source_task=sourceTaskbody,
            source_type="KAFKA_REPLICATOR_SOURCE",
            start_later=True,
            task_name="smart-connect-2"
        )
        response = client.create_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Criação de uma tarefa de replicação de dados do Redis que começa imediatamente: sincronização total, 10 tempos máximos de repetição, intervalo de repetição de 10.000 ms e limite de largura de banda de 10 KB/s.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateConnectorTaskRequest()
    request.instance_id = "{instance_id}"
    sinkTaskbody = SmartConnectTaskReqSinkConfig(
        redis_address="192.168.119.51:6379",
        redis_type="standalone",
        redis_password="*****"
    )
    sourceTaskbody = SmartConnectTaskReqSourceConfig(
redis_address="192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379",
        redis_type="cluster",
        redis_password="*****",
        sync_mode="RDB_ONLY",
        full_sync_wait_ms=10000,
        full_sync_max_retry=10,
        ratelimit=10
    )
    request.body = CreateSmartConnectTaskReq(
        sink_task=sinkTaskbody,
        sink_type="REDIS_REPLICATOR_SINK",
        source_task=sourceTaskbody,
        source_type="REDIS_REPLICATOR_SOURCE",
        topics="topic-sc-3",
        start_later=False,
        task_name="smart-connect-3"
    )
    response = client.create_connector_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Criação de uma tarefa de despejo que começa imediatamente.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
```

```
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateConnectorTaskRequest{}
request.InstanceId = "{instance_id}"
consumerStrategySinkTask:= "earliest"
destinationFileTypeSinkTask:= "TEXT"
deliverTimeIntervalSinkTask:= int32(300)
accessKeySinkTask:= "*****"
secretKeySinkTask:= "*****"
obsBucketNameSinkTask:= "obs_bucket"
obsPathSinkTask:= "obsTransfer-1810125534"
partitionFormatSinkTask:= "yyyy/MM/dd/HH/mm"
recordDelimiterSinkTask:= "\n"
storeKeysSinkTask:= false
sinkTaskbody := &model.SmartConnectTaskReqSinkConfig{
    ConsumerStrategy: &consumerStrategySinkTask,
    DestinationFileType: &destinationFileTypeSinkTask,
    DeliverTimeInterval: &deliverTimeIntervalSinkTask,
    AccessKey: &accessKeySinkTask,
    SecretKey: &secretKeySinkTask,
    ObsBucketName: &obsBucketNameSinkTask,
    ObsPath: &obsPathSinkTask,
    PartitionFormat: &partitionFormatSinkTask,
    RecordDelimiter: &recordDelimiterSinkTask,
    StoreKeys: &storeKeysSinkTask,
}
sinkTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSinkTypeEnum().OBS_SINK
sourceTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSourceTypeEnum().NONE
topicsRegexCreateSmartConnectTaskReq:= "topic-obs*"
startLaterCreateSmartConnectTaskReq:= false
taskNameCreateSmartConnectTaskReq:= "smart-connect-1"
request.Body = &model.CreateSmartConnectTaskReq{
    SinkTask: sinkTaskbody,
    SinkType: &sinkTypeCreateSmartConnectTaskReq,
    SourceType: &sourceTypeCreateSmartConnectTaskReq,
    TopicsRegex: &topicsRegexCreateSmartConnectTaskReq,
    StartLater: &startLaterCreateSmartConnectTaskReq,
    TaskName: &taskNameCreateSmartConnectTaskReq,
}
response, err := client.CreateConnectorTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Criação de uma tarefa de replicação de dados do Kafka que é iniciada mais tarde.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    currentClusterNameSourceTask := "A"
    clusterNameSourceTask := "B"
    userNameSourceTask := "user1"
    passwordSourceTask := "*****"
    saslMechanismSourceTask := "SCRAM-SHA-512"
    instanceIdSourceTask := "b54c9dd8-*****-*****"
    directionSourceTask := "two-way"
    syncConsumerOffsetsEnabledSourceTask := false
    replicationFactorSourceTask := int32(3)
    taskNumSourceTask := int32(2)
    renameTopicEnabledSourceTask := false
    provenanceHeaderEnabledSourceTask := true
    consumerStrategySourceTask := "latest"
    compressionTypeSourceTask := "snappy"
    topicsMappingSourceTask := "topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4"
    sourceTaskbody := &model.SmartConnectTaskReqSourceConfig{
        CurrentClusterName: &currentClusterNameSourceTask,
        ClusterName: &clusterNameSourceTask,
        UserName: &userNameSourceTask,
        Password: &passwordSourceTask,
        SaslMechanism: &saslMechanismSourceTask,
        InstanceId: &instanceIdSourceTask,
        Direction: &directionSourceTask,
        SyncConsumerOffsetsEnabled: &syncConsumerOffsetsEnabledSourceTask,
        ReplicationFactor: &replicationFactorSourceTask,
        TaskNum: &taskNumSourceTask,
        RenameTopicEnabled: &renameTopicEnabledSourceTask,
        ProvenanceHeaderEnabled: &provenanceHeaderEnabledSourceTask,
        ConsumerStrategy: &consumerStrategySourceTask,
        CompressionType: &compressionTypeSourceTask,
        TopicsMapping: &topicsMappingSourceTask,
    }
}
```

```
sourceTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSourceTypeEnum().KAFKA_REPLICATOR_SOURCE
startLaterCreateSmartConnectTaskReq:= true
taskNameCreateSmartConnectTaskReq:= "smart-connect-2"
request.Body = &model.CreateSmartConnectTaskReq{
    SourceTask: sourceTaskbody,
    SourceType: &sourceTypeCreateSmartConnectTaskReq,
    StartLater: &startLaterCreateSmartConnectTaskReq,
    TaskName: &taskNameCreateSmartConnectTaskReq,
}
response, err := client.CreateConnectorTask(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Criação de uma tarefa de replicação de dados do Redis que começa imediatamente: sincronização total, 10 tempos máximos de repetição, intervalo de repetição de 10.000 ms e limite de largura de banda de 10 KB/s.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    redisAddressSinkTask:= "192.168.119.51:6379"
    redisTypeSinkTask:= "standalone"
    redisPasswordSinkTask:= "*****"
    sinkTaskbody := &model.SmartConnectTaskReqSinkConfig{
        RedisAddress: &redisAddressSinkTask,
        RedisType: &redisTypeSinkTask,
        RedisPassword: &redisPasswordSinkTask,
    }
    redisAddressSourceTask:=
    "192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379"
    redisTypeSourceTask:= "cluster"
}
```

```

redisPasswordSourceTask:= "*****"
syncModeSourceTask:= "RDB_ONLY"
fullSyncWaitMsSourceTask:= int32(10000)
fullSyncMaxRetrySourceTask:= int32(10)
ratelimitSourceTask:= int32(10)
sourceTaskbody := &model.SmartConnectTaskReqSourceConfig{
    RedisAddress: &redisAddressSourceTask,
    RedisType: &redisTypeSourceTask,
    RedisPassword: &redisPasswordSourceTask,
    SyncMode: &syncModeSourceTask,
    FullSyncWaitMs: &fullSyncWaitMsSourceTask,
    FullSyncMaxRetry: &fullSyncMaxRetrySourceTask,
    Ratelimit: &ratelimitSourceTask,
}
sinkTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSinkTypeEnum().REDIS_REPLICATOR_SINK
sourceTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSourceTypeEnum().REDIS_REPLICATOR_SOURCE
topicsCreateSmartConnectTaskReq:= "topic-sc-3"
startLaterCreateSmartConnectTaskReq:= false
taskNameCreateSmartConnectTaskReq:= "smart-connect-3"
request.Body = &model.CreateSmartConnectTaskReq{
    SinkTask: sinkTaskbody,
    SinkType: &sinkTypeCreateSmartConnectTaskReq,
    SourceTask: sourceTaskbody,
    SourceType: &sourceTypeCreateSmartConnectTaskReq,
    Topics: &topicsCreateSmartConnectTaskReq,
    StartLater: &startLaterCreateSmartConnectTaskReq,
    TaskName: &taskNameCreateSmartConnectTaskReq,
}
response, err := client.CreateConnectorTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.4 Consulta de tarefas do Smart Connect

Função

Essa API é usada para consultar tarefas do Smart Connect.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/connector/tasks

Tabela 5-108 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-109 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
offset	Não	Integer	Deslocamento, que é a posição onde a consulta começa. O valor deve ser maior ou igual a 0. Padrão: 0
limit	Não	Integer	Número máximo de instâncias retornadas na consulta atual. O valor padrão é 10 . O valor varia de 1 a 50. Padrão: 10

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-110 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
tasks	Array of SmartConnect TaskEntity objects	Detalhes da tarefa do Smart Connect.

Parâmetro	Tipo	Descrição
total_number	Integer	Número de tarefas do Smart Connect.
max_tasks	Integer	Número máximo de tarefas do Smart Connect.
quota_tasks	Integer	Cota de tarefas do Smart Connect.

Tabela 5-111 SmartConnectTaskEntity

Parâmetro	Tipo	Descrição
task_name	String	Nome da tarefa do Smart Connect.
topics	String	Tópico de uma tarefa do Smart Connect.
topics_regex	String	Expressão regular do tópico de uma tarefa do Smart Connect.
source_type	String	Tipo de origem de uma tarefa do Smart Connect.
source_task	SmartConnectTaskRespSourceConfig object	Configuração de origem de uma tarefa do Smart Connect.
sink_type	String	Tipo de destino de uma tarefa do Smart Connect.
sink_task	SmartConnectTaskRespSinkConfig object	Tipo de destino de uma tarefa do Smart Connect.
id	String	ID de uma tarefa do Smart Connect.
status	String	Status da tarefa do Smart Connect.
create_time	Long	Hora em que a tarefa do Smart Connect foi criada.

Tabela 5-112 SmartConnectTaskRespSourceConfig

Parâmetro	Tipo	Descrição
redis_address	String	Endereço da instância do Redis. (Exibido somente quando o tipo de origem é Redis.)
redis_type	String	Tipo de instância do Redis. (Exibido somente quando o tipo de origem é Redis.)
dcs_instance_id	String	ID da instância do DCS. (Exibido somente quando o tipo de origem é Redis.)

Parâmetro	Tipo	Descrição
sync_mode	String	Tipo de sincronização: RDB_ONLY indica sincronização completa; CUSTOM_OFFSET indica sincronização completa e incremental. (Exibido somente quando o tipo de origem é Redis.)
full_sync_wait_ms	Integer	Intervalo de novas tentativas de sincronização total, em ms. (Exibido somente quando o tipo de origem é Redis.)
full_sync_max_retry	Integer	Máximo de novas tentativas de sincronização total. (Exibido somente quando o tipo de origem é Redis.)
ratelimit	Integer	Limite de taxa, em KB/s. -1 : desativar. (Exibido somente quando o tipo de origem é Redis.)
current_cluster_name	String	Nome da instância atual do Kafka. (Exibido somente quando o tipo de origem é Kafka.)
cluster_name	String	Nome da instância do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
user_name	String	Nome de usuário da instância de Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
sasl_mechanism	String	Modo de autenticação do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
instance_id	String	ID de instância do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
bootstrap_servers	String	Endereço de instância de Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
security_protocol	String	Autenticação do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
direction	String	Direção de sincronização. (Exibido somente quando o tipo de origem é Kafka.)
sync_consumer_offsets_enabled	Boolean	Indica se o progresso do consumo deve ser sincronizado. (Exibido somente quando o tipo de origem é Kafka.)
replication_factor	Integer	Número de réplicas. (Exibido somente quando o tipo de origem é Kafka.)
task_num	Integer	Número de tarefas. (Exibido somente quando o tipo de origem é Kafka.)
rename_topic_enabled	Boolean	Indica se um tópico deve ser renomeado. (Exibido somente quando o tipo de origem é Kafka.)
provenance_header_enabled	Boolean	Indica se o cabeçalho de origem deve ser adicionado. (Exibido somente quando o tipo de origem é Kafka.)

Parâmetro	Tipo	Descrição
consumer_strategy	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Exibido somente quando o tipo de origem é Kafka.)
compression_type	String	Algoritmo de compactação. (Exibido somente quando o tipo de origem é Kafka.)
topics_mapping	String	Mapeamento de tópicos. (Exibido somente quando o tipo de origem é Kafka.)

Tabela 5-113 SmartConnectTaskRespSinkConfig

Parâmetro	Tipo	Descrição
redis_address	String	Endereço da instância do Redis. (Exibido somente quando o tipo de destino é Redis.)
redis_type	String	Tipo de instância do Redis. (Exibido somente quando o tipo de destino é Redis.)
dcs_instance_id	String	ID da instância do DCS. (Exibido somente quando o tipo de destino é Redis.)
target_db	Integer	Banco de dados de destino. O valor padrão é -1 . (Exibido somente quando o tipo de destino é Redis.)
consumer_strategy	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Exibido somente quando o tipo de destino é OBS.)
destination_file_type	String	Formato do arquivo de despejo. Somente TEXT é suportado. (Exibido somente quando o tipo de destino é OBS.)
deliver_time_interval	Integer	Período(s) de despejo. (Exibido somente quando o tipo de destino é OBS.)
obs_bucket_name	String	Endereço de despejo. (Exibido somente quando o tipo de destino é OBS.)
obs_path	String	Diretório de despejo. (Exibido somente quando o tipo de destino é OBS.)
partition_format	String	Formato do diretório de hora. (Exibido somente quando o tipo de destino é OBS.)
record_delimiter	String	Quebra de linha. (Exibido somente quando o tipo de destino é OBS.)
store_keys	Boolean	Key de armazenamento. (Exibido somente quando o tipo de destino é OBS.)

Parâmetro	Tipo	Descrição
obs_part_size	Integer	Tamanho (em bytes) de cada arquivo a ser carregado. O valor padrão é 5242880 . (Exibido somente quando o tipo de destino é OBS.)
flush_size	Integer	flush_size. (Exibido somente quando o tipo de destino é OBS.)
timezone	String	Fuso horário. (Exibido somente quando o tipo de destino é OBS.)
connector_class	String	Classe de connector. O valor padrão é com.huawei.dms.connector.obs.OBSSinkConnector . (Exibido somente quando o tipo de destino é OBS.)
storage_class	String	Classe de storage. O valor padrão é com.huawei.dms.connector.obs.storage.OBSStorage . (Exibido somente quando o tipo de destino é OBS.)
format_class	String	Classe de format. O valor padrão é com.huawei.dms.connector.obs.format.bytearray.ByteArrayFormat . (Exibido somente quando o tipo de destino é OBS.)
schema_generator_class	String	Classe de schema_generator. O valor padrão é io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator . (Exibido somente quando o tipo de destino é OBS.)
partitioner_classes	String	Classe de partitioner. O valor padrão é io.confluent.connect.storage.partition器.TimeBasedPartitioner . (Exibido somente quando o tipo de destino é OBS.)
value_converter	String	value_converter. O valor padrão é org.apache.kafka.connect.converters.ByteArrayConverter . (Exibido somente quando o tipo de destino é OBS.)
key_converter	String	key_converter. O valor padrão é org.apache.kafka.connect.converters.ByteArrayConverter . (Exibido somente quando o tipo de destino é OBS.)
kv_delimiter	String	kv_delimiter. O valor padrão é : . (Exibido somente quando o tipo de destino é OBS.)

Exemplo de solicitações

Nenhum

Exemplo de respostas

Código de status: 200

Bem-sucedido.

```
{
  "tasks" : [ {
    "task_name" : "smart-connect-1571576841",
    "topics" : "topic-1643449744",
    "source_task" : {
      "current_cluster_name" : "A",
      "cluster_name" : "B",
      "direction" : "pull",
      "bootstrap_servers" :
"192.168.45.58:9092,192.168.44.1:9092,192.168.41.230:9092,192.168.43.112:9092",
      "instance_id" : "59f6d088-****-****-****-*****",
      "consumer_strategy" : "earliest",
      "sync_consumer_offsets_enabled" : false,
      "rename_topic_enabled" : false,
      "provenance_header_enabled" : false,
      "security_protocol" : "PLAINTEXT",
      "sasl_mechanism" : "PLAIN",
      "user_name" : "",
      "topics_mapping" : "",
      "compression_type" : "none",
      "task_num" : 2,
      "replication_factor" : 3
    },
    "source_type" : "KAFKA_REPLICATOR_SOURCE",
    "sink_task" : null,
    "sink_type" : "NONE",
    "id" : "194917d0-****-****-****-*****",
    "status" : "RUNNING",
    "create_time" : 1708427753133
  }, {
    "task_name" : "smart-connect-1",
    "topics_regex" : "topic-obs*",
    "source_task" : null,
    "source_type" : "NONE",
    "sink_task" : {
      "consumer_strategy" : "earliest",
      "destination_file_type" : "TEXT",
      "obs_bucket_name" : "abcabc",
      "obs_path" : "obsTransfer-1810125534",
      "partition_format" : "yyyy/MM/dd/HH/mm",
      "record_delimiter" : "\\n",
      "deliver_time_interva" : 300,
      "obs_part_size" : 5242880,
      "flush_size" : 1000000,
      "timezone" : "Asia/Chongqing",
      "connector_class" : "com.huawei.dms.connector.obs.OBSSinkConnector",
      "storage_class" : "com.huawei.dms.connector.obs.storage.OBSStorage",
      "format_class" :
"com.huawei.dms.connector.obs.format.bytearray.ByteArrayFormat",
      "schema_generator_class" :
"io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator",
      "partitioner_class" :
"io.confluent.connect.storage.partitionner.TimeBasedPartitionner",
      "value_converter" :
"org.apache.kafka.connect.converters.ByteArrayConverter",
      "key_converter" : "org.apache.kafka.connect.converters.ByteArrayConverter",
      "store_keys" : false,
      "kv_delimiter" : ":"
    },
    "sink_type" : "OBS_SINK",
    "id" : "3c0ac4d1-****-****-****-*****",
    "status" : "RUNNING",
    "create_time" : 1708565483911
  }, {
    "task_name" : "smart-connect-121248117",
    "topics" : "topic-sc",
    "source_task" : {
      "redis_address" : "192.168.91.179:6379",

```

```
"redis_type" : "standalone",
"dcs_instance_id" : "949190a2-598a-4afd-99a8-dad3cae1e7cd",
"sync_mode" : "RDB_ONLY",
"full_sync_wait_ms" : 13000,
"full_sync_max_retry" : 4,
"ratelimit" : -1
},
"source_type" : "REDIS_REPLICATOR_SOURCE",
"sink_task" : {
  "redis_address" : "192.168.119.51:6379",
  "redis_type" : "standalone",
  "dcs_instance_id" : "9b981368-a8e3-416a-87d9-1581a968b41b",
  "target_db" : -1
},
"sink_type" : "REDIS_REPLICATOR_SINK",
"id" : "8a205bbd-****-****-****-*****",
"status" : "RUNNING",
"create_time" : 1708427753133
} ],
"total_number" : 3,
"max_tasks" : 18,
"quota_tasks" : 18
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListConnectorTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListConnectorTasksRequest request = new ListConnectorTasksRequest();
        request.withInstanceId("{instance_id}");
        try {
```

```
        ListConnectorTasksResponse response =
client.listConnectorTasks(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListConnectorTasksRequest()
        request.instance_id = "{instance_id}"
        response = client.list_connector_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListConnectorTasksRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListConnectorTasks(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.5 Consulta de detalhes de tarefas do Smart Connect

Função

Essa API é usada para consultar detalhes da tarefa do Smart Connect.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}

Tabela 5-114 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa do Smart Connect.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-115 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
task_name	String	Nome da tarefa do Smart Connect.
topics	String	Tópico de uma tarefa do Smart Connect.
topics_regex	String	Expressão regular do tópico de uma tarefa do Smart Connect.
source_type	String	Tipo de origem de uma tarefa do Smart Connect.
source_task	SmartConnectTaskRespSourceConfig object	Configuração de origem de uma tarefa do Smart Connect.
sink_type	String	Tipo de destino de uma tarefa do Smart Connect.
sink_task	SmartConnectTaskRespSinkConfig object	Tipo de destino de uma tarefa do Smart Connect.
id	String	ID de uma tarefa do Smart Connect.
status	String	Status da tarefa do Smart Connect.
create_time	Long	Hora em que a tarefa do Smart Connect foi criada.

Tabela 5-116 SmartConnectTaskRespSourceConfig

Parâmetro	Tipo	Descrição
redis_address	String	Endereço da instância do Redis. (Exibido somente quando o tipo de origem é Redis.)
redis_type	String	Tipo de instância do Redis. (Exibido somente quando o tipo de origem é Redis.)
dcs_instance_id	String	ID da instância do DCS. (Exibido somente quando o tipo de origem é Redis.)
sync_mode	String	Tipo de sincronização: RDB_ONLY indica sincronização completa; CUSTOM_OFFSET indica sincronização completa e incremental. (Exibido somente quando o tipo de origem é Redis.)
full_sync_wait_ms	Integer	Intervalo de novas tentativas de sincronização total, em ms. (Exibido somente quando o tipo de origem é Redis.)
full_sync_max_retry	Integer	Máximo de novas tentativas de sincronização total. (Exibido somente quando o tipo de origem é Redis.)
ratelimit	Integer	Limite de taxa, em KB/s. -1 : desativar. (Exibido somente quando o tipo de origem é Redis.)
current_cluster_name	String	Nome da instância atual do Kafka. (Exibido somente quando o tipo de origem é Kafka.)
cluster_name	String	Nome da instância do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
user_name	String	Nome de usuário da instância de Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
sasl_mechanism	String	Modo de autenticação do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
instance_id	String	ID de instância do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
bootstrap_servers	String	Endereço de instância de Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
security_protocol	String	Autenticação do Kafka de destino. (Exibido somente quando o tipo de origem é Kafka.)
direction	String	Direção de sincronização. (Exibido somente quando o tipo de origem é Kafka.)
sync_consumer_offsets_enabled	Boolean	Indica se o progresso do consumo deve ser sincronizado. (Exibido somente quando o tipo de origem é Kafka.)
replication_factor	Integer	Número de réplicas. (Exibido somente quando o tipo de origem é Kafka.)

Parâmetro	Tipo	Descrição
task_num	Integer	Número de tarefas. (Exibido somente quando o tipo de origem é Kafka.)
rename_topic_enabled	Boolean	Indica se um tópico deve ser renomeado. (Exibido somente quando o tipo de origem é Kafka.)
provenance_header_enabled	Boolean	Indica se o cabeçalho de origem deve ser adicionado. (Exibido somente quando o tipo de origem é Kafka.)
consumer_strategy	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Exibido somente quando o tipo de origem é Kafka.)
compression_type	String	Algoritmo de compactação. (Exibido somente quando o tipo de origem é Kafka.)
topics_mapping	String	Mapeamento de tópicos. (Exibido somente quando o tipo de origem é Kafka.)

Tabela 5-117 SmartConnectTaskRespSinkConfig

Parâmetro	Tipo	Descrição
redis_address	String	Endereço da instância do Redis. (Exibido somente quando o tipo de destino é Redis.)
redis_type	String	Tipo de instância do Redis. (Exibido somente quando o tipo de destino é Redis.)
dcs_instance_id	String	ID da instância do DCS. (Exibido somente quando o tipo de destino é Redis.)
target_db	Integer	Banco de dados de destino. O valor padrão é -1 . (Exibido somente quando o tipo de destino é Redis.)
consumer_strategy	String	Deslocamento inicial. latest : obter os dados mais recentes; earliest : obter os dados mais antigos. (Exibido somente quando o tipo de destino é OBS.)
destination_file_type	String	Formato do arquivo de despejo. Somente TEXT é suportado. (Exibido somente quando o tipo de destino é OBS.)
deliver_time_interval	Integer	Período(s) de despejo. (Exibido somente quando o tipo de destino é OBS.)
obs_bucket_name	String	Endereço de despejo. (Exibido somente quando o tipo de destino é OBS.)
obs_path	String	Diretório de despejo. (Exibido somente quando o tipo de destino é OBS.)

Parâmetro	Tipo	Descrição
partition_format	String	Formato do diretório de hora. (Exibido somente quando o tipo de destino é OBS.)
record_delimiter	String	Quebra de linha. (Exibido somente quando o tipo de destino é OBS.)
store_keys	Boolean	Key de armazenamento. (Exibido somente quando o tipo de destino é OBS.)
obs_part_size	Integer	Tamanho (em bytes) de cada arquivo a ser carregado. O valor padrão é 5242880 . (Exibido somente quando o tipo de destino é OBS.)
flush_size	Integer	flush_size. (Exibido somente quando o tipo de destino é OBS.)
timezone	String	Fuso horário. (Exibido somente quando o tipo de destino é OBS.)
connector_class	String	Classe de connector. O valor padrão é com.huawei.dms.connector.obs.OBSSinkConnector . (Exibido somente quando o tipo de destino é OBS.)
storage_class	String	Classe de storage. O valor padrão é com.huawei.dms.connector.obs.storage.OBSStorage . (Exibido somente quando o tipo de destino é OBS.)
format_class	String	Classe de format. O valor padrão é com.huawei.dms.connector.obs.format.bytearray.ByteArrayFormat . (Exibido somente quando o tipo de destino é OBS.)
schema_generator_class	String	Classe de schema_generator. O valor padrão é io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator . (Exibido somente quando o tipo de destino é OBS.)
partitioner_classes	String	Classe de partitioner. O valor padrão é io.confluent.connect.storage.partitioners.TimeBasedPartitioner . (Exibido somente quando o tipo de destino é OBS.)
value_converter	String	value_converter. O valor padrão é org.apache.kafka.connect.converters.ByteArrayConverter . (Exibido somente quando o tipo de destino é OBS.)
key_converter	String	key_converter. O valor padrão é org.apache.kafka.connect.converters.ByteArrayConverter . (Exibido somente quando o tipo de destino é OBS.)
kv_delimiter	String	kv_delimiter. O valor padrão é : . (Exibido somente quando o tipo de destino é OBS.)

Exemplo de solicitações

Nenhum

Exemplo de respostas

Código de status: 200

Bem-sucedido.

```
{
  "task_name" : "smart-connect-121248117",
  "topics" : "topic-sc",
  "source_task" : {
    "redis_address" : "192.168.91.179:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "949190a2-598a-4afd-99a8-dad3caele7cd",
    "sync_mode" : "RDB_ONLY",
    "full_sync_wait_ms" : 13000,
    "full_sync_max_retry" : 4,
    "ratelimit" : -1
  },
  "source_type" : "REDIS_REPLICATOR_SOURCE",
  "sink_task" : {
    "redis_address" : "192.168.119.51:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "9b981368-a8e3-416a-87d9-1581a968b41b",
    "target_db" : -1
  },
  "sink_type" : "REDIS_REPLICATOR_SINK",
  "id" : "8a205bbd-7181-4b5e-9bd6-37274ce84577",
  "status" : "RUNNING",
  "create_time" : 1708427753133
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    }
}
```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowConnectorTaskRequest request = new ShowConnectorTaskRequest();
request.withInstanceId("{instance_id}");
request.withTaskId("{task_id}");
try {
    ShowConnectorTaskResponse response =
client.showConnectorTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.show_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.6 Exclusão de uma tarefa do Smart Connect

Função

Essa API é usada para excluir uma tarefa do Smart Connect.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

DELETE /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}

Tabela 5-118 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa do Smart Connect.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Nenhum

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();

        DeleteConnectorTaskRequest request = new DeleteConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            DeleteConnectorTaskResponse response =
client.deleteConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteConnectorTaskRequest()
    request.instance_id = "{instance_id}"
    request.task_id = "{task_id}"
    response = client.delete_connector_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.DeleteConnectorTask(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.7 Pausa de uma tarefa do Smart Connect

Função

Essa API é usada para pausar uma tarefa do Smart Connect.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/pause

Tabela 5-119 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa do Smart Connect.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Nenhum

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class PauseConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        PauseConnectorTaskRequest request = new PauseConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            PauseConnectorTaskResponse response =
            client.pauseConnectorTask(request);
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = PauseConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.pause_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.PauseConnectorTaskRequest{
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.PauseConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.8 Reinicialização de uma tarefa do Smart Connect

Função

Essa API é usada para reiniciar uma tarefa do Smart Connect.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/resume

Tabela 5-120 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa do Smart Connect.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Nenhum

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResumeConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

ResumeConnectorTaskRequest request = new ResumeConnectorTaskRequest();
request.withInstanceId("{instance_id}");
request.withTaskId("{task_id}");
try {
    ResumeConnectorTaskResponse response =
client.resumeConnectorTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResumeConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
```



```
response = client.resume_connector_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResumeConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ResumeConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia [Código de exemplo no API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.3.9 Inicialização de uma tarefa do Smart Connect ou reinicialização de uma tarefa do Smart Connect pausada ou em execução

Função

Essa API é usada para **iniciar uma tarefa do Smart Connect** ou **reiniciar uma tarefa do Smart Connect pausada ou em execução**. Observe que o progresso da sincronização será reiniciado e a tarefa será reiniciada.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/kafka/instances/{instance_id}/connector/tasks/{task_id}/restart

Tabela 5-121 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa do Smart Connect.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Nenhum

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class RestartConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();

        RestartConnectorTaskRequest request = new RestartConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            RestartConnectorTaskResponse response =
client.restartConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
    }  
  }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkkafka.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in  
    plaintext, which has great security risks. It is recommended that the AK and SK  
    be stored in ciphertext in configuration files or environment variables and  
    decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for  
    authentication. Before running this example, set environment variables  
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId) \  
  
    client = KafkaClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = RestartConnectorTaskRequest()  
        request.instance_id = "{instance_id}"  
        request.task_id = "{task_id}"  
        response = client.restart_connector_task(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in  
    plaintext, which has great security risks. It is recommended that the AK and SK  
    be stored in ciphertext in configuration files or environment variables and  
    decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for  
    authentication. Before running this example, set environment variables  
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().
```

```
WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.RestartConnectorTaskRequest{}
request.InstanceId = "{instance_id}"
request.TaskId = "{task_id}"
response, err := client.RestartConnectorTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Bem-sucedido.

Códigos de erro

Consulte [Códigos de erro](#).

5.4 Gerenciamento de modificação de especificação

5.4.1 Aumento das especificações da instância

Função

Essa API é usada para modificar as especificações da instância. Atualmente, essa API só pode ser usada para aumentar as especificações de instâncias de pagamento por uso.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{engine}/{project_id}/instances/{instance_id}/extend

Tabela 5-122 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens. Padrão: kafka
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-123 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
oper_type	Sim	String	Tipo de mudança. Intervalo de valores: <ul style="list-style-type: none"> ● storage: expandir o armazenamento sem alterar a quantidade do corretor. ● horizontal: adicionar corretores sem redimensionar o espaço de armazenamento de cada corretor. ● vertical: ampliar o flavor do corretor sem alterar a quantidade e o armazenamento do corretor.

Parâmetro	Obrigatório	Tipo	Descrição
new_storage_space	Não	Integer	<p>Novo espaço de armazenamento.</p> <p>Este parâmetro é válido e obrigatório quando oper_type é definido como storage ou horizontal.</p> <p>Espaço de armazenamento da instância = Número de corretores x Espaço de armazenamento de cada corretor.</p> <p>Se oper_type estiver definido como storage, o número de corretores permanece inalterado e o espaço de armazenamento de cada corretor deve ser expandido em pelo menos 100 GB.</p> <ul style="list-style-type: none"> ● Se oper_type for definido como horizontal, o espaço de armazenamento de cada corretor permanecerá inalterado.
new_broker_num	Não	Integer	<p>Este parâmetro é válido somente quando oper_type é definido como horizontal.</p> <p>Intervalo de valores: um máximo de 30 corretores são suportados.</p>
new_product_id	Não	String	<p>Novo ID de produto para aumento de escala.</p> <p>Este parâmetro é válido e obrigatório quando oper_type é definido como vertical.</p> <p>Obtenha o ID do produto a partir da Consulta à lista de especificações do produto.</p>
publicip_id	Não	String	<p>ID do EIP vinculado à instância.</p> <p>Use vírgulas (,) para separar vários IDs de EIP.</p> <p>Este parâmetro é obrigatório quando oper_type é definido como horizontal.</p>

Parâmetro	Obrigatório	Tipo	Descrição
tenant_ips	Não	Array of strings	Endereços IP privados IPv4 especificados. O número de endereços IP especificados deve ser menor ou igual ao número de novos corretores. Se o número de endereços IP especificados for menor que o número de corretores, os corretores não especificados receberão endereços IP privados aleatoriamente atribuídos.
second_tenant_subnet_id	Não	String	ID da sub-rede em espera usada pelos novos corretores na expansão da instância. Esse valor é transferido quando uma sub-rede em espera é usada na expansão da instância. Entre em contato com o atendimento ao cliente para usar o valor.

Parâmetros de resposta

Código de status: 200

Tabela 5-124 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
job_id	String	ID da tarefa de modificação da especificação.

Exemplo de solicitações

- **Expansão do espaço de armazenamento (pagamento por uso)**
 POST https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend

```
{
  "oper_type" : "storage",
  "new_storage_space" : 600
}
```
- **Adição de corretores (pagamento por uso)**
 POST https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend

```
{
  "oper_type" : "horizontal",
  "new_storage_space" : 1600,
}
```



```
"new_broker_num" : 4,  
"tenant_ips" : [ "127.0.0.1", "127.0.0.2", "127.0.0.3" ]  
}
```

- **Aumento do flavor do corretor (pagamento por uso)**

POST https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend

```
{  
  "oper_type" : "vertical",  
  "new_product_id" : "c6.4u8g.cluster"  
}
```

Exemplo de respostas

Código de status: 200

Especificações de instância aumentadas.

```
{  
  "job_id" : "93b94287-728d-4bb1-a158-cb66cb0854e7"  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

- **Expansão do espaço de armazenamento (pagamento por uso)**

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ResizeEngineInstanceSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
        plaintext, which has great security risks. It is recommended that the AK and  
        SK be stored in ciphertext in configuration files or environment variables  
        and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for  
        authentication. Before running this example, set environment variables  
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ResizeEngineInstanceRequest request = new  
        ResizeEngineInstanceRequest();
```

```
request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
});
    request.withInstanceId("{instance_id}");
    ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
    body.withNewStorageSpace(600);
    body.withOperType("storage");
    request.withBody(body);
    try {
        ResizeEngineInstanceResponse response =
client.resizeEngineInstance(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- **Adição de corretores (pagamento por uso)**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ResizeEngineInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ResizeEngineInstanceRequest request = new
ResizeEngineInstanceRequest();

request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine} "
));
        request.withInstanceId("{instance_id}");
```

```
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
List<String> listbodyTenantIps = new ArrayList<>();
listbodyTenantIps.add("127.0.0.1");
listbodyTenantIps.add("127.0.0.2");
listbodyTenantIps.add("127.0.0.3");
body.withTenantIps(listbodyTenantIps);
body.withNewBrokerNum(4);
body.withNewStorageSpace(1600);
body.withOperType("horizontal");
request.withBody(body);
try {
    ResizeEngineInstanceResponse response =
client.resizeEngineInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- **Aumento do flavor do corretor (pagamento por uso)**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResizeEngineInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ResizeEngineInstanceRequest request = new
ResizeEngineInstanceRequest();

        request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"
));
        request.withInstanceId("{instance_id}");
```

```
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
body.withNewProductId("c6.4u8g.cluster");
body.withOperType("vertical");
request.withBody(body);
try {
    ResizeEngineInstanceResponse response =
client.resizeEngineInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Expansão do espaço de armazenamento (pagamento por uso)

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeEngineInstanceRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        request.body = ResizeEngineInstanceReq(
            new_storage_space=600,
            oper_type="storage"
        )
        response = client.resize_engine_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Adição de corretores (pagamento por uso)

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeEngineInstanceRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        listTenantIpsbody = [
            "127.0.0.1",
            "127.0.0.2",
            "127.0.0.3"
        ]
        request.body = ResizeEngineInstanceReq(
            tenant_ips=listTenantIpsbody,
            new_broker_num=4,
            new_storage_space=1600,
            oper_type="horizontal"
        )
        response = client.resize_engine_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- **Aumento do flavor do corretor (pagamento por uso)**

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
```

```
client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ResizeEngineInstanceRequest()
    request.engine = "{engine}"
    request.instance_id = "{instance_id}"
    request.body = ResizeEngineInstanceReq(
        new_product_id="c6.4u8g.cluster",
        oper_type="vertical"
    )
    response = client.resize_engine_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Expansão do espaço de armazenamento (pagamento por uso)

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{
        request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
        request.InstanceId = "{instance_id}"
        newStorageSpaceResizeEngineInstanceReq:= int32(600)
        request.Body = &model.ResizeEngineInstanceReq{
            NewStorageSpace: &newStorageSpaceResizeEngineInstanceReq,
            OperType: "storage",
        }
    }
    response, err := client.ResizeEngineInstance(request)
```

```

    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- **Adição de corretores (pagamento por uso)**

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and
    // SK be stored in ciphertext in configuration files or environment variables
    // and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{}
    request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
    request.InstanceId = "{instance_id}"
    var listTenantIpsbody = []string{
        "127.0.0.1",
        "127.0.0.2",
        "127.0.0.3",
    }
    newBrokerNumResizeEngineInstanceReq:= int32(4)
    newStorageSpaceResizeEngineInstanceReq:= int32(1600)
    request.Body = &model.ResizeEngineInstanceReq{
        TenantIps: &listTenantIpsbody,
        NewBrokerNum: &newBrokerNumResizeEngineInstanceReq,
        NewStorageSpace: &newStorageSpaceResizeEngineInstanceReq,
        OperType: "horizontal",
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- **Aumento do flavor do corretor (pagamento por uso)**

```

package main

```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/
region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{
        request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
        request.InstanceId = "{instance_id}"
        newProductIdResizeEngineInstanceReq:= "c6.4u8g.cluster"
        request.Body = &model.ResizeEngineInstanceReq{
            NewProductId: &newProductIdResizeEngineInstanceReq,
            OperType: "vertical",
        }
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Especificações de instância aumentadas.

Códigos de erro

Consulte [Códigos de erro](#).

5.4.2 Consulta de informações do produto para modificação da especificação de instância

Função

Esta API é usada para consultar as informações do produto para modificação de especificações de instância.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}/extend

Tabela 5-125 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens. Padrão: kafka
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-126 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
type	Sim	String	Edição do produto. ● advanced : edição premium

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-127 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
engine	String	Mecanismo de mensagens: Kafka.
versions	Array of strings	Versões suportadas pelo mecanismo de mensagens.
products	Array of ExtendProductInfoEntity objects	Informações sobre o produto para modificação de especificações.

Tabela 5-128 ExtendProductInfoEntity

Parâmetro	Tipo	Descrição
type	String	Tipos de instâncias.
product_id	String	ID do produto.
ecs_flavor_id	String	Flavor de ECS usado pelo produto.
arch_types	Array of strings	Arquiteturas de CPU suportadas.
charging_mode	Array of strings	Modos de cobrança suportados.
ios	Array of ExtendProductIosEntity objects	Informações de I/O do disco.
support_features	Array of ExtendProductSupportFeaturesEntity objects	Recursos suportados.
properties	ExtendProductPropertiesEntity object	Descrição da especificação do produto.
available_zones	Array of strings	As AZs onde há recursos disponíveis.
unavailable_zones	Array of strings	AZs onde os recursos estão esgotados.

Tabela 5-129 ExtendProductIosEntity

Parâmetro	Tipo	Descrição
io_spec	String	Especificação de I/O de armazenamento.
available_zones	Array of strings	As AZs onde há recursos disponíveis.

Parâmetro	Tipo	Descrição
type	String	Tipo de I/O.
unavailable_zones	Array of strings	AZs onde os recursos estão esgotados.

Tabela 5-130 ExtendProductSupportFeaturesEntity

Parâmetro	Tipo	Descrição
name	String	Nome de recursos.
properties	Map<String,String>	Par de chave-valor de um recurso.

Tabela 5-131 ExtendProductPropertiesEntity

Parâmetro	Tipo	Descrição
max_partition_per_broker	String	Número máximo de partições de cada corretor.
max_broker	String	Número máximo de corretores.
max_storage_per_node	String	Espaço máximo de armazenamento de cada corretor. Unidade: GB.
max_consumer_per_broker	String	Número máximo de consumidores de cada corretor.
min_broker	String	Número mínimo de corretores.
max_bandwidth_per_broker	String	Largura de banda máxima de cada corretor.
min_storage_per_node	String	Espaço mínimo de armazenamento de cada corretor. Unidade: GB.
max_tps_per_broker	String	TPS máximo de cada corretor.
product_alias	String	Alias de product_id .

Exemplo de solicitações

Consulta de informações do produto para modificação da especificação de exemplo

```
GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend?
type={type}
```

Exemplo de respostas

Código de status: 200

As informações do produto para modificação da especificação da instância foram consultadas com sucesso.

- Informações do produto para modificação de especificação de instância consultadas.

```
{
  "engine" : "kafka",
  "versions" : [ "1.1.0", "2.7" ],
  "products" : [ {
    "type" : "cluster",
    "product_id" : "c6.2u4g.cluster",
    "ecs_flavor_id" : "c3.large.2",
    "arch_types" : [ "X86" ],
    "charging_mode" : [ "monthly", "hourly" ],
    "ios" : [ {
      "io_spec" : "dms.physical.storage.high.v2",
      "available_zones" : [ "xxx" ],
      "type" : "evs",
      "unavailable_zones" : [ ]
    }, {
      "io_spec" : "dms.physical.storage.ultra.v2",
      "available_zones" : [ "xxx" ],
      "type" : "evs",
      "unavailable_zones" : [ ]
    } ],
    "support_features" : [ {
      "name" : "connector_obs",
      "properties" : {
        "max_task" : "10",
        "max_node" : "10",
        "min_task" : "1",
        "min_node" : "2"
      }
    } ],
    "properties" : {
      "max_partition_per_broker" : "250",
      "max_broker" : "30",
      "max_storage_per_node" : "10000",
      "max_consumer_per_broker" : "4000",
      "min_broker" : "3",
      "product_alias" : "kafka.2u4g.cluster",
      "max_bandwidth_per_broker" : "100",
      "min_storage_per_node" : "100",
      "max_tps_per_broker" : "30000"
    },
    "available_zones" : [ "xxx" ],
    "unavailable_zones" : [ ]
  }, {
    "type" : "cluster",
    "product_id" : "c6.2u4g.cluster.dec",
    "ecs_flavor_id" : "c6.large.2",
    "arch_types" : [ "X86" ],
    "charging_mode" : [ "monthly", "hourly" ],
    "ios" : [ {
      "io_spec" : "dms.physical.storage.high.dss.v2",
      "available_zones" : [ "xxx" ],
      "type" : "evs",
      "unavailable_zones" : [ ]
    }, {
      "io_spec" : "dms.physical.storage.ultra.dss.v2",
      "available_zones" : [ "xxx" ],
      "type" : "evs",
      "unavailable_zones" : [ ]
    }, {
      "io_spec" : "dms.physical.storage.ultra.v2",
      "available_zones" : [ "xxx" ],

```

```

        "type" : "evs",
        "unavailable_zones" : [ ]
    }, {
        "io_spec" : "dms.physical.storage.high.v2",
        "available_zones" : [ "xxx" ],
        "type" : "evs",
        "unavailable_zones" : [ ]
    } ],
    "support_features" : [ {
        "name" : "connector_obs",
        "properties" : {
            "max_task" : "10",
            "max_node" : "10",
            "min_task" : "1",
            "min_node" : "2"
        }
    } ],
    "properties" : {
        "max_partition_per_broker" : "100",
        "max_broker" : "30",
        "max_storage_per_node" : "400",
        "max_consumer_per_broker" : "4000",
        "min_broker" : "3",
        "product_alias" : "kafka.2u4g.cluster.dec",
        "max_bandwidth_per_broker" : "100",
        "min_storage_per_node" : "100",
        "max_tps_per_broker" : "30000"
    },
    "available_zones" : [ ],
    "unavailable_zones" : [ "xxx" ]
}, {
    "type" : "cluster",
    "product_id" : "c6.4u8g.cluster",
    "ecs_flavor_id" : "c3.xlarge.2",
    "arch_types" : [ "X86" ],
    "charging_mode" : [ "monthly", "hourly" ],
    "ios" : [ {
        "io_spec" : "dms.physical.storage.high.v2",
        "available_zones" : [ "xxx" ],
        "type" : "evs",
        "unavailable_zones" : [ ]
    }, {
        "io_spec" : "dms.physical.storage.ultra.v2",
        "available_zones" : [ "xxx" ],
        "type" : "evs",
        "unavailable_zones" : [ ]
    } ],
    "support_features" : [ {
        "name" : "connector_obs",
        "properties" : {
            "max_task" : "10",
            "max_node" : "10",
            "min_task" : "1",
            "min_node" : "2"
        }
    } ],
    "properties" : {
        "max_partition_per_broker" : "500",
        "max_broker" : "30",
        "max_storage_per_node" : "20000",
        "max_consumer_per_broker" : "4000",
        "min_broker" : "3",
        "product_alias" : "kafka.4u8g.cluster",
        "max_bandwidth_per_broker" : "100",
        "min_storage_per_node" : "100",
        "max_tps_per_broker" : "100000"
    },
    "available_zones" : [ "xxx" ],
    "unavailable_zones" : [ ]

```

```

}, {
  "type" : "cluster",
  "product_id" : "c6.8u16g.cluster",
  "ecs_flavor_id" : "c3.2xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "1000",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "product_alias" : "kafka.8u16g.cluster",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "100",
    "max_tps_per_broker" : "150000"
  },
  "available_zones" : [ "xxx" ],
  "unavailable_zones" : [ ]
}, {
  "type" : "cluster",
  "product_id" : "c6.12u24g.cluster",
  "ecs_flavor_id" : "c3.3xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "1500",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",

```

```
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "product_alias" : "kafka.12u24g.cluster",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "100",
    "max_tps_per_broker" : "200000"
  },
  "available_zones" : [ "xxx" ],
  "unavailable_zones" : [ ]
}, {
  "type" : "cluster",
  "product_id" : "c6.16u32g.cluster",
  "ecs_flavor_id" : "c3.4xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "2000",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "product_alias" : "kafka.16u32g.cluster",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "100",
    "max_tps_per_broker" : "250000"
  },
  "available_zones" : [ "xxx" ],
  "unavailable_zones" : [ ]
} ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class ShowEngineInstanceExtendProductInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ShowEngineInstanceExtendProductInfoRequest request = new
ShowEngineInstanceExtendProductInfoRequest();

request.withEngine(ShowEngineInstanceExtendProductInfoRequest.EngineEnum.fromValu
e("{engine}"));
        request.withInstanceId("{instance_id}");
        try {
            ShowEngineInstanceExtendProductInfoResponse response =
client.showEngineInstanceExtendProductInfo(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```



```
credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowEngineInstanceExtendProductInfoRequest()
    request.engine = "{engine}"
    request.instance_id = "{instance_id}"
    response = client.show_engine_instance_extend_product_info(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowEngineInstanceExtendProductInfoRequest{}
    request.Engine =
model.GetShowEngineInstanceExtendProductInfoRequestEngineEnum().ENGINE
    request.InstanceId = "{instance_id}"
    response, err := client.ShowEngineInstanceExtendProductInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As informações do produto para modificação da especificação da instância foram consultadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.5 Gerenciamento de tópicos

5.5.1 Produção de mensagens para o Kafka

Função

Essa API é usada para enviar mensagens especificadas no console para instâncias do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/messages/action

Tabela 5-132 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-133 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
action_id	Sim	String	ID da ação, que é send para produção.

Parâmetros de solicitação

Tabela 5-134 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
topic	Sim	String	Tópicos do Kafka.
body	Sim	String	Conteúdo da mensagem.
property_list	Sim	Array of property_list objects	Informações sobre a partição do tópico.

Tabela 5-135 property_list

Parâmetro	Obrigatório	Tipo	Descrição
name	Não	String	Nome de recursos.
value	Não	String	Valor do recurso.

Parâmetros de resposta

Código de status: 200

Tabela 5-136 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
topic	String	Tópico do Kafka.
body	String	Conteúdo da mensagem.
property_list	Array of objects	Informações sobre a partição do tópico.

Exemplo de solicitações

Envio de mensagens no console do Kafka

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/messages/action?
action_id={action_id}
```

```
{
  "topic" : "XXXX",
  "body" : "hello world",
  "property_list" : [ {
    "name" : "KEY",
    "value" : "testKey"
  }, {
    "name" : "PARTITION",
    "value" : "0"
  } ]
}
```

Exemplo de respostas

Código de status: 200

Mensagem produzida.

```
{
  "topic" : "XXXX",
  "body" : "XXXX",
  "property_list" : [ ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Envio de mensagens no console do Kafka

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class SendKafkaMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
SendKafkaMessageRequest request = new SendKafkaMessageRequest();
request.withInstanceId("{instance_id}");
SendKafkaMessageRequestBody body = new SendKafkaMessageRequestBody();
List<SendKafkaMessageRequestBodyPropertyList> listbodyPropertyList = new
ArrayList<>();
listbodyPropertyList.add(
    new SendKafkaMessageRequestBodyPropertyList()
        .withName("KEY")
        .withValue("testKey")
);
listbodyPropertyList.add(
    new SendKafkaMessageRequestBodyPropertyList()
        .withName("PARTITION")
        .withValue("0")
);
body.withPropertyList(listbodyPropertyList);
body.withBody("hello world");
body.withTopic("XXXX");
request.withBody(body);
try {
    SendKafkaMessageResponse response = client.sendKafkaMessage(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Envio de mensagens no console do Kafka

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = SendKafkaMessageRequest()
    request.instance_id = "{instance_id}"
    listPropertyListbody = [
        SendKafkaMessageRequestBodyPropertyList(
            name="KEY",
            value="testKey"
        ),
        SendKafkaMessageRequestBodyPropertyList(
            name="PARTITION",
            value="0"
        )
    ]
    request.body = SendKafkaMessageRequestBody(
        property_list=listPropertyListbody,
        body="hello world",
        topic="XXXX"
    )
    response = client.send_kafka_message(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Envio de mensagens no console do Kafka

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SendKafkaMessageRequest{
        request.InstanceId = "{instance_id}"
        namePropertyList:= "KEY"
        valuePropertyList:= "testKey"
```

```
namePropertyList1:= "PARTITION"
valuePropertyList1:= "0"
var listPropertyListbody = []model.SendKafkaMessageRequestBodyPropertyList{
    {
        Name: &namePropertyList,
        Value: &valuePropertyList,
    },
    {
        Name: &namePropertyList1,
        Value: &valuePropertyList1,
    },
}
request.Body = &model.SendKafkaMessageRequestBody{
    PropertyList: listPropertyListbody,
    Body: "hello world",
    Topic: "XXXX",
}
response, err := client.SendKafkaMessage(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Mensagem produzida.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.2 Criação de um tópico para uma instância do Kafka

Função

Essa API é usada para criar um tópico para uma instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/topics

Tabela 5-137 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto. Mínimo: 1 Máximo: 64
instance_id	Sim	String	ID da instância. Mínimo: 1 Máximo: 64

Parâmetros de solicitação

Tabela 5-138 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
id	Sim	String	Nome do tópico. O nome de um tópico tem de 3 a 200 caracteres, começa com uma letra e contém apenas letras, hifens (-), sublinhados (_), pontos (.) e dígitos. Mínimo: 3 Máximo: 200
replication	Não	Integer	Número de réplicas, que é configurado para garantir a confiabilidade dos dados. Intervalo de valores: 1 a 3. Mínimo: 1 Máximo: 3
sync_message_flush	Não	Boolean	Se a liberação síncrona está ativada. O valor padrão é false . A liberação síncrona compromete o desempenho.
partition	Não	Integer	Número de partições de tópicos, que é usado para definir o número de mensagens consumidas simultaneamente. Intervalo de valores: 1–100. Mínimo: 1 Máximo: 100

Parâmetro	Obrigatório	Tipo	Descrição
sync_replication	Não	Boolean	Se a replicação síncrona está ativada. Depois que essa função for ativada, o parâmetro acks no cliente do produtor deve ser definido como -1 . Caso contrário, esse parâmetro não terá efeito.
retention_time	Não	Integer	Indica o período de retenção de uma mensagem. O valor padrão é 72 . Intervalo de valores: 1–168. Unidade: hora.
topic_other_configs	Não	Array of topic_other_configs objects	Configuração do tópico. Comprimento da matriz: 0–100
topic_desc	Não	String	Descrição do tópico. Mínimo: 0 Máximo: 200

Tabela 5-139 topic_other_configs

Parâmetro	Obrigatório	Tipo	Descrição
name	Não	String	Nome da configuração.
value	Não	String	Valor de configuração.

Parâmetros de resposta

Código de status: 200

Tabela 5-140 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
name	String	Nome do tópico. Mínimo: 3 Máximo: 200

Exemplo de solicitações

Criação de um tópico chamado test01

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics
{
```

```
"id" : "test01",
"partition" : 3,
"replication" : 3,
"retention_time" : 72,
"sync_message_flush" : false,
"sync_replication" : "false",
"topic_other_configs" : [ {
  "name" : "message.timestamp.type",
  "value" : "LogAppendTime"
}, {
  "name" : "max.message.bytes",
  "value" : 10485760
} ],
"topic_desc" : ""
}
```

Exemplo de respostas

Código de status: 200

Bem-sucedido. Um nome de tópico é retornado.

```
{
  "name" : "test01"
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Criação de um tópico chamado test01

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceTopicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CreateInstanceTopicRequest request = new CreateInstanceTopicRequest();
request.withInstanceId("{instance_id}");
CreateInstanceTopicReq body = new CreateInstanceTopicReq();
List<CreateInstanceTopicReqTopicOtherConfigs> listbodyTopicOtherConfigs =
new ArrayList<>();
listbodyTopicOtherConfigs.add(
    new CreateInstanceTopicReqTopicOtherConfigs()
        .withName("message.timestamp.type")
        .withValue("LogAppendTime")
);
listbodyTopicOtherConfigs.add(
    new CreateInstanceTopicReqTopicOtherConfigs()
        .withName("max.message.bytes")
        .withValue("10485760")
);
body.withTopicDesc("");
body.withTopicOtherConfigs(listbodyTopicOtherConfigs);
body.withRetentionTime(72);
body.withSyncReplication(false);
body.withPartition(3);
body.withSyncMessageFlush(false);
body.withReplication(3);
body.withId("test01");
request.withBody(body);
try {
    CreateInstanceTopicResponse response =
client.createInstanceTopic(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Criação de um tópico chamado test01

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateInstanceTopicRequest()
    request.instance_id = "{instance_id}"
    listTopicOtherConfigsbody = [
        CreateInstanceTopicReqTopicOtherConfigs(
            name="message.timestamp.type",
            value="LogAppendTime"
        ),
        CreateInstanceTopicReqTopicOtherConfigs(
            name="max.message.bytes",
            value="10485760"
        )
    ]
    request.body = CreateInstanceTopicReq(
        topic_desc="",
        topic_other_configs=listTopicOtherConfigsbody,
        retention_time=72,
        sync_replication=False,
        partition=3,
        sync_message_flush=False,
        replication=3,
        id="test01"
    )
    response = client.create_instance_topic(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Criação de um tópico chamado test01

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```

Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateInstanceTopicRequest{}
request.InstanceId = "{instance_id}"
nameTopicOtherConfigs:= "message.timestamp.type"
valueTopicOtherConfigs:= "LogAppendTime"
nameTopicOtherConfigs1:= "max.message.bytes"
valueTopicOtherConfigs1:= "10485760"
var listTopicOtherConfigsbody =
[]model.CreateInstanceTopicReqTopicOtherConfigs{
    {
        Name: &nameTopicOtherConfigs,
        Value: &valueTopicOtherConfigs,
    },
    {
        Name: &nameTopicOtherConfigs1,
        Value: &valueTopicOtherConfigs1,
    },
}
topicDescCreateInstanceTopicReq:= ""
retentionTimeCreateInstanceTopicReq:= int32(72)
syncReplicationCreateInstanceTopicReq:= false
partitionCreateInstanceTopicReq:= int32(3)
syncMessageFlushCreateInstanceTopicReq:= false
replicationCreateInstanceTopicReq:= int32(3)
request.Body = &model.CreateInstanceTopicReq{
    TopicDesc: &topicDescCreateInstanceTopicReq,
    TopicOtherConfigs: &listTopicOtherConfigsbody,
    RetentionTime: &retentionTimeCreateInstanceTopicReq,
    SyncReplication: &syncReplicationCreateInstanceTopicReq,
    Partition: &partitionCreateInstanceTopicReq,
    SyncMessageFlush: &syncMessageFlushCreateInstanceTopicReq,
    Replication: &replicationCreateInstanceTopicReq,
    Id: "test01",
}
response, err := client.CreateInstanceTopic(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	Bem-sucedido. Um nome de tópico é retornado.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.3 Listagem dos tópicos de uma instância de Kafka

Função

Essa API é usada para consultar detalhes sobre tópicos de uma instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/topics

Tabela 5-141 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.

Tabela 5-142 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
offset	Não	String	Deslocamento, que é a posição onde a consulta começa. O valor deve ser maior ou igual a 0.
limit	Não	String	Número máximo de instâncias retornadas na consulta atual. O valor padrão é 10 . O valor varia de 1 a 50.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-143 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
total	Integer	Número total de tópicos.
size	Integer	Número máximo de registros a serem exibidos em uma página.
remain_partitions	Integer	Número de partições restantes.
max_partitions	Integer	Número total de partições.
topic_max_partitions	Integer	Número máximo de partições em um único tópico.
topics	Array of TopicEntity objects	Lista de tópicos.

Tabela 5-144 TopicEntity

Parâmetro	Tipo	Descrição
policiesOnly	Boolean	Se esta política é a política padrão.
name	String	Nome do tópico.
replication	Integer	Número de réplicas, que é configurado para garantir a confiabilidade dos dados.
partition	Integer	Número de partições de tópico, que é usado para definir o número de mensagens consumidas simultaneamente.
retention_time	Integer	Período de retenção de uma mensagem.
sync_replication	Boolean	Se a replicação síncrona está ativada. Depois que essa função for ativada, o parâmetro acks no cliente do produtor deve ser definido como -1 . Caso contrário, esse parâmetro não terá efeito.
sync_message_flush	Boolean	Se a liberação síncrona está ativada. O valor padrão é false . A liberação síncrona compromete o desempenho.
external_configs	Object	Configuração estendida.
topic_type	Integer	Tipo de tópico. Opções: 0 : tópico comum; 1 : tópico do sistema (interno).
topic_other_configs	Array of topic_other_configs objects	Outras configurações de tópicos.

Parâmetro	Tipo	Descrição
topic_desc	String	Descrição do tópico.
created_at	Long	Tempo de criação do tópico.

Tabela 5-145 topic_other_configs

Parâmetro	Tipo	Descrição
name	String	Nome da configuração.
valid_values	String	Valor válido.
default_value	String	Valor padrão.
config_type	String	Tipo de configuração. O valor pode ser dynamic ou static .
value	String	Valor de configuração.
value_type	String	Tipo de valor de configuração.

Exemplo de solicitações

Consulta da lista de tópicos

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics?
offset=0&limit=10
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "total" : 1,
  "size" : 0,
  "topics" : [ {
    "policiesOnly" : false,
    "name" : "Topic-test01",
    "replication" : 3,
    "partition" : 3,
    "retention_time" : 72,
    "sync_replication" : "false",
    "sync_message_flush" : "false",
    "topic_other_configs" : [ {
      "name" : "max.message.bytes",
      "valid_values" : "[0...10485760]",
      "default_value" : "10485760",
      "config_type" : "dynamic",
      "value" : "10485760",
      "value_type" : "int"
    }, {
      "name" : "message.timestamp.type",
      "valid_values" : "[CreateTime, LogAppendTime]",
      "default_value" : "LogAppendTime",
```



```
    "config_type" : "dynamic",
    "value" : "LogAppendTime",
    "value_type" : "string"
  } ],
  "external_configs" : { },
  "topic_type" : 0,
  "topic_desc" : "This is a test topic",
  "created_at" : 1688112779916
} ],
"remain_partitions" : 294,
"max_partitions" : 300,
"topic_max_partitions" : 200
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListInstanceTopicsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInstanceTopicsRequest request = new ListInstanceTopicsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListInstanceTopicsResponse response =
client.listInstanceTopics(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceTopicsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_instance_topics(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListInstanceTopicsRequest{
    request.InstanceId = "{instance_id}"
}
response, err := client.ListInstanceTopics(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.4 Modificação dos tópicos de uma instância de Kafka

Função

Essa API é usada para modificar o endereço IP privado para acesso entre VPCs.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/crossvpc/modify

Tabela 5-146 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-147 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
advertised_ip_contents	Sim	Map<String,String>	<p>Par de chave-valor advertised_ip_contents definido pelo usuário.</p> <p>A chave é o endereço IP dos ouvintes.</p> <p>O valor é o endereço IP advertised.listeners ou nome de domínio.</p> <p>NOTA</p> <p>Preencha os itens que não são modificados durante a alteração de endereço IP.</p>

Parâmetros de resposta

Código de status: 200

Tabela 5-148 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
success	Boolean	Resultado da modificação de acesso entre VPCs.
results	Array of results objects	Detalhes do resultado da modificação de acesso entre VPCs.

Tabela 5-149 results

Parâmetro	Tipo	Descrição
advertised_ip	String	Endereço IP advertised.listeners ou nome de domínio.

Parâmetro	Tipo	Descrição
success	Boolean	Status da modificação de acesso entre VPCs.
ip	String	Endereço IP dos ouvintes.

Exemplo de solicitações

Modificação do endereço IP privado para acesso entre VPCs.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/crossvpc/modify
{
  "advertised_ip_contents" : {
    "192.168.245.246" : "192.168.245.247",
    "192.168.197.36" : "192.168.197.38",
    "192.168.190.11" : "192.168.190.11"
  }
}
```

Exemplo de respostas

Código de status: 200

O endereço IP privado para acesso entre VPCs foi modificado com sucesso.

```
{
  "success" : true,
  "results" : [ {
    "advertised_ip" : "192.168.197.36",
    "success" : true,
    "ip" : "192.168.197.36"
  }, {
    "advertised_ip" : "192.168.190.11",
    "success" : true,
    "ip" : "192.168.190.11"
  }, {
    "advertised_ip" : "192.168.245.255",
    "success" : true,
    "ip" : "192.168.245.246"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Modificação do endereço IP privado para acesso entre VPCs.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
import java.util.Map;
import java.util.HashMap;

public class UpdateInstanceCrossVpcIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateInstanceCrossVpcIpRequest request = new
        UpdateInstanceCrossVpcIpRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceCrossVpcIpReq body = new UpdateInstanceCrossVpcIpReq();
        Map<String, String> listbodyAdvertisedIpContents = new HashMap<>();
        listbodyAdvertisedIpContents.put("192.168.245.246", "192.168.245.247");
        listbodyAdvertisedIpContents.put("192.168.197.36", "192.168.197.38");
        listbodyAdvertisedIpContents.put("192.168.190.11", "192.168.190.11");
        body.withAdvertisedIpContents(listbodyAdvertisedIpContents);
        request.withBody(body);
        try {
            UpdateInstanceCrossVpcIpResponse response =
            client.updateInstanceCrossVpcIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Modificação do endereço IP privado para acesso entre VPCs.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
```

```
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateInstanceCrossVpcIpRequest()
    request.instance_id = "{instance_id}"
    listAdvertisedIpContentsbody = {
        "192.168.245.246": "192.168.245.247",
        "192.168.197.36": "192.168.197.38",
        "192.168.190.11": "192.168.190.11"
    }
    request.body = UpdateInstanceCrossVpcIpReq(
        advertised_ip_contents=listAdvertisedIpContentsbody
    )
    response = client.update_instance_cross_vpc_ip(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Modificação do endereço IP privado para acesso entre VPCs.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
```

```
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build()  
  
    request := &model.UpdateInstanceCrossVpcIpRequest{  
    request.InstanceId = "{instance_id}"  
    var listAdvertisedIpContentsbody = map[string]string{  
        "192.168.245.246": "192.168.245.247",  
        "192.168.197.36": "192.168.197.38",  
        "192.168.190.11": "192.168.190.11",  
    }  
    request.Body = &model.UpdateInstanceCrossVpcIpReq{  
        AdvertisedIpContents: listAdvertisedIpContentsbody,  
    }  
    response, err := client.UpdateInstanceCrossVpcIp(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	O endereço IP privado para acesso entre VPCs foi modificado com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.5 Exclusão dos tópicos de uma instância de Kafka em lote

Função

Essa API é usada para excluir vários tópicos de uma instância do Kafka em lotes. Se alguns tópicos forem excluídos com sucesso enquanto alguns não forem excluídos, uma resposta de sucesso será retornada com informações sobre tópicos que não foram excluídos.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/topics/delete

Tabela 5-150 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto. Mínimo: 1 Máximo: 64
instance_id	Sim	String	ID da instância. Mínimo: 1 Máximo: 64

Parâmetros de solicitação

Tabela 5-151 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
topics	Não	Array of strings	Tópicos a serem excluídos. Este parâmetro é obrigatório quando os tópicos da instância são excluídos em lotes.

Parâmetros de resposta

Código de status: 200

Tabela 5-152 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
topics	Array of topics objects	Lista de tópicos.

Tabela 5-153 topics

Parâmetro	Tipo	Descrição
id	String	Nome do tópico.
success	Boolean	Se os tópicos são excluídos.

Exemplo de solicitações

Exclusão de tópicos em lote

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics/delete
{
  "topics" : [ "topic01" ]
}
```

Exemplo de respostas

Código de status: 200

A exclusão foi bem-sucedida.

```
{
  "topics" : [ {
    "id" : "topic01",
    "success" : true
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Exclusão de tópicos em lote

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteInstanceTopicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
        BatchDeleteInstanceTopicRequest request = new
BatchDeleteInstanceTopicRequest();
        request.withInstanceId("{instance_id}");
        BatchDeleteInstanceTopicReq body = new BatchDeleteInstanceTopicReq();
        List<String> listbodyTopics = new ArrayList<>();
        listbodyTopics.add("topic01");
        body.withTopics(listbodyTopics);
        request.withBody(body);
        try {
            BatchDeleteInstanceTopicResponse response =
client.batchDeleteInstanceTopic(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Exclusão de tópicos em lote

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteInstanceTopicRequest()
        request.instance_id = "{instance_id}"
        listTopicsbody = [
            "topic01"
        ]
        request.body = BatchDeleteInstanceTopicReq(
            topics=listTopicsbody
        )
        response = client.batch_delete_instance_topic(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Exclusão de tópicos em lote

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteInstanceTopicRequest{}
    request.InstanceId = "{instance_id}"
    var listTopicsbody = []string{
        "topic01",
    }
    request.Body = &model.BatchDeleteInstanceTopicReq{
        Topics: &listTopicsbody,
    }
    response, err := client.BatchDeleteInstanceTopic(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A exclusão foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.6 Consulta da lista de partições de um tópico

Função

Essa API é usada para consultar a lista de partições de um tópico.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/partitions

Tabela 5-154 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto . Mínimo: 1 Máximo: 64
instance_id	Sim	String	ID da instância. Mínimo: 1 Máximo: 64
topic	Sim	String	Tópico. Mínimo: 1 Máximo: 200

Tabela 5-155 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
offset	Não	Integer	Deslocamento. Os registros após esse deslocamento serão consultados. Mínimo: 0 Máximo: 10000
limit	Não	Integer	Número máximo de registros que podem ser retornados. Mínimo: 1 Máximo: 50

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-156 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
total	Integer	Total de registros. Mínimo: 0 Máximo: 100
partitions	Array of partitions objects	Matriz de partições. Comprimento da matriz: 0 - 50

Tabela 5-157 partitions

Parâmetro	Tipo	Descrição
partition	Integer	ID da partição. Mínimo: 0 Máximo: 100
start_offset	Long	Deslocamento inicial. Mínimo: 0 Máximo: 9223372036854775807

Parâmetro	Tipo	Descrição
last_offset	Long	Último deslocamento. Mínimo: 0 Máximo: 9223372036854775807
message_count	Long	Número de mensagens em uma partição. Mínimo: 0 Máximo: 9223372036854775807
last_update_time	Long	Hora da atualização mais recente. Mínimo: 0 Máximo: 9223372036854775807

Exemplo de solicitações

Consulta da lista de partições de um tópico

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/partitions?start=1&limit=10
```

Exemplo de respostas

Código de status: 200

A lista de partições do tópico foi consultada com sucesso.

```
{
  "total" : 3,
  "partitions" : [ {
    "partition" : 0,
    "start_offset" : 0,
    "last_offset" : 1216303,
    "message_count" : 1216303,
    "last_update_time" : 1688011291458
  }, {
    "partition" : 1,
    "start_offset" : 0,
    "last_offset" : 985447,
    "message_count" : 985447,
    "last_update_time" : 1688011291469
  }, {
    "partition" : 2,
    "start_offset" : 0,
    "last_offset" : 923340,
    "message_count" : 923340,
    "last_update_time" : 1688011291526
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListTopicPartitionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ListTopicPartitionsRequest request = new ListTopicPartitionsRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        try {
            ListTopicPartitionsResponse response =
client.listTopicPartitions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
```



```
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListTopicPartitionsRequest()
    request.instance_id = "{instance_id}"
    request.topic = "{topic}"
    response = client.list_topic_partitions(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTopicPartitionsRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    response, err := client.ListTopicPartitions(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A lista de partições do tópico foi consultada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.7 Consulta da lista atual de produtores de um tópico

Função

Esta API é usada para consultar a lista atual de produtores de um tópico.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/producers

Tabela 5-158 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto. Mínimo: 1 Máximo: 64
instance_id	Sim	String	ID da instância. Mínimo: 1 Máximo: 64

Parâmetro	Obrigatório	Tipo	Descrição
topic	Sim	String	Tópico. Mínimo: 1 Máximo: 200

Tabela 5-159 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
offset	Não	Integer	Deslocamento. Os registros após esse deslocamento serão consultados. Mínimo: 0 Máximo: 10000
limit	Não	Integer	Número máximo de registros que podem ser retornados. Mínimo: 1 Máximo: 50

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-160 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
total	Integer	Total de registros. Mínimo: 0 Máximo: 10000
producers	Array of producers objects	Lista de produtores. Comprimento da matriz: 0-50

Tabela 5-161 producers

Parâmetro	Tipo	Descrição
producer_addresses	String	Endereço do produtor. Mínimo: 0 Máximo: 64
broker_address	String	Endereço do corretor. Mínimo: 0 Máximo: 64
join_time	Long	Hora em que o corretor foi conectado. Mínimo: 0 Máximo: 9223372036854775807

Exemplo de solicitações

Consulta da lista atual de produtores de um tópico

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/producers?offset=0&limit=10
```

Exemplo de respostas

Código de status: 200

A lista atual de produtores do tópico é consultada com sucesso.

```
{
  "total" : 3,
  "producers" : [ {
    "producer_address" : "192.0.0.149:40443",
    "broker_address" : "192.0.0.146:9092",
    "join_time" : 1687204743328
  }, {
    "producer_address" : "192.0.0.149:13807",
    "broker_address" : "192.0.0.80:9092",
    "join_time" : 1687204745939
  }, {
    "producer_address" : "192.0.0.149:31876",
    "broker_address" : "192.0.0.71:9092",
    "join_time" : 1687204744934
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListTopicProducersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ListTopicProducersRequest request = new ListTopicProducersRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        try {
            ListTopicProducersResponse response =
client.listTopicProducers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
```

```
sk = __import__ ('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListTopicProducersRequest()
    request.instance_id = "{instance_id}"
    request.topic = "{topic}"
    response = client.list_topic_producers(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTopicProducersRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    response, err := client.ListTopicProducers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A lista atual de produtores do tópico é consultada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.5.8 Consulta de detalhes do tópico

Função

Essa API é usada para consultar detalhes de tópicos de uma instância do Kafka. (Até 1s para cada chamada de instância)

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}

Tabela 5-162 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-163 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.
partitions	Array of partitions objects	Lista de partições.
group_subscribed	Array of strings	Lista de grupos de consumidores que assinam o tópico.

Tabela 5-164 partitions

Parâmetro	Tipo	Descrição
partition	Integer	ID da partição.
leader	Integer	ID do corretor onde reside a réplica líder.
leo	Integer	LEO da réplica líder de partição.
hw	Integer	Marca d'água alta (HW) da partição.
lso	Integer	Deslocamento de início de log (LSO) da réplica líder da partição.
last_update_timestamp	Long	Hora em que a última mensagem foi gravada na partição. O valor é um carimbo de data/hora Unix. Unidade: ms
replicas	Array of replicas objects	Lista de réplicas.

Tabela 5-165 replicas

Parâmetro	Tipo	Descrição
broker	Integer	ID do corretor onde reside a réplica.
leader	Boolean	Se a réplica é o líder.
in_sync	Boolean	Se a réplica está em ISR.
size	Integer	Tamanho atual do log da réplica. Unidade: byte.

Parâmetro	Tipo	Descrição
lag	Long	Número de mensagens que ficam atrás da marca d'água alta na réplica.

Exemplo de solicitações

Consulta de detalhes sobre um tópico especificado

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "topic" : "test",
  "partitions" : [ {
    "partition" : 0,
    "leader" : 2,
    "replicas" : [ {
      "broker" : 2,
      "leader" : true,
      "in_sync" : true,
      "size" : 123971146,
      "lag" : 0
    }, {
      "broker" : 1,
      "leader" : false,
      "in_sync" : true,
      "size" : 123971146,
      "lag" : 0
    }, {
      "broker" : 0,
      "leader" : false,
      "in_sync" : true,
      "size" : 123971146,
      "lag" : 0
    }
  ],
  "lso" : 0,
  "leo" : 13598,
  "hw" : 13598,
  "last_update_timestamp" : 1571477180985
}, {
  "partition" : 2,
  "leader" : 1,
  "replicas" : [ {
    "broker" : 1,
    "leader" : true,
    "in_sync" : true,
    "size" : 123889531,
    "lag" : 0
  }, {
    "broker" : 0,
    "leader" : false,
    "in_sync" : true,
    "size" : 123889531,
    "lag" : 0
  }, {
    "broker" : 2,
    "leader" : false,
```

```
    "in_sync" : true,
    "size" : 123889531,
    "lag" : 0
  } ],
  "lso" : 0,
  "leo" : 13601,
  "hw" : 13601,
  "last_update_timestamp" : 1571477077146
}, {
  "partition" : 1,
  "leader" : 0,
  "replicas" : [ {
    "broker" : 0,
    "leader" : true,
    "in_sync" : true,
    "size" : 127245604,
    "lag" : 0
  }, {
    "broker" : 2,
    "leader" : false,
    "in_sync" : true,
    "size" : 127245604,
    "lag" : 0
  }, {
    "broker" : 1,
    "leader" : false,
    "in_sync" : true,
    "size" : 127245604,
    "lag" : 0
  } ],
  "lso" : 0,
  "leo" : 13599,
  "hw" : 13599,
  "last_update_timestamp" : 1571477172959
} ],
"group_subscribed" : [ "test-consumer-group" ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceTopicDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and SK
        // be stored in ciphertext in configuration files or environment variables and
        // decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

ShowInstanceTopicDetailRequest request = new
ShowInstanceTopicDetailRequest();
request.withInstanceId("{instance_id}");
request.withTopic("{topic}");
try {
    ShowInstanceTopicDetailResponse response =
client.showInstanceTopicDetail(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceTopicDetailRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        response = client.show_instance_topic_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceTopicDetailRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    response, err := client.ShowInstanceTopicDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.6 Gerenciamento de usuário

5.6.1 Consulta da lista de usuários

Função

Essa API é usada para consultar a lista de usuários.

O gerenciamento de usuários é suportado somente quando o SASL está ativado para a instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/users

Tabela 5-166 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-167 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
users	Array of ShowInstanceUsersEntity objects	Lista de usuários.

Tabela 5-168 ShowInstanceUsersEntity

Parâmetro	Tipo	Descrição
user_name	String	Nome de usuário. Um nome de usuário deve começar com uma letra. Pode conter de 4 a 64 caracteres. Apenas letras, dígitos, sublinhados (_) e hifens (-) são permitidos.
user_desc	String	Descrição do usuário.
role	String	Função do usuário.
default_app	Boolean	Se uma aplicação é a aplicação padrão.
created_time	Long	Hora de criação.

Exemplo de solicitações

Consulta da lista de usuários.

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/users
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "users" : [ {
    "user_name" : "xxxxa",
    "role" : "guest",
    "default_app" : false,
    "created_time" : 1615431764734
  }, {
    "user_name" : "test",
    "role" : "guest",
    "default_app" : false,
    "created_time" : 1615364062463
  }, {
    "user_name" : "ROOT",
    "role" : "guest",
    "default_app" : false,
    "created_time" : 1617194246328
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceUsersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ShowInstanceUsersRequest request = new ShowInstanceUsersRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowInstanceUsersResponse response =
client.showInstanceUsers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowInstanceUsersRequest()
    request.instance_id = "{instance_id}"
    response = client.show_instance_users(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceUsersRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceUsers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.6.2 Criação de um usuário

Função

Esta API é usada para criar um usuário para uma instância do Kafka para a qual o SASL está ativado. Para instâncias do Kafka criadas antes de 15 de julho de 2023, é possível criar no máximo 20 usuários para cada instância. Para instâncias do Kafka criadas desde 15 de julho de 2023, é possível criar no máximo 500 usuários para cada instância.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/users

Tabela 5-169 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-170 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
user_name	Não	String	Nome de usuário. Este parâmetro é obrigatório para criar um usuário.
user_desc	Não	String	Descrição do usuário. Mínimo: 0 Máximo: 200
user_passwd	Não	String	Senha. Este parâmetro é obrigatório para criar um usuário. A senha deve ser diferente do nome de usuário. A senha deve atender aos seguintes requisitos de complexidade: <ul style="list-style-type: none"> ● Pode conter de 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais incluem (~!@#\$%^&*()-_+=+[[{}]:'"<.>/?)) e espaços, e não podem começar com um hífen (-).

Parâmetros de resposta

Código de status: 400

Tabela 5-171 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Código de status: 403

Tabela 5-172 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Exemplo de solicitações

Criação de um usuário cujo nome de usuário é test e a senha é Cxxx3

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/users
{
  "user_name" : "test",
  "user_passwd" : "Cxxx3"
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Criação de um usuário cujo nome de usuário é test e a senha é Cxxx3

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateInstanceUserSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CreateInstanceUserRequest request = new CreateInstanceUserRequest();
request.withInstanceId("{instance_id}");
CreateInstanceUserReq body = new CreateInstanceUserReq();
body.withUserPasswd("Cxxx3");
body.withUserName("test");
request.withBody(body);
try {
    CreateInstanceUserResponse response =
client.createInstanceUser(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}
```

Python

Criação de um usuário cujo nome de usuário é test e a senha é Cxxx3

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceUserRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateInstanceUserReq(
            user_passwd="Cxxx3",
            user_name="test"
        )
```

```
response = client.create_instance_user(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Criação de um usuário cujo nome de usuário é test e a senha é Cxxx3

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceUserRequest{
        request.InstanceId = "{instance_id}"
        userPasswdCreateInstanceUserReq:= "Cxxx3"
        userNameCreateInstanceUserReq:= "test"
        request.Body = &model.CreateInstanceUserReq{
            UserPasswd: &userPasswdCreateInstanceUserReq,
            UserName: &userNameCreateInstanceUserReq,
        }
    }
    response, err := client.CreateInstanceUser(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A criação foi bem-sucedida.
400	Parâmetros inválidos.
403	Falha na autenticação.

Códigos de erro

Consulte [Códigos de erro](#).

5.6.3 Exclusão de usuários em lote

Função

Essa API é usada para excluir vários usuários de uma instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/users

Tabela 5-173 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-174 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
action	Não	String	Tipo de exclusão. Atualmente, apenas delete é suportado. Este parâmetro é obrigatório para excluir um usuário.

Parâmetro	Obrigatório	Tipo	Descrição
users	Não	Array of strings	Lista de usuários. Este parâmetro é obrigatório para excluir um usuário.

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Exclusão dos usuários em lote.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/users
{
  "action" : "delete",
  "users" : [ "testuser" ]
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Exclusão dos usuários em lote.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteInstanceUsersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and SK
        // be stored in ciphertext in configuration files or environment variables and
        // decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

BatchDeleteInstanceUsersRequest request = new
BatchDeleteInstanceUsersRequest();
request.withInstanceId("{instance_id}");
BatchDeleteInstanceUsersReq body = new BatchDeleteInstanceUsersReq();
List<String> listbodyUsers = new ArrayList<>();
listbodyUsers.add("testuser");
body.withUsers(listbodyUsers);

body.withAction(BatchDeleteInstanceUsersReq.ActionEnum.fromValue("delete"));
request.withBody(body);
try {
    BatchDeleteInstanceUsersResponse response =
client.batchDeleteInstanceUsers(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Exclusão dos usuários em lote.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
try:
    request = BatchDeleteInstanceUsersRequest()
    request.instance_id = "{instance_id}"
    listUsersbody = [
        "testuser"
    ]
    request.body = BatchDeleteInstanceUsersReq(
        users=listUsersbody,
        action="delete"
    )
    response = client.batch_delete_instance_users(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Exclusão dos usuários em lote.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteInstanceUsersRequest{}
    request.InstanceId = "{instance_id}"
    var listUsersbody = []string{
        "testuser",
    }
    actionBatchDeleteInstanceUsersReq:=
model.GetBatchDeleteInstanceUsersReqActionEnum().DELETE
    request.Body = &model.BatchDeleteInstanceUsersReq{
        Users: &listUsersbody,
        Action: &actionBatchDeleteInstanceUsersReq,
    }
    response, err := client.BatchDeleteInstanceUsers(request)
    if err == nil {
```

```
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A exclusão foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.6.4 Redefinição de uma senha do usuário

Função

Esta API é usada para redefinir uma senha de usuário.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/users/{user_name}

Tabela 5-175 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
user_name	Sim	String	Nome de usuário.

Parâmetros de solicitação

Tabela 5-176 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
new_password	Não	String	<p>Nova senha.</p> <p>Este parâmetro é obrigatório para redefinir uma senha de usuário.</p> <p>Não pode ser o nome de usuário ou o nome de usuário escrito ao contrário. A senha deve atender aos seguintes requisitos de complexidade:</p> <ul style="list-style-type: none"> ● Pode conter de 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais incluem (~!@#\$%^&*()-_+=+[{ }]:";<.>/?) e espaços, e não podem começar com um hífen (-).

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Redefinição de uma senha de usuário.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/users/{user_name}
{
  "new_password" : "Cxxx3"
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Redefinição de uma senha de usuário.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetUserPasswrodSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetUserPasswrodRequest request = new ResetUserPasswrodRequest();
        request.withInstanceId("{instance_id}");
        request.withUserName("{user_name}");
        ResetUserPasswrodReq body = new ResetUserPasswrodReq();
        body.withNewPassword("Cxxx3");
        request.withBody(body);
        try {
            ResetUserPasswrodResponse response =
client.resetUserPasswrod(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Redefinição de uma senha de usuário.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetUserPasswrodRequest()
        request.instance_id = "{instance_id}"
        request.user_name = "{user_name}"
        request.body = ResetUserPasswrodReq(
            new_password="Cxxx3"
        )
        response = client.reset_user_passwrod(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Redefinição de uma senha de usuário.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```

Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResetUserPasswrodRequest{
request.InstanceId = "{instance_id}"
request.UserName = "{user_name}"
newPasswordResetUserPasswrodReq:= "Cxxx3"
request.Body = &model.ResetUserPasswrodReq{
    NewPassword: &newPasswordResetUserPasswrodReq,
}
response, err := client.ResetUserPasswrod(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A senha foi redefinida com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.6.5 Modificação dos parâmetros do usuário

Função

Esta API é usada para modificar os parâmetros do usuário.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

PUT /v2/{engine}/{project_id}/instances/{instance_id}/users/{user_name}

Tabela 5-177 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens.
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
user_name	Sim	String	Nome de usuário.

Parâmetros de solicitação

Tabela 5-178 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
new_password	Não	String	<p>Nova senha.</p> <p>Não pode ser o nome de usuário ou o nome de usuário escrito ao contrário. A senha deve atender aos seguintes requisitos de complexidade:</p> <ul style="list-style-type: none"> ● Pode conter de 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais incluem (~!@#\$%^&*()-_+= [{}]:",<.>/?) e espaços, e não podem começar com um hífen (-). <p>Mínimo: 8 Máximo: 32</p>
user_name	Não	String	<p>Nome de usuário.</p> <p>Mínimo: 4 Máximo: 64</p>

Parâmetro	Obrigatório	Tipo	Descrição
user_desc	Não	String	Descrição do usuário. Mínimo: 0 Máximo: 200

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Modificação dos parâmetros do usuário

```
PUT https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/users/{user_name}
{
  "new_password" : "Cxxx3",
  "user_name" : "Username.",
  "user_desc" : "User description."
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Modificação dos parâmetros do usuário

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceUserSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```



```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

UpdateInstanceUserRequest request = new UpdateInstanceUserRequest();
request.withEngine("{engine}");
request.withInstanceId("{instance_id}");
request.withUserName("{user_name}");
UpdateUserReq body = new UpdateUserReq();
body.withUserDesc("User description.");
body.withUserName("Username.");
body.withNewPassword("Cxxx3");
request.withBody(body);
try {
    UpdateInstanceUserResponse response =
client.updateInstanceUser(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Modificação dos parâmetros do usuário

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
```

```
request = UpdateInstanceUserRequest()
request.engine = "{engine}"
request.instance_id = "{instance_id}"
request.user_name = "{user_name}"
request.body = UpdateUserReq(
    user_desc="User description.",
    user_name="Username.",
    new_password="Cxxx3"
)
response = client.update_instance_user(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Modificação dos parâmetros do usuário

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build()

    request := &model.UpdateInstanceUserRequest{
        request.Engine = "{engine}"
        request.InstanceId = "{instance_id}"
        request.UserName = "{user_name}"
        userDescUpdateUserReq:= "User description."
        userNameUpdateUserReq:= "Username."
        newPasswordUpdateUserReq:= "Cxxx3"
        request.Body = &model.UpdateUserReq{
            UserDesc: &userDescUpdateUserReq,
            UserName: &userNameUpdateUserReq,
            NewPassword: &newPasswordUpdateUserReq,
        }
    }
    response, err := client.UpdateInstanceUser(request)
    if err == nil {
```

```
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	Modificado.

Códigos de erro

Consulte [Códigos de erro](#).

5.6.6 Consulta das permissões de usuário

Função

Essa API é usada para consultar permissões de usuários.

O gerenciamento de usuários é suportado somente quando o SASL está ativado para a instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v1/{project_id}/instances/{instance_id}/topics/{topic_name}/accesspolicy

Tabela 5-179 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
topic_name	Sim	String	Nome do tópico.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-180 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
name	String	Descrição do grupo de consumidores.
topic_type	Integer	Endereços IP privados IPv4 especificados. O número de endereços IP especificados deve ser menor ou igual ao número de novos corretores. Se o número de endereços IP especificados for menor que o número de corretores, os corretores não especificados receberão endereços IP privados aleatoriamente atribuídos.
policies	Array of PolicyEntity objects	Os novos corretores usam o ID de sub-rede em espera na expansão da instância. Esse valor é transferido quando uma sub-rede em espera é usada na expansão da instância. Entre em contato com o atendimento ao cliente para usar o valor.

Tabela 5-181 PolicyEntity

Parâmetro	Tipo	Descrição
owner	Boolean	Se o usuário é aquele selecionado durante a criação do tópico.
user_name	String	Nome de usuário.
access_policy	String	Tipo de permissão. <ul style="list-style-type: none"> ● all: permissões de publicação e assinatura. ● pub: permissões de publicação. ● sub: permissões de assinatura.

Exemplo de solicitações

Consulta de permissões de usuário para um tópico.

```
GET https://{endpoint}/v1/{project_id}/instances/{instance_id}/topics/{topic_name}/accesspolicy
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "name" : "topic-test",
  "policies" : [ {
    "owner" : false,
    "user_name" : "xxxxa",
    "access_policy" : "pub"
  }, {
    "owner" : false,
    "user_name" : "root",
    "access_policy" : "all"
  } ],
  "topic_type" : 0
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowTopicAccessPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTopicAccessPolicyRequest request = new ShowTopicAccessPolicyRequest();
        request.withInstanceId("{instance_id}");
        request.withTopicName("{topic_name}");
        try {
            ShowTopicAccessPolicyResponse response =
            client.showTopicAccessPolicy(request);
        } catch (Exception e) {
            // Handle exception
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTopicAccessPolicyRequest()
        request.instance_id = "{instance_id}"
        request.topic_name = "{topic_name}"
        response = client.show_topic_access_policy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowTopicAccessPolicyRequest{}
request.InstanceId = "{instance_id}"
request.TopicName = "{topic_name}"
response, err := client.ShowTopicAccessPolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.6.7 Concessão de permissões de usuário

Função

Essa API é usada para conceder permissões ao usuário.

O gerenciamento de usuários é suportado somente quando o SASL está ativado para a instância do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v1/{project_id}/instances/{instance_id}/topics/accesspolicy

Tabela 5-182 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-183 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
topics	Sim	Array of AccessPolicyTopicEntity objects	Lista de tópicos.

Tabela 5-184 AccessPolicyTopicEntity

Parâmetro	Obrigatório	Tipo	Descrição
name	Sim	String	Nome do tópico.
policies	Sim	Array of AccessPolicyEntity objects	Lista de permissões.

Tabela 5-185 AccessPolicyEntity

Parâmetro	Obrigatório	Tipo	Descrição
user_name	Não	String	Nome de usuário. Esse parâmetro é obrigatório quando você define permissões de usuário.

Parâmetro	Obrigatório	Tipo	Descrição
access_policy	Não	String	Tipo de permissão. <ul style="list-style-type: none"> ● all: permissões de publicação e assinatura. ● pub: permissões de publicação. ● sub: permissões de assinatura. Esse parâmetro é obrigatório quando você define permissões de usuário.

Parâmetros de resposta

Código de status: 400

Tabela 5-186 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Código de status: 403

Tabela 5-187 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Exemplo de solicitações

Concessão ao usuário root a permissão para publicar e assinar o topic-test

```
POST https://{endpoint}/v1/{project_id}/instances/{instance_id}/topics/
accesspolicy

{
  "topics" : [ {
    "name" : "topic-test",
    "policies" : [ {
      "user_name" : "root",
      "access_policy" : "all"
    } ]
  } ]
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Concessão ao usuário root a permissão para publicar e assinar o topic-test

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateTopicAccessPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTopicAccessPolicyRequest request = new
UpdateTopicAccessPolicyRequest();
        request.withInstanceId("{instance_id}");
        UpdateTopicAccessPolicyReq body = new UpdateTopicAccessPolicyReq();
        List<AccessPolicyEntity> listTopicsPolicies = new ArrayList<>();
        listTopicsPolicies.add(
            new AccessPolicyEntity()
                .withUserName("root")
                .withAccessPolicy(AccessPolicyEntity.AccessPolicyEnum.fromValue("a
ll"))
        );
        List<AccessPolicyTopicEntity> listbodyTopics = new ArrayList<>();
        listbodyTopics.add(
            new AccessPolicyTopicEntity()
                .withName("topic-test")
                .withPolicies(listTopicsPolicies)
        );
        body.withTopics(listbodyTopics);
```

```
        request.withBody(body);
        try {
            UpdateTopicAccessPolicyResponse response =
client.updateTopicAccessPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Concessão ao usuário root a permissão para publicar e assinar o topic-test

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTopicAccessPolicyRequest()
        request.instance_id = "{instance_id}"
        listPoliciesTopics = [
            AccessPolicyEntity(
                user_name="root",
                access_policy="all"
            )
        ]
        listTopicsbody = [
            AccessPolicyTopicEntity(
                name="topic-test",
                policies=listPoliciesTopics
            )
        ]
        request.body = UpdateTopicAccessPolicyReq(
            topics=listTopicsbody
        )
        response = client.update_topic_access_policy(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Concessão ao usuário root a permissão para publicar e assinar o topic-test

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTopicAccessPolicyRequest{
        request.InstanceId = "{instance_id}"
        userNamePolicies:= "root"
        accessPolicyPolicies:= model.GetAccessPolicyEntityAccessPolicyEnum().ALL
        var listPoliciesTopics = []model.AccessPolicyEntity{
            {
                UserName: &userNamePolicies,
                AccessPolicy: &accessPolicyPolicies,
            },
        }
    }
    var listTopicsbody = []model.AccessPolicyTopicEntity{
        {
            Name: "topic-test",
            Policies: listPoliciesTopics,
        },
    }
    request.Body = &model.UpdateTopicAccessPolicyReq{
        Topics: listTopicsbody,
    }
    response, err := client.UpdateTopicAccessPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A atualização foi bem sucedida.
400	Parâmetros inválidos.
403	Falha na autenticação.

Códigos de erro

Consulte [Códigos de erro](#).

5.7 Consulta de mensagem

5.7.1 Consulta das mensagens

Função

Essa API é usada para consultar o deslocamento e o conteúdo de uma mensagem. Essa API consulta o deslocamento da mensagem com base no carimbo de data/hora e, em seguida, consulta o conteúdo da mensagem com base no deslocamento.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/messages

Tabela 5-188 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.

Parâmetro	Obrigatório	Tipo	Descrição
instance_id	Sim	String	ID da instância.

Tabela 5-189 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
topic	Sim	String	Nome do tópico. Um nome de tópico deve começar com uma letra e só pode conter letras, hifens (-), sublinhados (_) e dígitos.
asc	Não	Boolean	Se deseja classificar as mensagens por horário.
start_time	Não	String	Hora de início. O valor é um carimbo de data/hora UNIX, em milissegundos. Esse parâmetro é obrigatório quando você consulta o deslocamento da mensagem.
end_time	Não	String	Hora de término. O valor é um carimbo de data/hora UNIX, em milissegundos. Esse parâmetro é obrigatório quando você consulta o deslocamento da mensagem.
limit	Não	String	Número de mensagens exibidas em cada página.
offset	Não	String	Número da página.
download	Não	Boolean	Se o download é necessário.
message_offset	Não	String	Deslocamento da mensagem. Esse parâmetro é obrigatório quando você consulta o conteúdo da mensagem. Se start_time e end_time não estiverem vazios, este parâmetro é inválido.

Parâmetro	Obrigatório	Tipo	Descrição
partition	Não	String	Partição. Esse parâmetro é obrigatório quando você consulta o conteúdo da mensagem. Se start_time e end_time não estiverem vazios, este parâmetro é inválido.
keyword	Não	String	Palavra-chave. O valor varia de 0 a 50. Mínimo: 0 Máximo: 50

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-190 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
messages	Array of MessagesEntity objects	Lista de mensagens.
total	Long	Número total de mensagens.
size	Long	Número de registros em cada página.

Tabela 5-191 MessagesEntity

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.
partition	Integer	Partição onde a mensagem está localizada.
key	String	Key de mensagem.
value	String	Conteúdo da mensagem.
size	Integer	Tamanho da mensagem.

Parâmetro	Tipo	Descrição
timestamp	Long	Tempo de produção da mensagem. O valor é um carimbo de data/hora UNIX, em ms.
huge_message	Boolean	Bandeira de big data.
message_offset	Long	Deslocamento da mensagem.
message_id	String	ID da mensagem.
app_id	String	ID da aplicação.
tag	String	Rótulo da mensagem.

Código de status: 400

Tabela 5-192 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Código de status: 403

Tabela 5-193 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
error_code	String	Código de erro.
error_msg	String	Descrição do erro.

Exemplo de solicitações

- Consulta do deslocamento da mensagem.
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/messages?asc=false&end_time=1608609032042&limit=10&offset=0&start_time=1608608432042&topic=topic-test
- Consulta do conteúdo da mensagem.
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/messages?download=false&message_offset=0&partition=0&topic=topic-test

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "messages" : [ {
```



```
"topic" : "topic-test",
"partition" : 0,
"value" : "hello world",
"size" : 21,
"timestamp" : 1607598463502,
"huge_message" : false,
"message_offset" : 4,
"message_id" : "",
"app_id" : "",
"tag" : ""
} ],
"total" : 1,
"size" : 1
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceMessagesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))
            .build();
        ShowInstanceMessagesRequest request = new ShowInstanceMessagesRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowInstanceMessagesResponse response =
client.showInstanceMessages(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceMessagesRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_messages(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowInstanceMessagesRequest{
    request.InstanceId = "{instance_id}"
}
response, err := client.ShowInstanceMessages(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia [Código de exemplo no API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.
400	Parâmetros inválidos.
403	Falha na autenticação.

Códigos de erro

Consulte [Códigos de erro](#).

5.7.2 Consulta de uma mensagem com um deslocamento especificado

Função

Essa API é usada para consultar uma mensagem com um deslocamento especificado.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/message

Tabela 5-194 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico. Um nome de tópico deve começar com uma letra e só pode conter letras, hifens (-), sublinhados (_) e dígitos.
partition	Sim	Integer	Número da partição.

Tabela 5-195 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
message_offset	Sim	String	Deslocamento da mensagem.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-196 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
message	Array of ShowPartitionMessageEntity objects	Lista de mensagens.

Tabela 5-197 ShowPartitionMessageEntity

Parâmetro	Tipo	Descrição
key	String	Key de mensagem.
value	String	Conteúdo da mensagem.
topic	String	Nome do tópico.
partition	Integer	Número da partição.
message_offset	Long	Deslocamento da mensagem.
size	Integer	Tamanho da mensagem em bytes.
timestamp	Long	Hora em que uma mensagem é criada. O valor é um carimbo de data/hora Unix. A unidade é milissegundo.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/message?message_offset={message_offset}
```

Exemplo de respostas

Código de status: 200

A mensagem com o deslocamento especificado foi consultada com sucesso.

```
{
  "message" : [ {
    "topic" : "mytest",
    "partition" : 0,
    "message_offset" : 7,
    "key" : null,
    "value" : "kasjdf",
    "size" : 6,
    "timestamp" : 1568125036045
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class ShowPartitionMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowPartitionMessageRequest request = new ShowPartitionMessageRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        request.withPartition({partition});
        try {
            ShowPartitionMessageResponse response =
client.showPartitionMessage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
        client = KafkaClient.new_builder() \
```

```
.with_credentials(credentials) \
.with_region(KafkaRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ShowPartitionMessageRequest()
    request.instance_id = "{instance_id}"
    request.topic = "{topic}"
    request.partition = {partition}
    response = client.show_partition_message(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionMessageRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    request.Partition = int32({partition})
    response, err := client.ShowPartitionMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A mensagem com o deslocamento especificado foi consultada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.7.3 Consulta de uma mensagem com um período de tempo especificado

Função

Essa API é usada para consultar uma mensagem com um período de tempo especificado.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/messages

Tabela 5-198 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico. Um nome de tópico deve começar com uma letra e só pode conter letras, hifens (-), sublinhados (_) e dígitos.

Tabela 5-199 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
start_time	Não	String	Hora de início da consulta como um carimbo de data/hora Unix. Valor padrão: 0 .
end_time	Não	String	Hora de término da consulta, como um carimbo de data/hora Unix. Valor padrão: hora atual do sistema.
limit	Não	Integer	Número de mensagens retornadas em uma página. Valor padrão: 10 .
offset	Não	Integer	Deslocamento, que é a posição onde a consulta começa. O valor deve ser maior ou igual a 0.
partition	Não	String	Número da partição. O valor padrão é -1 , indicando que todas as partições são consultadas.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-200 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
messages	Array of messages objects	Lista de mensagens.
messages_count	Integer	Número total de mensagens.
offsets_count	Integer	Número total de páginas.
offset	Integer	Página atual.

Tabela 5-201 messages

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.

Parâmetro	Tipo	Descrição
partition	Integer	Número da partição.
message_offset	Long	Deslocamento da mensagem.
size	Integer	Tamanho da mensagem em bytes.
timestamp	Long	Hora em que uma mensagem é criada. O valor é um carimbo de data/hora Unix. A unidade é milissegundo.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/messages
```

Exemplo de respostas

Código de status: 200

A mensagem com o período de tempo especificado foi consultada com sucesso.

```
{
  "messages" : [ {
    "topic" : "mytest",
    "partition" : 0,
    "message_offset" : 7,
    "size" : 6,
    "timestamp" : 1568125036045
  } ],
  "messages_count" : 1,
  "offsets_count" : 1,
  "offset" : 1
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowMessagesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

ShowMessagesRequest request = new ShowMessagesRequest();
request.withInstanceId("{instance_id}");
request.withTopic("{topic}");
try {
    ShowMessagesResponse response = client.showMessages(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMessagesRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        response = client.show_messages(request)
```

```
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMessagesRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    response, err := client.ShowMessages(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A mensagem com o período de tempo especificado foi consultada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.7.4 Consulta do deslocamento da mensagem mais antiga em uma partição

Função

Essa API é usada para consultar o deslocamento da mensagem mais antiga em uma partição.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/beginning-message

Tabela 5-202 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico. Um nome de tópico deve começar com uma letra e só pode conter letras, hifens (-), sublinhados (_) e dígitos.
partition	Sim	Integer	Número da partição.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-203 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.
partition	Integer	Número da partição.
offset	Integer	Deslocamento da última mensagem.
timestamp	Long	Hora em que uma mensagem é criada. O valor é um carimbo de data/hora Unix. A unidade é milissegundo.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/beginning-message
```

Exemplo de respostas

Código de status: 200

O deslocamento da mensagem mais antiga em uma partição é consultado com sucesso.

```
{
  "topic" : "mytest",
  "partition" : 0,
  "offset" : 9,
  "timestamp" : 1568125039164
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowPartitionBeginningMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
```

```
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();

    ShowPartitionBeginningMessageRequest request = new
    ShowPartitionBeginningMessageRequest();
    request.withInstanceId("{instance_id}");
    request.withTopic("{topic}");
    request.withPartition({partition});
    try {
        ShowPartitionBeginningMessageResponse response =
    client.showPartitionBeginningMessage(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ShowPartitionBeginningMessageRequest()
request.instance_id = "{instance_id}"
request.topic = "{topic}"
request.partition = {partition}
response = client.show_partition_beginning_message(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionBeginningMessageRequest{
        InstanceId = "{instance_id}"
        Topic = "{topic}"
        Partition = int32({partition})
    }
    response, err := client.ShowPartitionBeginningMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	O deslocamento da mensagem mais antiga em uma partição é consultado com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.7.5 Consulta do deslocamento da última mensagem em uma partição

Função

Essa API é usada para consultar o deslocamento da última mensagem em uma partição.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/end-message

Tabela 5-204 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico. Um nome de tópico deve começar com uma letra e só pode conter letras, hifens (-), sublinhados (_) e dígitos.
partition	Sim	Integer	Número da partição.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-205 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.
partition	Integer	Número da partição.
offset	Integer	Deslocamento da última mensagem.
timestamp	Long	Hora em que uma mensagem é criada. O valor é um carimbo de data/hora Unix. A unidade é milissegundo.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/  
{topic}/partitions/{partition}/end-message
```

Exemplo de respostas

Código de status: 200

O deslocamento da última mensagem em uma partição é consultado com sucesso.

```
{  
  "topic" : "mytest",  
  "partition" : 0,  
  "offset" : 9,  
  "timestamp" : 1568125039164  
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ShowPartitionEndMessageSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
        plaintext, which has great security risks. It is recommended that the AK and SK  
        be stored in ciphertext in configuration files or environment variables and
```

```
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();

    ShowPartitionEndMessageRequest request = new
ShowPartitionEndMessageRequest();
    request.withInstanceId("{instance_id}");
    request.withTopic("{topic}");
    request.withPartition({partition});
    try {
        ShowPartitionEndMessageResponse response =
client.showPartitionEndMessage(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ShowPartitionEndMessageRequest()
request.instance_id = "{instance_id}"
request.topic = "{topic}"
request.partition = {partition}
response = client.show_partition_end_message(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionEndMessageRequest{
        request.InstanceId = "{instance_id}"
        request.Topic = "{topic}"
        request.Partition = int32({partition})
    }
    response, err := client.ShowPartitionEndMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	O deslocamento da última mensagem em uma partição é consultado com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.8 Gerenciamento de tarefas em segundo plano

5.8.1 Listagem das tarefas em segundo plano

Função

Essa API é usada para listar tarefas em segundo plano de uma instância.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/tasks

Tabela 5-206 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Tabela 5-207 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
start	Não	Integer	ID da tarefa onde a consulta é iniciada.
limit	Não	Integer	Número de tarefas a serem consultadas.

Parâmetro	Obrigatório	Tipo	Descrição
begin_time	Não	String	Hora da tarefa onde a consulta é iniciada. O formato é YYYYMMDDHHmmss.
end_time	Não	String	Hora da tarefa onde a consulta termina. O formato é YYYYMMDDHHmmss.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-208 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
task_count	String	Número de tarefas.
tasks	Array of tasks objects	Lista de tarefas.

Tabela 5-209 tasks

Parâmetro	Tipo	Descrição
id	String	ID da tarefa.
name	String	Nome da tarefa.
user_name	String	Nome de usuário.
user_id	String	ID do usuário.
params	String	Parâmetros da tarefa.
status	String	Status da tarefa.
created_at	String	Hora de início.
updated_at	String	Hora de término.

Exemplo de solicitações

```
'GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/tasks?
start={start}&limit={limit}&begin_time={begin_time}&end_time={end_time}'
```

Exemplo de respostas

Código de status: 200

As tarefas em segundo plano são listadas com sucesso.

```
{
  "task_count" : "4",
  "tasks" : [ {
    "id" : "8abfa7b372160bfd0172165864064079",
    "name" : "modifyAutoTopic",
    "user_name" : "paas_dms",
    "user_id" : "3df5acbc24a54fadb62a043c9000a307",
    "params" : "{\"old_auto_status\":true,\"new_auto_status\":false}",
    "status" : "EXECUTING",
    "created_at" : "2020-05-15T03:19:51.046Z",
    "updated_at" : "2020-05-15T03:19:51.065Z"
  }, {
    "id" : "8abfa7b372160bfd017216560af83e6e",
    "name" : "changeRetentionPolicy",
    "user_name" : "paas_dms",
    "user_id" : "3df5acbc24a54fadb62a043c9000a307",
    "params" : "{\"new_retention_policy\":\"produce_reject\", \"origin_retention_policy\":\"time_base\"}",
    "status" : "SUCCESS",
    "created_at" : "2020-05-15T03:17:17.176Z",
    "updated_at" : "2020-05-15T03:17:22.162Z"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListBackgroundTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
    }
}
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ListBackgroundTasksRequest request = new ListBackgroundTasksRequest();
request.withInstanceId("{instance_id}");
try {
    ListBackgroundTasksResponse response =
client.listBackgroundTasks(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBackgroundTasksRequest()
        request.instance_id = "{instance_id}"
        response = client.list_background_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```



```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBackgroundTasksRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListBackgroundTasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As tarefas em segundo plano são listadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.8.2 Consulta das tarefas em segundo plano

Função

Essa API é usada para consultar uma tarefa em segundo plano especificada.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/tasks/{task_id}

Tabela 5-210 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-211 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
task_count	String	Número de tarefas.
tasks	Array of tasks objects	Lista de tarefas.

Tabela 5-212 tasks

Parâmetro	Tipo	Descrição
id	String	ID da tarefa.
name	String	Nome da tarefa.

Parâmetro	Tipo	Descrição
user_name	String	Nome de usuário.
user_id	String	ID do usuário.
params	String	Parâmetros da tarefa.
status	String	Status da tarefa.
created_at	String	Hora de início.
updated_at	String	Hora de término.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/tasks/{task_id}
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "task_count" : "1",
  "tasks" : [ {
    "id" : "8abfa7b272adc5b40172b73130065ae7",
    "name" : "bindInstancePublicIp",
    "user_name" : "paas_dms",
    "user_id" : "3df5acbc24a54fadb62a043c9000a307",
    "params" : "{\"public_ip_id\":\"1aea7aed-e7d8-40ea-b3de-6f3ee9d5db9f\", \"public_ip_address\":\"100.93.2.18\", \"enable_public_ip\":true}",
    "status" : "SUCCESS",
    "created_at" : "2020-06-15T08:55:53.606Z",
    "updated_at" : "2020-06-15T08:55:56.600Z"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowBackgroundTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();

    ShowBackgroundTaskRequest request = new ShowBackgroundTaskRequest();
    request.withInstanceId("{instance_id}");
    request.withTaskId("{task_id}");
    try {
        ShowBackgroundTaskResponse response =
client.showBackgroundTask(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ShowBackgroundTaskRequest()
request.instance_id = "{instance_id}"
request.task_id = "{task_id}"
response = client.show_background_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBackgroundTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowBackgroundTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.8.3 Exclusão de uma tarefa em segundo plano

Função

Essa API é usada para excluir uma tarefa em segundo plano especificada.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

DELETE /v2/{project_id}/instances/{instance_id}/tasks/{task_id}

Tabela 5-213 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
task_id	Sim	String	ID da tarefa.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Exclusão de uma tarefa em segundo plano especificada

```
DELETE https://{endpoint}/v2/{project_id}/instances/{instance_id}/tasks/{task_id}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteBackgroundTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBackgroundTaskRequest request = new DeleteBackgroundTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            DeleteBackgroundTaskResponse response =
client.deleteBackgroundTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBackgroundTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.delete_background_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```



```
client := kafka.NewKafkaClient(  
    kafka.KafkaClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.DeleteBackgroundTaskRequest{  
    request.InstanceId = "{instance_id}"  
    request.TaskId = "{task_id}"  
    response, err := client.DeleteBackgroundTask(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	A tarefa em segundo plano foi excluída com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.9 Gerenciamento de tags

5.9.1 Adição ou exclusão das tags em lote

Função

Essa API é usada para adicionar ou excluir tags de instância em lotes.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

POST /v2/{project_id}/kafka/{instance_id}/tags/action

Tabela 5-214 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Tabela 5-215 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
action	Não	String	Operação. Somente letras minúsculas são suportadas. <ul style="list-style-type: none"> ● create: as tags são criadas. ● delete: as tags são excluídas.
tags	Não	Array of TagEntity objects	Lista de tags.

Tabela 5-216 TagEntity

Parâmetro	Obrigatório	Tipo	Descrição
key	Não	String	Chave de tag, que: <ul style="list-style-type: none"> ● Deve ser especificada. ● Deve ser exclusiva para a mesma instância. ● Pode conter de 1 a 128 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>_ :=+ - @</code> ● Não pode começar ou terminar com um espaço.

Parâmetro	Obrigatório	Tipo	Descrição
value	Não	String	Valor da tag. <ul style="list-style-type: none"> ● Pode conter de 0 a 255 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code> ● Não pode começar ou terminar com um espaço.

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Criação de tags de instância com chaves de tag key1 e key2 e valores de tag value1 e value2

```
POST https://{endpoint}/v2/{project_id}/kafka/{instance_id}/tags/action
{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

Exemplo de respostas

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Criação de tags de instância com chaves de tag key1 e key2 e valores de tag value1 e value2

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;
```

```
public class BatchCreateOrDeleteKafkaTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchCreateOrDeleteKafkaTagRequest request = new
BatchCreateOrDeleteKafkaTagRequest();
        request.withInstanceId("{instance_id}");
        BatchCreateOrDeleteTagReq body = new BatchCreateOrDeleteTagReq();
        List<TagEntity> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagEntity()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new TagEntity()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchCreateOrDeleteTagReq.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            BatchCreateOrDeleteKafkaTagResponse response =
client.batchCreateOrDeleteKafkaTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Criação de tags de instância com chaves de tag key1 e key2 e valores de tag value1 e value2

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateOrDeleteKafkaTagRequest()
        request.instance_id = "{instance_id}"
        listTagsbody = [
            TagEntity(
                key="key1",
                value="value1"
            ),
            TagEntity(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchCreateOrDeleteTagReq(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_or_delete_kafka_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Criação de tags de instância com chaves de tag key1 e key2 e valores de tag value1 e value2

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```

sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchCreateOrDeleteKafkaTagRequest{}
request.InstanceId = "{instance_id}"
keyTags:= "key1"
valueTags:= "value1"
keyTags1:= "key2"
valueTags1:= "value2"
var listTagsbody = []model.TagEntity{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
    {
        Key: &keyTags1,
        Value: &valueTags1,
    },
}
actionBatchCreateOrDeleteTagReq:=
model.GetBatchCreateOrDeleteTagReqActionEnum().CREATE
request.Body = &model.BatchCreateOrDeleteTagReq{
    Tags: &listTagsbody,
    Action: &actionBatchCreateOrDeleteTagReq,
}
response, err := client.BatchCreateOrDeleteKafkaTag(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
204	As tags são adicionadas ou excluídas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.9.2 Listagem das tags de uma instância

Função

Essa API é usada para consultar tags de instância.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/kafka/{instance_id}/tags

Tabela 5-217 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-218 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
tags	Array of TagEntity objects	Lista de tags.

Tabela 5-219 TagEntity

Parâmetro	Tipo	Descrição
key	String	Chave de tag, que: <ul style="list-style-type: none"> ● Deve ser especificada. ● Deve ser exclusiva para a mesma instância. ● Pode conter de 1 a 128 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code> ● Não pode começar ou terminar com um espaço.
value	String	Valor da tag. <ul style="list-style-type: none"> ● Pode conter de 0 a 255 caracteres. ● Pode conter letras, dígitos, espaços e caracteres especiais <code>._:=-@</code>

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/kafka/{instance_id}/tags
```

Exemplo de respostas

Código de status: 200

As tags de instância são listadas com sucesso.

```
{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaTagsSolution {
```



```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();

    ShowKafkaTagsRequest request = new ShowKafkaTagsRequest();
    request.withInstanceId("{instance_id}");
    try {
        ShowKafkaTagsResponse response = client.showKafkaTags(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ShowKafkaTagsRequest()
request.instance_id = "{instance_id}"
response = client.show_kafka_tags(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    // plaintext, which has great security risks. It is recommended that the AK and SK
    // be stored in ciphertext in configuration files or environment variables and
    // decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    // authentication. Before running this example, set environment variables
    // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKafkaTagsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowKafkaTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As tags de instância são listadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.9.3 Listagem das tags de um projeto

Função

Essa API é usada para consultar tags de projeto.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/kafka/tags

Tabela 5-220 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-221 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
tags	Array of TagMultyValu eEntity objects	Lista de tags.

Tabela 5-22 TagMultyValueEntity

Parâmetro	Tipo	Descrição
key	String	Chave da tag.
values	Array of strings	Valor da tag.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/kafka/tags
```

Exemplo de respostas

Código de status: 200

As tags do projeto são listadas com sucesso.

```
{
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value-test", "value1" ]
  }, {
    "key" : "key2",
    "values" : [ "value2" ]
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaProjectTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowKafkaProjectTagsRequest request = new ShowKafkaProjectTagsRequest();
    try {
        ShowKafkaProjectTagsResponse response =
client.showKafkaProjectTags(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowKafkaProjectTagsRequest()
        response = client.show_kafka_project_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKafkaProjectTagsRequest{}
    response, err := client.ShowKafkaProjectTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As tags do projeto são listadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.10 Outras APIs

5.10.1 Listagem de janelas de tempo de manutenção

Função

Essa API é usada para consultar a hora de início e a hora de término das janelas de tempo de manutenção.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/instances/maintain-windows

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-223 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
maintain_windows	Array of MaintainWindowsEntity objects	Lista de janelas de tempo de manutenção suportadas.

Tabela 5-224 MaintainWindowsEntity

Parâmetro	Tipo	Descrição
default	Boolean	Se a janela de tempo de manutenção está definida para o segmento de tempo padrão.
end	String	Hora de término da janela de tempo de manutenção.
begin	String	Hora de início da janela de tempo de manutenção.
seq	Integer	Número de sequência.

Exemplo de solicitações

```
GET https://{endpoint}/v2/instances/maintain-windows
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "maintain_windows" : [ {
    "default" : false,
    "seq" : 1,
    "begin" : "22",
    "end" : "02"
  }, {
    "default" : true,
    "seq" : 2,
    "begin" : "02",
    "end" : "06"
  }, {
    "default" : false,
    "seq" : 3,
    "begin" : "06",
    "end" : "10"
  }, {
    "default" : false,
    "seq" : 4,
    "begin" : "10",
    "end" : "14"
  }, {
    "default" : false,
    "seq" : 5,
    "begin" : "14",
    "end" : "18"
  }, {
    "default" : false,
    "seq" : 6,
    "begin" : "18",
    "end" : "22"
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowMaintainWindowsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
```



```
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowMaintainWindowsRequest request = new ShowMaintainWindowsRequest();
try {
    ShowMaintainWindowsResponse response =
client.showMaintainWindows(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMaintainWindowsRequest()
        response = client.show_maintain_windows(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMaintainWindowsRequest{}
    response, err := client.ShowMaintainWindows(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

5.10.2 Listagem de informações de AZ

Função

Essa API é usada para consultar o ID da AZ para criar uma instância.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/available-zones

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-225 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
region_id	String	ID da região.
available_zones	Array of AvailableZonesResp objects	Matriz de AZs.

Tabela 5-226 AvailableZonesResp

Parâmetro	Tipo	Descrição
soldOut	Boolean	Indica se os recursos estão esgotados.
id	String	ID da AZ.
code	String	Código de AZ.
name	String	Nome de AZ.
port	String	Porta de AZ.
resource_availability	String	Indica se a AZ tem recursos disponíveis.
default_az	Boolean	Indica se a AZ é a AZ padrão.
remain_time	Long	Tempo restante.

Parâmetro	Tipo	Descrição
ipv6_enable	Boolean	Indica se há suporte para IPv6.

Exemplo de solicitações

```
GET https://{endpoint}/v2/available-zones
```

Exemplo de respostas

Código de status: 200

As informações da AZ foram consultadas com sucesso.

- A instância é consultada. Um exemplo de resultado retornado é o seguinte.

```
{
  "region_id" : "xxx",
  "available_zones" : [ {
    "soldOut" : false,
    "id" : "d539378ec1314c85b76fef3f7071458",
    "code" : "xxx",
    "name" : "AZ 2.",
    "port" : "8003",
    "resource_availability" : "true",
    "default_az" : true,
    "remain_time" : 9223372036854776000,
    "ipv6_enable" : false
  }, {
    "soldOut" : false,
    "id" : "9f1c5806706d4c1fb0eb72f0a9b18c77",
    "code" : "xxx",
    "name" : "AZ 3.",
    "port" : "443",
    "resource_availability" : "true",
    "default_az" : false,
    "remain_time" : 9223372036854776000,
    "ipv6_enable" : false
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListAvailableZonesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
```

```
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ListAvailableZonesRequest request = new ListAvailableZonesRequest();
try {
    ListAvailableZonesResponse response =
client.listAvailableZones(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAvailableZonesRequest()
        response = client.list_available_zones(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAvailableZonesRequest{}
    response, err := client.ListAvailableZones(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As informações da AZ foram consultadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.10.3 Consulta da lista de especificações do produto

Função

Essa API é usada para consultar a lista de especificações do produto.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{engine}/products

Tabela 5-227 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
engine	Sim	String	Mecanismo de mensagens. Padrão: kafka

Tabela 5-228 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
product_id	Não	String	ID do produto.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-229 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
engine	String	Mecanismo de mensagens do DMS.
versions	Array of strings	Versões suportadas.

Parâmetro	Tipo	Descrição
products	Array of ListEngineProductsEntity objects	Detalhes da especificação do produto.

Tabela 5-230 ListEngineProductsEntity

Parâmetro	Tipo	Descrição
type	String	Tipo de produto. Atualmente, os tipos de nó único e cluster são suportados.
product_id	String	ID do produto.
ecs_flavor_id	String	Flavor do ECS.
billing_code	String	Modo de cobrança.
arch_types	Array of strings	Arquitetura da CPU.
charging_mode	Array of strings	Modo de cobrança. monthly : anual/mensal; hourly : pagamento por uso
ios	Array of ListEngineIosEntity objects	Lista de tipos de I/O de disco suportados.
support_features	Array of ListEngineSupportFeaturesEntity objects	Lista de recursos suportados por instâncias das especificações atuais.
properties	ListEnginePropertiesEntity object	Atributo de instâncias das especificações atuais.

Tabela 5-231 ListEngineIosEntity

Parâmetro	Tipo	Descrição
io_spec	String	Código de I/O do disco.
type	String	Tipo de disco.
available_zones	Array of strings	AZs disponíveis.
unavailable_zones	Array of strings	AZs indisponíveis.

Tabela 5-232 ListEngineSupportFeaturesEntity

Parâmetro	Tipo	Descrição
name	String	Nome de recursos.
properties	ListEngineSupportFeaturesPropertiesEntity object	Descrição dos recursos suportados pela instância.

Tabela 5-233 ListEngineSupportFeaturesPropertiesEntity

Parâmetro	Tipo	Descrição
max_task	String	Número máximo de tarefas de despejo.
min_task	String	Número mínimo de tarefas de despejo.
max_node	String	Número máximo de nós de despejo.
min_node	String	Número mínimo de nós de despejo.

Tabela 5-234 ListEnginePropertiesEntity

Parâmetro	Tipo	Descrição
max_partition_per_broker	String	Número máximo de partições de cada corretor.
max_broker	String	Número máximo de corretores.
max_storage_per_node	String	Espaço máximo de armazenamento de cada corretor. A unidade é GB.
max_consumer_per_broker	String	Número máximo de consumidores de cada corretor.
min_broker	String	Número mínimo de corretores.
max_bandwidth_per_broker	String	Largura de banda máxima de cada corretor.
min_storage_per_node	String	Espaço mínimo de armazenamento de cada corretor. A unidade é GB.
max_tps_per_broker	String	TPS máximo de cada corretor.
product_alias	String	Alias de product_id .

Exemplo de solicitações

```
GET https://{endpoint}/v2/kafka/products
```

Exemplo de respostas

Código de status: 200

As especificações do produto são listadas com sucesso.

- Bem-sucedido.

```
{
  "engine" : "kafka",
  "versions" : [ "1.1.0", "2.3.0" ],
  "products" : [ {
    "type" : "cluster",
    "product_id" : "c6.2u4g.cluster",
    "ecs_flavor_id" : "c6.large.2",
    "billing_code" : "dms.platinum.c6",
    "arch_types" : [ "X86" ],
    "charging_mode" : [ "monthly", "hourly" ],
    "ios" : [ {
      "io_spec" : "dms.physical.storage.high.v2",
      "type" : "evs",
      "available_zones" : [ "xxx", "xxx" ],
      "unavailable_zones" : [ "xxx", "xxx" ]
    }, {
      "io_spec" : "dms.physical.storage.ultra.v2",
      "type" : "evs",
      "available_zones" : [ "xxx", "xxx" ],
      "unavailable_zones" : [ "xxx", "xxx" ]
    } ],
    "support_features" : [ {
      "name" : "connector_obs",
      "properties" : {
        "max_task" : "10",
        "max_node" : "10",
        "min_task" : "1",
        "min_node" : "2"
      }
    } ],
    "properties" : {
      "max_partition_per_broker" : "250",
      "max_broker" : "30",
      "max_storage_per_node" : "10000",
      "max_consumer_per_broker" : "4000",
      "min_broker" : "3",
      "max_bandwidth_per_broker" : "100",
      "min_storage_per_node" : "200",
      "max_tps_per_broker" : "30000",
      "product_alias" : "kafka.2u4g.cluster"
    }
  }, {
    "type" : "cluster",
    "product_id" : "c6.4u8g.cluster",
    "ecs_flavor_id" : "c6.xlarge.2",
    "billing_code" : "dms.platinum.c6",
    "arch_types" : [ "X86" ],
    "charging_mode" : [ "monthly", "hourly" ],
    "ios" : [ {
      "io_spec" : "dms.physical.storage.high.v2",
      "type" : "evs",
      "available_zones" : [ "xxx", "xxx" ],
      "unavailable_zones" : [ "xxx", "xxx" ]
    }, {
      "io_spec" : "dms.physical.storage.ultra.v2",
      "type" : "evs",
      "available_zones" : [ "xxx", "xxx" ],

```

```
    "unavailable_zones" : [ "xxx", "xxx" ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "500",
    "max_broker" : "30",
    "max_storage_per_node" : "20000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "400",
    "max_tps_per_broker" : "100000",
    "product_alias" : "kafka.4u8g.cluster"
  }
}, {
  "type" : "cluster",
  "product_id" : "c6.8u16g.cluster",
  "ecs_flavor_id" : "c6.2xlarge.2",
  "billing_code" : "dms.platinum.c6",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "1000",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "800",
    "max_tps_per_broker" : "150000",
    "product_alias" : "kafka.8u16g.cluster"
  }
}, {
  "type" : "cluster",
  "product_id" : "c6.12u24g.cluster",
  "ecs_flavor_id" : "c6.3xlarge.2",
  "billing_code" : "dms.platinum.c6",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
```

```

        "type" : "evs",
        "available_zones" : [ "xxx", "xxx" ],
        "unavailable_zones" : [ "xxx", "xxx" ]
    }, {
        "io_spec" : "dms.physical.storage.ultra.v2",
        "type" : "evs",
        "available_zones" : [ "xxx", "xxx" ],
        "unavailable_zones" : [ "xxx", "xxx" ]
    } ],
    "support_features" : [ {
        "name" : "connector_obs",
        "properties" : {
            "max_task" : "10",
            "max_node" : "10",
            "min_task" : "1",
            "min_node" : "2"
        }
    } ],
    "properties" : {
        "max_partition_per_broker" : "1500",
        "max_broker" : "30",
        "max_storage_per_node" : "30000",
        "max_consumer_per_broker" : "4000",
        "min_broker" : "3",
        "max_bandwidth_per_broker" : "100",
        "min_storage_per_node" : "1200",
        "max_tps_per_broker" : "200000",
        "product_alias" : "kafka.12u24g.cluster"
    }
}, {
    "type" : "cluster",
    "product_id" : "c6.16u32g.cluster",
    "ecs_flavor_id" : "c6.4xlarge.2",
    "billing_code" : "dms.platinum.c6",
    "arch_types" : [ "X86" ],
    "charging_mode" : [ "monthly", "hourly" ],
    "ios" : [ {
        "io_spec" : "dms.physical.storage.high.v2",
        "type" : "evs",
        "available_zones" : [ "xxx", "xxx" ],
        "unavailable_zones" : [ "xxx", "xxx" ]
    }, {
        "io_spec" : "dms.physical.storage.ultra.v2",
        "type" : "evs",
        "available_zones" : [ "xxx", "xxx" ],
        "unavailable_zones" : [ "xxx", "xxx" ]
    } ],
    "support_features" : [ {
        "name" : "connector_obs",
        "properties" : {
            "max_task" : "10",
            "max_node" : "10",
            "min_task" : "1",
            "min_node" : "2"
        }
    } ],
    "properties" : {
        "max_partition_per_broker" : "2000",
        "max_broker" : "30",
        "max_storage_per_node" : "30000",
        "max_consumer_per_broker" : "4000",
        "min_broker" : "3",
        "max_bandwidth_per_broker" : "100",
        "min_storage_per_node" : "1600",
        "max_tps_per_broker" : "250000",
        "product_alias" : "kafka.16u32g.cluster"
    }
} ]
}

```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListEngineProductsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEngineProductsRequest request = new ListEngineProductsRequest();
request.withEngine(ListEngineProductsRequest.EngineEnum.fromValue("{engine}"));
        try {
            ListEngineProductsResponse response =
client.listEngineProducts(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListEngineProductsRequest()
        request.engine = "{engine}"
        response = client.list_engine_products(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEngineProductsRequest{}
    request.Engine = model.GetListEngineProductsRequestEngineEnum().ENGINE
    response, err := client.ListEngineProducts(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```
} else {  
    fmt.Println(err)  
}  
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	As especificações do produto são listadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

5.10.4 Consulta das dimensões de monitoramento de instâncias do Kafka

Função

Essa API é usada para consultar dimensões de monitoramento de instâncias do Kafka.

Método de chamada

Para obter detalhes, consulte [Chamada de APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/ces-hierarchy

Tabela 5-235 Parâmetros de caminho

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obtê-lo, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 5-236 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
dimensions	Array of dimensions objects	Dimensões de monitoramento.
instance_ids	Array of instance_ids objects	Informações da instância.
nodes	Array of nodes objects	Informações do corretor.
queues	Array of queues objects	Informação do tópico.
groups	Array of groups objects	Informações do grupo de consumidores.

Tabela 5-237 dimensions

Parâmetro	Tipo	Descrição
name	String	Nome da dimensão de monitoramento.
metrics	Array of strings	Nome da métrica.
key_name	Array of strings	Chave usada para a consulta de monitoramento.
dim_router	Array of strings	Rota da dimensão de monitoramento.
children	Array of children objects	Lista de dimensões secundárias.

Tabela 5-238 children

Parâmetro	Tipo	Descrição
name	String	Nome da dimensão secundária.
metrics	Array of strings	Métricas na dimensão secundária.
key_name	Array of strings	Chave usada para a consulta de monitoramento.
dim_router	Array of strings	Rota da dimensão de monitoramento.

Tabela 5-239 instance_ids

Parâmetro	Tipo	Descrição
name	String	ID da instância.

Tabela 5-240 nodes

Parâmetro	Tipo	Descrição
name	String	Nome do corretor.

Tabela 5-241 queues

Parâmetro	Tipo	Descrição
name	String	Nome do tópico.
partitions	Array of partitions objects	Lista de partições.

Tabela 5-242 partitions

Parâmetro	Tipo	Descrição
name	String	Nome da partição.

Tabela 5-243 groups

Parâmetro	Tipo	Descrição
name	String	Nome do grupo de consumidores.
queues	Array of queues objects	Informação do tópico.

Tabela 5-244 queues

Parâmetro	Tipo	Descrição
name	String	Nome do tópico.
partitions	Array of partitions objects	Informação da partição.

Tabela 5-245 partitions

Parâmetro	Tipo	Descrição
name	String	Nome da partição.

Exemplo de solicitações

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/ces-hierarchy
```

Exemplo de respostas

Código de status: 200

A consulta foi bem-sucedida.

```
{
  "dimensions" : [ {
    "name" : "kafka_instance_id",
    "metrics" : [ "current_partitions", "current_topics", "group_messages" ],
    "key_name" : [ "instance_ids" ],
    "dim_router" : [ "kafka_instance_id" ]
  }, {
    "name" : "kafka_broker",
    "metrics" : [ "broker_data_size", "broker_messages_in_rate",
"broker_bytes_out_rate", "broker_bytes_in_rate", "broker_produce_mean",
"broker_fetch_mean" ],
    "key_name" : [ "nodes" ],
    "dim_router" : [ "kafka_instance_id", "kafka_broker" ]
  }, {
    "name" : "kafka_rest",
    "metrics" : [ "rest_produce_success", "rest_produce_failed",
"rest_produce_latency", "rest_produce_msg_num", "rest_produce_flow",
"rest_consume_success", "rest_consume_failed", "rest_consume_latency",
"rest_consume_msg_num", "rest_consume_flow", "rest_commit_success",
"rest_commit_failed", "rest_commit_latency", "rest_commit_msg_num",
"rest_commit_flow" ],
    "key_name" : [ "nodes" ],
    "dim_router" : [ "kafka_instance_id", "kafka_rest" ]
  }, {
    "name" : "kafka_topics",
    "metrics" : [ "topic_data_size", "topic_messages_in_rate",
"topic_bytes_out_rate", "topic_bytes_in_rate", "topic_messages" ],
    "key_name" : [ "queues" ],
    "dim_router" : [ "kafka_instance_id", "kafka_topics" ],
    "children" : [ {
      "name" : "kafka_partitions",
      "metrics" : [ "produced_messages", "partition_messages" ],
      "key_name" : [ "queues", "partitions" ],
      "dim_router" : [ "kafka_instance_id", "kafka_topics", "kafka_partitions" ]
    } ]
  }, {
    "name" : "kafka_groups_partitions",
    "metrics" : [ "messages_consumed", "messages_remaining" ],
    "key_name" : [ "groups", "queues", "partitions" ],
    "dim_router" : [ "kafka_instance_id", "kafka_groups", "kafka_groups_topics",
"kafka_groups_partitions" ]
  } ],
  "instance_ids" : [ {
    "name" : "68f3f6a0-3741-453b-bda9-a6ff6b5bb6f7"
  } ],
  "nodes" : [ {
    "name" : "broker-0"
  }, {
    "name" : "broker-1"
  } ]
}
```

```
    }, {
      "name" : "broker-2"
    } ],
    "queues" : [ {
      "name" : "aaaa",
      "partitions" : [ {
        "name" : "0"
      } ]
    }, {
      "name" : "mytest",
      "partitions" : [ {
        "name" : "0"
      }, {
        "name" : "1"
      }, {
        "name" : "2"
      } ]
    }, {
      "name" : "topic-84234378",
      "partitions" : [ {
        "name" : "0"
      }, {
        "name" : "1"
      }, {
        "name" : "2"
      } ]
    } ],
    "groups" : [ {
      "name" : "test-consumer-group",
      "queues" : [ {
        "name" : "mytest",
        "partitions" : [ {
          "name" : "0"
        }, {
          "name" : "1"
        }, {
          "name" : "2"
        } ]
      } ]
    } ]
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowCesHierarchySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
```

```
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowCesHierarchyRequest request = new ShowCesHierarchyRequest();
request.withInstanceId("{instance_id}");
try {
    ShowCesHierarchyResponse response = client.showCesHierarchy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCesHierarchyRequest()
        request.instance_id = "{instance_id}"
        response = client.show_ces_hierarchy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowCesHierarchyRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowCesHierarchy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Mais

Para obter o código de exemplo do SDK de mais linguagens de programação, consulte a guia Código de exemplo no [API Explorer](#). O código de exemplo do SDK pode ser gerado automaticamente.

Códigos de status

Código de status	Descrição
200	A consulta foi bem-sucedida.

Códigos de erro

Consulte [Códigos de erro](#).

6 Permissões e ações suportadas

Este capítulo descreve o gerenciamento refinado de permissões para suas instâncias do Kafka. Se sua HUAWEI ID não precisar de usuários individuais do IAM, você poderá pular este capítulo.

Por padrão, os novos usuários do IAM não têm permissões atribuídas. Você precisa adicionar um usuário a um ou mais grupos e anexar políticas de permissões ou funções a esses grupos. Os usuários herdam permissões dos grupos aos quais são adicionados e podem executar operações especificadas em serviços de nuvem com base nas permissões.

Você pode conceder permissões aos usuários usando **funções** e **políticas**. As funções são um tipo de mecanismo de autorização de alta granularidade que define permissões relacionadas às responsabilidades do usuário. As políticas definem permissões baseadas em API para operações em recursos específicos sob determinadas condições, permitindo um controle de acesso mais refinado e seguro dos recursos na nuvem.

Para obter detalhes sobre as políticas do sistema DMS for Kafka, consulte [Gerenciamento de permissões](#).

NOTA

A autorização baseada em políticas é útil se você deseja permitir ou negar o acesso a uma API.

Uma conta tem todas as permissões necessárias para chamar todas as APIs, mas as permissões necessárias devem ser atribuídas aos usuários do IAM. As permissões necessárias para chamar uma API são determinadas pelas ações suportadas pela API. Somente os usuários que receberam permissões que permitem as ações podem chamar a API com sucesso. Por exemplo, se um usuário do IAM quiser consultar instâncias do Kafka usando uma API, o usuário deverá ter recebido permissões que permitam a ação **dms:instance:create**.

Ações suportadas

O DMS for Kafka fornece políticas definidas pelo sistema que podem ser usadas diretamente no IAM. Você também pode criar políticas personalizadas e usá-las para complementar políticas definidas pelo sistema, implementando um controle de acesso mais refinado. As operações suportadas pelas políticas são específicas das APIs. Seguem-se conceitos comuns relacionados com as políticas:

- Permissão: uma declaração em uma política que permite ou nega certas operações.
- APIs: as APIs REST que podem ser chamadas por um usuário que recebeu permissões específicas.

- Ação: operações específicas que são permitidas ou negadas.
- Projetos do IAM ou projetos empresariais: uma política personalizada pode ser aplicada a projetos do IAM ou projetos empresariais ou a ambos. As políticas que contêm ações para projetos do IAM e empresariais podem ser usadas e entrar em vigor para o IAM e o Enterprise Management. As políticas que contêm apenas ações para projetos do IAM podem ser usadas e só entram em vigor para o IAM. Os administradores podem verificar se uma ação é compatível com projetos do IAM ou projetos empresariais na lista de ações. Para obter detalhes sobre as diferenças entre o IAM e os projetos empresariais, consulte [Quais são as diferenças entre o IAM e o Enterprise Management?](#)

O DMS for Kafka oferece suporte às seguintes ações em políticas personalizadas:

- **Ações de gerenciamento do ciclo de vida**, incluindo ações suportadas pelas APIs de gerenciamento do ciclo de vida da instância do Kafka, como as APIs para criar uma instância, consultar a lista de instâncias, modificar informações de instâncias e reiniciar ou excluir instâncias em lote.
- **Ações de gerenciamento de instâncias**, incluindo ações suportadas pelas APIs de gerenciamento de instâncias do Kafka, como as APIs para redefinir senhas e ativar ou desativar a criação automática de tópicos.
- **Ações de gerenciamento de despejo**, incluindo ações suportadas por APIs de gerenciamento de despejo, como as APIs para criar um conector de despejo, criar uma tarefa de despejo e consultar a lista de tarefas de despejo.
- **Ação de gerenciamento de modificação de especificação**, suportada pela API de gerenciamento de modificação de especificação para modificar especificações de instância.
- **Ações de gerenciamento de tópicos**, incluindo ações suportadas por APIs de gerenciamento de tópicos, como as APIs para criar, consultar e modificar tópicos.
- **Ações de gerenciamento de usuários**, incluindo ações suportadas por APIs de gerenciamento de usuários, como as APIs para criar usuários, consultar usuários e configurar permissões de usuário.
- **Ações de consulta de mensagens**, incluindo ações suportadas por APIs de consulta de mensagens, como a API para consultar mensagens.
- **Ações de gerenciamento de tarefas em segundo plano**, incluindo ações suportadas por APIs de gerenciamento de tarefas em segundo plano, como as APIs para consultar a lista de tarefas em segundo plano de uma instância e consultar uma tarefa em segundo plano especificada.
- **Ações de gerenciamento de tags**, incluindo ações suportadas por APIs de gerenciamento de tags, como as APIs para consultar tags de instância e tags de projeto.

Gerenciamento do ciclo da vida

Tabela 6-1 Gerenciamento do ciclo da vida

Permissão	API	Ação	IAM (Projeto)	Empresarial (Projeto empresarial)
Criação de uma instância	POST /v2/{engine}/{project_id}/instances	dms:instance:create	√	√
Consulta de todas as instâncias	GET /v2/{project_id}/instances	dms:instance:list	√	√
Consulta de uma instância	GET /v2/{project_id}/instances/{instance_id}	dms:instance:get	√	√
Exclusão de uma instância	DELETE /v2/{project_id}/instances/{instance_id}	dms:instance:delete	√	√
Modificação das informações da instância	PUT /v2/{project_id}/instances/{instance_id}	dms:instance:modify	√	√
Reinicialização ou exclusão em lote de instâncias	POST /v2/{project_id}/instances/action	Reinicialização: dms:instance:modifyStatus Exclusão: dms:instance:delete	√	√

Gerenciamento de instâncias

Tabela 6-2 Gerenciamento de instâncias

Permissão	API	Ação	IAM (Projeto)	Empresarial (Projeto empresarial)
Redefinição de uma senha	POST /v2/{project_id}/instances/{instance_id}/password	dms:instance:resetAuthInfo	√	√

Permissão	API	Ação	IAM (Projeto)	Empresarial (Projeto empresarial)
Redefinição da senha do Kafka Manager	PUT /v2/{project_id}/instances/{instance_id}/kafka-manager-password	dms:instance:resetAuthInfo	√	√
Reinicialização do Kafka Manager	PUT /v2/{project_id}/instances/{instance_id}/restart-kafka-manager	dms:instance:modifyStatus	√	√
Configuração da criação automática dos tópicos	POST /v2/{project_id}/instances/{instance_id}/autotopic	dms:instance:modify	√	√
Modificação do endereço IP privado para acesso entre VPCs	POST /v2/{project_id}/instances/{instance_id}/crossvpc/modify	dms:instance:modify	√	√
Redefinição do deslocamento do grupo de consumidores para a posição especificada	POST /v2/{project_id}/instances/{instance_id}/management/groups/{group}/reset-message-offset	dms:instance:modify	√	√

Gerenciamento de despejos

Tabela 6-3 Gerenciamento de despejos

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Criação de um conector de despejo para uma instância	POST /v2/{project_id}/instances/{instance_id}/connector	dms:instance:connector	√	√

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Criação de uma tarefa de despejo	POST /v2/{project_id}/connectors/{connector_id}/sink-tasks	dms:instance:createConnectorSinkTask	√	√
Listagem de tarefas de despejo	GET /v2/{project_id}/connectors/{connector_id}/sink-tasks	dms:instance:listConnectorSinkTask	√	√
Modificação das cotas de tarefas de despejo	PUT /v2/{project_id}/connectors/{connector_id}/sink-tasks	dms:instance:createConnectorSinkTask	√	√
Consulta de uma tarefa de despejo	GET /v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}	dms:instance:getConnectorSinkTask	√	√
Exclusão de uma tarefa de despejo	POST /v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}	dms:instance:deleteConnectorSinkTask	√	√

Gerenciamento de modificação de especificação

Tabela 6-4 Gerenciamento de modificação de especificação

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Modificação das especificações da instância	POST /v2/{engine}/{project_id}/instances/{instance_id}/extend	dms:instance:scale	√	√

Gerenciamento de tópicos

Tabela 6-5 Gerenciamento de tópicos

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Criação de um tópico em uma instância do Kafka	POST /v2/{project_id}/instances/{instance_id}/topics	dms:instance:modify	√	√
Consulta de um tópico em uma instância de Kafka	GET /v2/{project_id}/instances/{instance_id}/topics	dms:instance:get	√	√
Modificação dos tópicos de uma instância de Kafka	PUT /v2/{project_id}/instances/{instance_id}/topics	dms:instance:modify	√	√
Exclusão dos tópicos em uma instância de Kafka em lote	POST /v2/{project_id}/instances/{instance_id}/topics/delete	dms:instance:modify	√	√

Gerenciamento de usuário

Tabela 6-6 Gerenciamento de usuário

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Consulta da lista de usuários	GET /v2/{project_id}/instances/{instance_id}/users	dms:instance:get	√	√
Criação de um usuário	POST /v2/{project_id}/instances/{instance_id}/users	dms:instance:modify	√	√
Exclusão de usuários em lote	PUT /v2/{project_id}/instances/{instance_id}/users	dms:instance:modify	√	√
Redefinição de uma senha do usuário	PUT /v2/{project_id}/instances/{instance_id}/users/{user_name}	dms:instance:get	√	√

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Consulta das permissões de usuário	GET /v1/{project_id}/instances/{instance_id}/topics/{topic_name}/accesspolicy	dms:instance:get	√	√
Concessão de permissões de usuário	POST /v1/{project_id}/instances/{instance_id}/topics/accesspolicy	dms:instance:modify	√	√

Consulta de mensagem

Tabela 6-7 Consulta de mensagem

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Consulta das mensagens	GET /v2/{project_id}/instances/{instance_id}/messages	dms:instance:get	√	√

Gerenciamento de tarefas em segundo plano

Tabela 6-8 Gerenciamento de tarefas em segundo plano

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Listagem das tarefas em segundo plano	GET /v2/{project_id}/instances/{instance_id}/tasks	dms:instance:getBackgroundTask	√	√
Consulta das tarefas em segundo plano	GET /v2/{project_id}/instances/{instance_id}/tasks/{task_id}	dms:instance:getBackgroundTask	√	√
Exclusão de uma tarefa em segundo plano	DELETE /v2/{project_id}/instances/{instance_id}/tasks/{task_id}	dms:instance:deleteBackgroundTask	√	√

Gerenciamento de tags

Tabela 6-9 Gerenciamento de tags

Permissão	API	Ação	Projetos de IAM	Projetos empresariais
Adição ou exclusão das tags em lote	POST /v2/{project_id}/kafka/{instance_id}/tags/action	dms:instance:modify	√	√
Listagem das tags de uma instância	GET /v2/{project_id}/kafka/{instance_id}/tags	dms:instance:get	√	√
Listagem das tags de um projeto	GET /v2/{project_id}/kafka/tags	dms:instance:get	√	√

7 APIs desatualizadas

7.1 API V1

7.1.1 APIs para gerenciamento de instâncias

7.1.1.1 Criação de uma instância

 **NOTA**

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Criação de uma instância](#).

Função

Essa API é usada para criar uma instância de **pay-per-use**.

URI

POST /v1.0/{project_id}/instances

[Tabela 7-1](#) descreve os parâmetros.

Tabela 7-1 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica o ID de um projeto.

Solicitação

Parâmetros de solicitação

[Tabela 7-2](#) descreve os parâmetros.

Tabela 7-2 Parâmetros de solicitação

Parâmetro	Tipo	Obrigatório	Descrição
name	String	Sim	Indica o nome da instância. Um nome de instância começa com uma letra, consiste de 4 a 64 caracteres e pode conter apenas letras, dígitos, sublinhados (_) e hifens (-).
description	String	Não	Indica a descrição de uma instância. É uma cadeia de caracteres que não contém mais de 1024 caracteres. NOTA A barra invertida (\) e aspas (") são caracteres especiais para pacotes JSON. Ao usar esses caracteres em um valor de parâmetro, adicione o caractere de escape (\) antes desses caracteres, por exemplo, \\ e \".
engine	String	Sim	Indica o mecanismo de mensagens. Defina o valor como kafka .
engine_version	String	Sim	Indica a versão do mecanismo de mensagens. Valor: 1.1.0 , 2.3.0 ou 2.7 .
specification	String	Sim	Indica a largura de banda da linha de base de uma instância do Kafka, ou seja, a quantidade máxima de dados transferidos por unidade de tempo. Unidade: MB Opções: <ul style="list-style-type: none"> ● 100 MB ● 300 MB ● 600 MB ● 1200 MB
storage_space	Integer	Sim	Indica o espaço de armazenamento da mensagem. Unidade: GB. Intervalo de valores: <ul style="list-style-type: none"> ● Instância do Kafka com specification de 100MB: 600–90.000 GB ● Instância do Kafka com specification de 300MB: 1200–90.000 GB ● Instância do Kafka com specification de 600MB: 2400–90.000 GB ● Instância do Kafka com specification de 1200MB: 4800–90.000 GB

Parâmetro	Tipo	Obrigatório	Descrição
partition_num	Integer	Sim	Indica o número máximo de partições em uma instância do Kafka. Opções: <ul style="list-style-type: none"> ● Quando specification for 100MB: 300 ● Quando specification for 300MB: 900 ● Quando specification for 600MB: 1800 ● Quando specification for 1200 MB: 1800
access_user	String	Não	Este parâmetro é obrigatório quando ssl_enable é definido como true . Este parâmetro é inválido quando ssl_enable é definido como false . Indica um nome de usuário. Um nome de usuário consiste de 4 a 64 caracteres e pode conter letras, dígitos e hifens (-).
password	String	Não	Este parâmetro é obrigatório quando ssl_enable é definido como true . Este parâmetro é inválido quando ssl_enable é definido como false . Indica uma senha de instância. A senha deve atender aos seguintes requisitos de complexidade: <ul style="list-style-type: none"> ● Deve ser uma cadeia de caracteres com 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais `~!@#\$\$%^&*()-_+=+ [{}];:','<.>/?
vpc_id	String	Sim	Indica o ID da VPC. Obtenha o valor utilizando um dos seguintes métodos: <ul style="list-style-type: none"> ● Método 1: faça login no console da VPC e visualize o ID da VPC nos detalhes da VPC. ● Método 2: consulte o ID da VPC por meio da API da VPC. Para obter detalhes, consulte Consulta das VPCs.

Parâmetro	Tipo	Obrigatório	Descrição
security_group_id	String	Sim	<p>Indica o grupo de segurança ao qual a instância pertence.</p> <p>Obtenha o valor utilizando um dos seguintes métodos:</p> <ul style="list-style-type: none"> ● Método 1: faça logon no console da VPC. Escolha Access Control > Security Groups no painel de navegação à esquerda. Na página exibida, clique no grupo de segurança de destino. Você pode visualizar o ID do grupo de segurança na página exibida. ● Método 2: chame a API para consultar grupos de segurança. Para obter detalhes, consulte Consulta dos grupos de segurança.
subnet_id	String	Sim	<p>Indica o ID da sub-rede.</p> <p>Obtenha o valor utilizando um dos seguintes métodos:</p> <ul style="list-style-type: none"> ● Método 1: faça logon no console da VPC e clique na sub-rede de destino na página de guia Subnets. Você pode visualizar o ID da rede na página exibida. ● Método 2: chame a API para consultar sub-redes. Para obter detalhes, consulte Consulta das sub-redes.
available_zones	Array	Sim	<p>Indica o ID da AZ onde os corretores residem e que tem recursos disponíveis. Esse parâmetro não pode ser uma matriz nula ou uma matriz em branco. Para obter detalhes sobre como obter o valor, consulte Consulta de informações da AZ. Verifique se a AZ tem recursos disponíveis.</p> <p>Ao criar uma instância do Kafka, você pode selecionar 1 AZ ou pelo menos 3 AZs. Ao especificar as AZs para corretores, use vírgulas (,) para separar várias AZs. Exemplo de configurações de parâmetros:</p> <ul style="list-style-type: none"> ● Uma AZ: "available_zones": ["a0865121f83b41cbafce65930a22a6e8"] ● Três ou mais AZs: "available_zones": ["a0865121f83b41cbafce65930a22a6e8","a0865121f83b41cbafce65930a22a6e7","a0865121f83b41cbafce65930a22a6e6"]
product_id	String	Sim	<p>Indica o ID do produto.</p> <p>Para obter detalhes sobre como obter o ID, consulte Consulta das especificações do produto.</p>

Parâmetro	Tipo	Obrigatório	Descrição
kafka_manager_user	String	Sim	Indica o nome de usuário para fazer login no Kafka Manager. O nome de usuário consiste de 4 a 64 caracteres e pode conter letras, dígitos, hifens (-) e sublinhados (_).
kafka_manager_password	String	Sim	Indica a senha para efetuar login no Kafka Manager. A senha deve atender aos seguintes requisitos de complexidade: <ul style="list-style-type: none"> ● Deve ser uma cadeia de caracteres com 8 a 32 caracteres. ● Deve conter pelo menos três dos seguintes tipos de caracteres: <ul style="list-style-type: none"> – Letras minúsculas – Letras maiúsculas – Dígitos – Caracteres especiais `~!@#%&*()-_+=+ [{}];:’,<.>/?
maintain_begin	String	Não	Indica a hora em que uma janela de tempo de manutenção é iniciada. Formato: HH:mm:ss <ul style="list-style-type: none"> ● A hora de início e a hora de término da janela de tempo de manutenção devem indicar o segmento de tempo de uma janela de tempo de manutenção suportada. Para obter detalhes sobre como consultar os segmentos de tempo das janelas de tempo de manutenção suportadas, consulte Consultando janelas de tempo de manutenção. ● A hora de início deve ser definida para 22:00:00, 02:00:00, 06:00:00, 10:00:00, 14:00:00 ou 18:00:00. ● Os parâmetros maintain_begin e maintain_end devem ser definidos em pares. Se o parâmetro maintain_begin for deixado em branco, o parâmetro maintain_end também será deixado em branco. Neste caso, o sistema define automaticamente a hora de início para 02:00:00.

Parâmetro	Tipo	Obrigatório	Descrição
maintain_end	String	Não	<p>Indica a hora em que uma janela de tempo de manutenção termina.</p> <p>Formato: HH:mm:ss</p> <ul style="list-style-type: none"> ● A hora de início e a hora de término da janela de tempo de manutenção devem indicar o segmento de tempo de uma janela de tempo de manutenção suportada. Para obter detalhes sobre como consultar os segmentos de tempo das janelas de tempo de manutenção suportadas, consulte Consultando janelas de tempo de manutenção. ● A hora de término é quatro horas mais tarde do que a hora de início. Por exemplo, se a hora de início for 22:00:00, a hora de término será 02:00:00. ● Os parâmetros maintain_begin e maintain_end devem ser definidos em pares. Se o parâmetro maintain_end for deixado em branco, o parâmetro maintain_start também estará em branco. Nesse caso, o sistema define automaticamente a hora de término para 06:00:00.
enable_public_ip	Boolean	Não	<p>Indica se deve ativar o acesso público para uma instância.</p> <ul style="list-style-type: none"> ● true: ativar ● false: desativar
public_bandwidth	String	Não	<p>Indica a largura de banda da rede pública. Unidade: Mbit/s</p> <p>Intervalo de valores:</p> <ul style="list-style-type: none"> ● Quando specification é 100MB, o valor deve ser um múltiplo do número de corretores e estar no intervalo de 3 a 900. ● Quando specification é 300MB, o valor deve ser um múltiplo do número de corretores e estar no intervalo de 3 a 900. ● Quando specification é 600MB, o valor deve ser um múltiplo do número de corretores e estar no intervalo de 4 a 1200. ● Quando specification é 1200MB, o valor deve ser um múltiplo do número de corretores e estar no intervalo de 8 a 2400.

Parâmetro	Tipo	Obrigatório	Descrição
publicip_id	String	Não	Indica o ID do endereço IP elástico (EIP) vinculado a uma instância. Use vírgulas (,) para separar vários IDs de EIP. Esse parâmetro é obrigatório se o acesso público estiver ativado (ou seja, enable_publicip estiver definido como true).
ssl_enable	Boolean	Não	Indica se deve ativar o acesso criptografado por SSL. <ul style="list-style-type: none"> ● true: ativar ● false: desativar
retention_policy	String	Não	Indica a ação a ser executada quando o uso da memória atingir o limite de capacidade do disco. Opções: <ul style="list-style-type: none"> ● time_base: excluir automaticamente as mensagens mais anteriores. ● produce_reject: parar de produzir novas mensagens
enable_auto_topic	Boolean	Não	Indica se a criação automática de tópicos deve ser ativada. <ul style="list-style-type: none"> ● true: ativar ● false: desativar <p>Se a criação automática de tópicos estiver ativada, um tópico será criado automaticamente com 3 partições e 3 réplicas quando uma mensagem for produzida ou consumida a partir de um tópico que não existe.</p>
storage_spec_code	String	Sim	Indica a especificação de I/O de armazenamento. Para obter detalhes sobre como selecionar um tipo de disco, consulte Tipos de disco e desempenho . Opções: <ul style="list-style-type: none"> ● dms.physical.storage.high ou dms.physical.storage.ultra quando o parâmetro specification for 100MB ● dms.physical.storage.high ou dms.physical.storage.ultra quando o parâmetro specification for 300MB ● dms.physical.storage.ultra quando o parâmetro specification for 600MB ● dms.physical.storage.ultra quando o parâmetro specification for 1200MB

Parâmetro	Tipo	Obrigatório	Descrição
enterprise_project_id	String	Não	Indica o ID do projeto empresarial.
tags	Array<Object>	Não	Indica a lista de tags.

Tabela 7-3 tags

Parâmetro	Tipo	Obrigatório	Descrição
key	String	Não	Indica a chave da tag. Uma chave de tag pode conter um máximo de 36 caracteres Unicode. A chave não pode ser deixada em branco ou ser uma cadeia vazia. Não pode conter caracteres ASCII (0–31) não imprimíveis e os seguintes caracteres especiais: =*<>\\,/
value	String	Não	Indica o valor. Um valor de tag pode conter no máximo 43 caracteres Unicode. O valor não pode ser deixado em branco ou ser uma cadeia de caracteres vazia. Não pode conter caracteres ASCII (0–31) não imprimíveis e os seguintes caracteres especiais: =*<>\\,/

```
{
  "name": "kafka-test",
  "engine": "kafka",
  "engine_version": "2.3.0",
  "specification": "100MB",
  "storage_space": 600,
  "partition_num": 300,
  "vpc_id": "b50claa7-39e0-420e-936b-ee5d35288f9c",
  "security_group_id": "d8c81e0f-de6a-4110-8c96-81af3eacb3d1",
  "subnet_id": "0b6cfaea-bce7-48eb-b38d-267c24df5f79",
  "available_zones": [
    "38b0f7a602344246bcb0da47b5d548e7"
  ],
  "product_id": "00300-30308-0--0",
  "kafka_manager_user": "test",
  "kafka_manager_password": "Zxxxx",
  "enable_publicip": true,
  "publicip_id": "87864b85-7097-4c06-9d62-718d7359a503,72c12ba7-fade-4b06-a680-01d335cf786d,11b535df-ed6d-4521-8d00-12bb60beb617",
  "storage_spec_code": "dms.physical.storage.high"
}
```

Resposta

Parâmetros de resposta

[Tabela 7-4](#) descreve os parâmetros.

Tabela 7-4 Parâmetros de resposta

Parâmetro	Tipo	Descrição
instance_id	String	Indica o ID da instância.

Exemplo de respostas

```
{  
  "instance_id": "8959ab1c-7n1a-yyb1-a05t-93dfc361b32d"  
}
```

Código de status

[Tabela 7-5](#) descreve o código de status das operações bem-sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-5 Código de status

Código de status	Descrição
200	A instância foi criada com sucesso.

7.1.1.2 Consulta de uma instância

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Consulta a uma instância](#).

Função

Essa API é usada para consultar os detalhes sobre uma instância.

URI

GET /v1.0/{project_id}/instances/{instance_id}

[Tabela 7-6](#) descreve os parâmetros.

Tabela 7-6 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica o ID de um projeto.
instance_id	String	Sim	Indica o ID da instância.

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

```
GET https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
```

Resposta

Parâmetros de resposta

[Tabela 7-7](#) descreve os parâmetros.

Tabela 7-7 Parâmetros de resposta

Parâmetro	Tipo	Descrição
name	String	Indica o nome da instância.
engine	String	Indica o mecanismo de mensagens.
engine_version	String	Indica a versão do mecanismo de mensagens.
specification	String	Indica a especificação da instância.
storage_space	Integer	Indica o espaço de armazenamento da mensagem. Unidade: GB
partition_num	String	Indica o número total de partições em uma instância de Kafka.
used_storage_space	Integer	Indica o espaço de armazenamento de mensagens usado. Unidade: GB
connect_address	String	Indica o endereço IP de uma instância.
port	Integer	Indica o número da porta de uma instância.
status	String	Indica o status de uma instância. Para mais detalhes, consulte Status da instância .
instance_id	String	Indica o ID da instância.

Parâmetro	Tipo	Descrição
resource_spec_code	String	Indica o identificador das especificações do recurso. <ul style="list-style-type: none"> ● dms.instance.kafka.cluster.c3.mini: instância do Kafka, largura de banda de referência de 100 MB/s ● dms.instance.kafka.cluster.c3.small.2: instância do Kafka, largura de banda de referência de 300 MB/s ● dms.instance.kafka.cluster.c3.middle.2: instância do Kafka, largura de banda de referência de 600 MB/s ● dms.instance.kafka.cluster.c3.high.2: instância do Kafka, largura de banda de referência de 1200 MB/s
type	String	Indica o tipo de instância. Valor: cluster
charging_mode	Integer	Indica o modo de cobrança. 1 : modo de pagamento por uso; 0 : cobrança anual/mensal.
vpc_id	String	Indica o ID de uma VPC.
vpc_name	String	Indica o nome de uma VPC.
created_at	String	Indica a hora em que uma instância é criada. A hora está no formato de carimbo de data/hora, ou seja, o deslocamento em milissegundos de 01/01/1970 00:00:00 UTC até a hora especificada.
product_id	String	Indica o ID do produto.
security_group_id	String	Indica o ID do grupo de segurança.
security_group_name	String	Indica o nome do grupo de segurança.
subnet_id	String	Indica o ID da sub-rede.
subnet_name	String	Indica o nome da sub-rede.
subnet_cidr	String	Indica o bloco CIDR da sub-rede.
available_zones	Array	Indica o ID da AZ à qual o nó da instância pertence. O ID da AZ é retornado.
user_id	String	Indica o ID do usuário.
user_name	String	Indica o nome de usuário.
access_user	String	Indica o nome de usuário de uma instância.
order_id	String	Indica o ID do pedido.
maintain_begin	String	Indica a hora em que uma janela de tempo de manutenção é iniciada. Formato: HH:mm:ss

Parâmetro	Tipo	Descrição
maintain_end	String	Indica a hora em que uma janela de tempo de manutenção termina. Formato: HH:mm:ss
enable_publicip	Boolean	Indica se deve ativar o acesso público para uma instância. <ul style="list-style-type: none"> ● true: ativar ● false: desativar
management_connect_address	String	Indica o endereço de conexão do Kafka Manager de uma instância do Kafka.
ssl_enable	Boolean	Indica se a autenticação de segurança deve ser ativada. <ul style="list-style-type: none"> ● true: ativar ● false: desativar
enterprise_project_id	String	Indica o ID do projeto empresarial.
is_logical_volume	Boolean	Distingue instâncias anteriores de novas instâncias durante a expansão da capacidade da instância. <ul style="list-style-type: none"> ● true: nova instância, que permite a expansão dinâmica da capacidade do disco sem reiniciar a instância. ● false: instância anterior.
extend_times	Integer	Indica o número de vezes de expansão do disco. Se exceder 20, o disco não pode ser expandido.
enable_autotopic	Boolean	Indica se a criação automática de tópicos está ativada. <ul style="list-style-type: none"> ● true: ativado ● false: desativado
total_storage_space	Integer	Indica o espaço de armazenamento da mensagem. Unidade: GB
storage_resource_id	String	Indica o ID do recurso de armazenamento.
storage_spec_code	String	Indica a especificação de I/O.
service_type	String	Indica o tipo de serviço.
storage_type	String	Indica o tipo de armazenamento.
retention_policy	String	Indica a política de retenção de mensagens.
kafka_public_status	String	Indica se o acesso público do Kafka está ativado.
public_bandwidth	Integer	Indica a largura de banda da rede pública.

Parâmetro	Tipo	Descrição
public_connect_address	String	Indica o endereço IP da instância para acesso público. Esse parâmetro é exibido somente quando o acesso público está ativado.
kafka_manager_user	String	Indica o nome de usuário para fazer logon no Kafka Manager.
enable_log_collection	Boolean	Indica se a coleta de log está ativada.
cross_vpc_info	String	Indica informações de acesso entre VPCs.
ipv6_enable	Boolean	Indica se o IPv6 está ativado.
ipv6_connect_addresses	Array of strings	Indica o endereço de conexão IPv6.
rest_enable	Boolean	Indica se a função REST do Kafka está ativada.
rest_connect_address	String	Indica o endereço de conexão REST do Kafka.
message_query_instance_enable	Boolean	Indica se a consulta de mensagens está ativada.
vpc_client_plain	Boolean	Indica se o acesso de texto não criptografado dentro da VPC está ativado.
support_features	String	Indica a lista de recursos suportados pela instância de Kafka.
trace_enable	Boolean	Indica se o rastreamento de mensagens está ativado.
pod_connect_address	String	Indica o endereço de conexão no lado do locatário.
disk_encrypted	Boolean	Indica se a criptografia de disco está ativada. <ul style="list-style-type: none"> ● true: ativado ● false: desativado
kafka_private_connect_address	String	Indica o endereço de conexão privada de uma instância do Kafka.
ces_version	String	Indica a versão do Cloud Eye.
tags	Array<Object>	Indica a lista de tags.

Tabela 7-8 tags

Parâmetro	Tipo	Descrição
key	String	Indica a chave da tag.

Parâmetro	Tipo	Descrição
value	String	Indica o valor da tag.

Exemplo de respostas

```
{
  "name": "kafka-100230526",
  "engine": "kafka",
  "port": 9092,
  "status": "RUNNING",
  "type": "cluster",
  "specification": "100MB",
  "engine_version": "XXX",
  "connect_address": "192.168.1.116,192.168.1.152,192.168.1.78",
  "connect_dn": "",
  "instance_id": "ef84dd5f-3ece-4336-8c99-987defd62e3a",
  "resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
  "charging_mode": 1,
  "vpc_id": "2477879f-aebf-496f-a08a-67812885ce9b",
  "vpc_name": "vpc-y00502467",
  "created_at": "1568797295209",
  "product_id": "00300-30308-0--0",
  "security_group_id": "008a08e2-10cc-4d9b-90ab-3f3b8f6c3333",
  "security_group_name": "z00417080-cce-node-na7j",
  "subnet_id": "5ca08fb7-7522-4d95-9fa5-ff6b3592a29d",
  "subnet_name": "subnet-cyd-6102",
  "subnet_cidr": "192.168.1.0/24",
  "available_zones": [
    "ae04cf9d61544df3806a3feeb401b204"
  ],
  "user_id": "2b4af4428ec840dfa1f0fla32e965567",
  "user_name": "laiyh",
  "kafka_manager_user": "root",
  "maintain_begin": "22:00:00",
  "maintain_end": "02:00:00",
  "storage_space": 492,
  "total_storage_space": 600,
  "used_storage_space": 25,
  "partition_num": "300",
  "ssl_enable": false,
  "management_connect_address": "https://192.168.1.116:9999",
  "storage_resource_id": "81982562-ce8b-490a-95fa-2b225c292271",
  "storage_spec_code": "dms.physical.storage.ultra",
  "service_type": "advanced",
  "storage_type": "hec",
  "enterprise_project_id": "0",
  "is_logical_volume": true,
  "extend_times": 0,
  "retention_policy": "produce_reject",
  "ipv6_enable": false,
  "ipv6_connect_addresses": [],
  "connector_enable": false,
  "connector_id": "",
  "rest_enable": false,
  "rest_connect_address": "",
  "message_query_inst_enable": true,
  "vpc_client_plain": false,
  "support_features":
"feature.physerver.kafka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.physerver.kafka.publlic.dynamic,feature.physerver.kafka.user.manager",
  "trace_enable": false,
  "agent_enable": false,
  "pod_connect_address":
"100.113.16.105:9100,100.113.5.197:9100,100.113.15.231:9100",
}
```

```
"disk_encrypted": false,  
"enable_auto_topic": true  
}
```

Código de status

Tabela 7-9 descreve o código de status das operações bem-sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-9 Código de status

Código de status	Descrição
200	A instância especificada foi consultada com sucesso.

7.1.1.3 Modificação de uma instância

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Modificação das informações de uma instância](#).

Função

Essa API é usada para modificar as informações da instância, incluindo o nome da instância, a descrição, a janela de manutenção e o grupo de segurança.

URI

PUT /v1.0/{project_id}/instances/{instance_id}

Tabela 7-10 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica o ID de um projeto.
instance_id	String	Sim	Indica o ID da instância.

Solicitação

Parâmetros de solicitação

Tabela 7-11 descreve os parâmetros.

Tabela 7-11 Parâmetros de solicitação

Parâmetro	Tipo	Obrigatório	Descrição
name	String	Não	Indica o nome da instância. Um nome de instância consiste em 4 a 64 caracteres, incluindo letras, dígitos e hífens (-) e deve começar com uma letra.
description	String	Não	Indica a descrição de uma instância. É uma cadeia de caracteres que não contém mais de 1024 caracteres. NOTA A barra invertida (\) e aspas (") são caracteres especiais para pacotes JSON. Ao usar esses caracteres em um valor de parâmetro, adicione o caractere de escape (\) antes desses caracteres, por exemplo, \\ e \".
maintain_begin	String	Não	Indica a hora em que uma janela de tempo de manutenção é iniciada. Formato: HH:mm:ss <ul style="list-style-type: none"> ● A hora de início e a hora de término da janela de tempo de manutenção devem indicar o segmento de tempo de uma janela de tempo de manutenção suportada. Para obter detalhes sobre como consultar os segmentos de tempo das janelas de tempo de manutenção suportadas, consulte Consultando janelas de tempo de manutenção. ● A hora de início deve ser definida para 22:00:00, 02:00:00, 06:00:00, 10:00:00, 14:00:00 ou 18:00:00. ● Os parâmetros maintain_begin e maintain_end devem ser definidos em pares. Se o parâmetro maintain_begin for deixado em branco, o parâmetro maintain_end também será deixado em branco. Neste caso, o sistema define automaticamente a hora de início para 02:00:00.

Parâmetro	Tipo	Obrigatório	Descrição
maintain_end	String	Não	<p>Indica a hora em que uma janela de tempo de manutenção termina.</p> <p>Formato: HH:mm:ss</p> <ul style="list-style-type: none"> ● A hora de início e a hora de término da janela de tempo de manutenção devem indicar o segmento de tempo de uma janela de tempo de manutenção suportada. Para obter detalhes sobre como consultar os segmentos de tempo das janelas de tempo de manutenção suportadas, consulte Consultando janelas de tempo de manutenção. ● A hora de término é quatro horas mais tarde do que a hora de início. Por exemplo, se a hora de início for 22:00:00, a hora de término será 02:00:00. ● Os parâmetros maintain_begin e maintain_end devem ser definidos em pares. Se o parâmetro maintain_end for deixado em branco, o parâmetro maintain_start também será deixado em branco. Nesse caso, o sistema define automaticamente a hora de término para 06:00:00.
security_group_id	String	Não	Indica o ID do grupo de segurança.
retention_policy	String	Não	<p>Indica a política de limite de capacidade.</p> <p>Opções:</p> <ul style="list-style-type: none"> ● produce_reject: novas mensagens não podem ser criadas. ● time_base: as mensagens mais antigas são excluídas.
enterprise_project_id	String	Não	Indica o ID do projeto empresarial.

Exemplos de solicitações

Exemplo 1:

```
PUT https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
{
  "name": "dms002",
  "description": "instance description"
}
```

Exemplo 2:

```
PUT https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
{
  "name": "dms002",
  "description": "instance description",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00"
}
```

Resposta

Parâmetros de resposta

Nenhum.

Exemplo de respostas

Nenhum.

Código de status

Tabela 7-12 descreve o código de status das operações bem-sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-12 Código de status

Código de status	Descrição
204	A instância foi modificada com sucesso.

7.1.1.4 Exclusão de uma instância

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Exclusão de uma instância](#).

Função

Essa API é usada para excluir uma instância para liberar todos os recursos ocupados por ela.

URI

DELETE /v1.0/{project_id}/instances/{instance_id}

Tabela 7-13 descreve os parâmetros.

Tabela 7-13 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica a ID de um projeto.

Parâmetro	Tipo	Obrigatório	Descrição
instance_id	String	Sim	Indica o ID da instância.

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

```
DELETE https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
```

Resposta

Parâmetros de resposta

Nenhum.

Exemplo de respostas

Nenhum.

Código de status

Tabela 7-14 descreve o código de status das operações sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-14 Código de status

Código de status	Descrição
204	A instância foi excluída com sucesso.

7.1.1.5 Reinicialização ou exclusão de instâncias em lote

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Reinicialização ou exclusão das instâncias em lote](#).

Função

Essa API é usada para reiniciar ou excluir instâncias em lote.

Quando uma instância estiver sendo reiniciada, as solicitações de recuperação e criação de mensagens do cliente serão rejeitadas.

A exclusão de uma instância excluirá os dados na instância sem nenhum backup. Tenha cuidado ao realizar esta operação.

URI

POST /v1.0/{project_id}/instances/action

Tabela 7-15 descreve os parâmetros.

Tabela 7-15 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica a ID de um projeto.

Solicitação

Parâmetros de solicitação

Tabela 7-16 descreve os parâmetros.

Tabela 7-16 Parâmetros de solicitação

Parâmetro	Tipo	Obrigatório	Descrição
action	String	Sim	Indica a operação a ser executada nas instâncias. O valor deste parâmetro pode ser restart ou delete .
instances	Array	Não	Indica a lista dos ID de instância.
allFailure	String	Não	Quando definido como kafka , indica que todas as instâncias de Kafka que não forem criadas serão excluídas.

Exemplos de solicitações

Reiniciação das instâncias em lote

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/action
{
  "action" : "restart",
  "instances" : ["54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-
dbad-4d79-9610-7163e6f8b640"]
}
```

Exclusão das instâncias em lote

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/action
{
  "action" : "delete",
  "instances" : ["54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-
dbad-4d79-9610-7163e6f8b640"]
}
```

Exclusão de todas as instâncias de Kafka que não foram criadas

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/action
{
```

```
"action" : "delete",  
"allFailure" : "kafka"  
}
```

Resposta

Parâmetros de resposta

Quando **action** é definida como **delete**, **allFailure** é definida como **kafka** e uma resposta vazia é retornada, as instâncias são excluídas com sucesso. [Tabela 7-17](#) descreve os parâmetros.

Tabela 7-17 Parâmetros de resposta

Parâmetro	Tipo	Descrição
results	Array	Indica o resultado da modificação da instância.

Tabela 7-18 descrição do parâmetro de resultados

Parâmetro	Tipo	Descrição
instance	String	Indica o ID da instância.
result	String	Indica um resultado da operação, que pode ser success ou failed

Exemplo de respostas

```
{  
  "results": [  
    {  
      "result": "success",  
      "instance": "afc90a2a-a02c-4cba-94d5-58dfa9ad1e0d"  
    },  
    {  
      "result": "success",  
      "instance": "67fc5f8d-3986-4f02-bb75-4075a23112de"  
    }  
  ]  
}
```

Código de status

[Tabela 7-19](#) descreve o código de status das operações sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-19 Código de status

Código de status	Descrição
200	As instâncias são reiniciadas ou excluídas com êxito.
204	Exclusão sucedida de uma instância que falhou ao ser criada

7.1.1.6 Consulta de todas as instâncias

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Listagem de todas as instâncias](#).

Função

Essa API é usada para consultar as instâncias de um locatário por condições definidas.

URI

GET /v1.0/{project_id}/instances?
engine={engine}&name={name}&status={status}&id={id}&includeFailure={includeFailure}
&exactMatchName={exactMatchName}&enterprise_project_id={enterprise_project_id}

[Tabela 7-20](#) descreve os parâmetros.

Tabela 7-20 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica o ID de um projeto.
engine	String	Não	Indica o mecanismo de mensagens. Valor: kafka
name	String	Não	Indica o nome da instância.
id	String	Não	Indica o ID da instância.
status	String	Não	Indica o status da instância. Para mais detalhes, consulte Status da instância .
includeFailure	String	Não	Indica se deve ser retornado o número de instâncias que não foram criadas. Se o valor for true , será retornado o número de instâncias que não foram criadas. Se o valor não for true , o número não será retornado.

Parâmetro	Tipo	Obrigatório	Descrição
exactMatchName	String	Não	Indica se deve pesquisar a instância que corresponde precisamente a um nome de instância especificado. O valor padrão é false , indicando que uma pesquisa difusa é executada com base em um nome de instância especificado. Se o valor for true , a instância que corresponde precisamente a um nome de instância especificado será consultada.
enterprise_project_id	String	Não	Indica o ID do projeto empresarial.

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

```
GET https://{dms_endpoint}/v1.0/{project_id}/instances?
start=1&limit=10&name=&status=&id=&includeFailure=true&exactMatchName=false
```

Resposta

Parâmetros de resposta

[Tabela 7-21](#) descreve os parâmetros.

Tabela 7-21 Parâmetros de resposta

Parâmetro	Tipo	Descrição
instances	Array	Indica detalhes da instância.
instance_num	Integer	Indica o número de instâncias.

Tabela 7-22 Descrição do parâmetro de instância

Parâmetro	Tipo	Descrição
name	String	Indica o nome da instância.
engine	String	Indica o mecanismo de mensagens.
engine_version	String	Indica a versão do mecanismo.

Parâmetro	Tipo	Descrição
specification	String	Indica as especificações de uma instância.
storage_space	Integer	Indica o espaço de armazenamento da mensagem. Unidade: GB
partition_num	String	Indica o número máximo de tópicos em uma instância do Kafka.
used_storage_space	Integer	Indica o espaço de armazenamento de mensagens usado. Unidade: GB
connect_address	String	Indica o endereço IP de uma instância.
port	Integer	Indica o número da porta de uma instância.
status	String	Indica o status de uma instância. Para mais detalhes, consulte Status da instância .
instance_id	String	Indica o ID da instância.
resource_spec_code	String	Indica o identificador das especificações do recurso. <ul style="list-style-type: none">● dms.instance.kafka.cluster.c3.mini: instância do Kafka, largura de banda de referência de 100 MB/s● dms.instance.kafka.cluster.c3.small.2: instância do Kafka, largura de banda de referência de 300 MB/s● dms.instance.kafka.cluster.c3.middle.2: instância do Kafka, largura de banda de referência de 600 MB● dms.instance.kafka.cluster.c3.high.2: instância do Kafka, largura de banda de referência de 1200 MB
charging_mode	Integer	Indica o modo de cobrança. Opções: 1 : pagamento por uso; 0 : anual/mensal
vpc_id	String	Indica o ID de uma VPC.
vpc_name	String	Indica o nome de uma VPC.
created_at	String	Indica a hora em que uma instância é criada. A hora está no formato de carimbo de data/hora, ou seja, o deslocamento em milissegundos de 01/01/1970 00:00:00 UTC até a hora especificada.
user_id	String	Indica o ID do usuário.
user_name	String	Indica o nome de usuário.
access_user	String	Indica o nome de usuário de uma instância.
order_id	String	Indica o ID do pedido.

Parâmetro	Tipo	Descrição
maintain_begin	String	Indica a hora em que uma janela de tempo de manutenção é iniciada. Formato: HH:mm:ss
maintain_end	String	Hora em que a janela de tempo de manutenção termina. Formato: HH:mm:ss
enable_publicip	Boolean	Indica se deve ativar o acesso público para uma instância. <ul style="list-style-type: none"> ● true: ativar ● false: desativar
management_connect_address	String	Indica o endereço de conexão do Kafka Manager de uma instância do Kafka.
ssl_enable	Boolean	Indica se a autenticação de segurança deve ser ativada. <ul style="list-style-type: none"> ● true: ativar ● false: desativar
enterprise_project_id	String	Indica o ID do projeto empresarial.
is_logical_volume	Boolean	Distingue instâncias anteriores de novas instâncias durante a expansão da capacidade da instância. <ul style="list-style-type: none"> ● true: nova instância, que permite a expansão dinâmica da capacidade do disco sem reiniciar a instância. ● false: instância anterior.
extend_times	Integer	Indica o número de vezes de expansão do disco. Se exceder 20, o disco não pode ser expandido.
enable_auto_topic	Boolean	Indica se a criação automática de tópicos está ativada. <ul style="list-style-type: none"> ● true: ativado ● false: desativado
type	String	Indica o tipo de instância. Valor: cluster .
product_id	String	Indica o ID do produto.
security_group_id	String	Indica o ID do grupo de segurança.
security_group_name	String	Indica o nome do grupo de segurança.
subnet_id	String	Indica o ID da sub-rede.

Parâmetro	Tipo	Descrição
available_zones	Array	Indica a AZ à qual o nó da instância pertence. O ID da AZ é retornado.
total_storage_space	Integer	Indica o espaço de armazenamento da mensagem. Unidade: GB
public_connect_address	String	Indica o endereço IP da instância para acesso público. Esse parâmetro é exibido somente quando o acesso público está ativado.
storage_resource_id	String	Indica o ID do recurso de armazenamento.
storage_spec_code	String	Indica a especificação de I/O.
service_type	String	Indica o tipo de serviço.
storage_type	String	Indica o tipo de armazenamento.
retention_policy	String	Indica a política de retenção de mensagens.
kafka_public_status	String	Indica se o acesso público do Kafka está ativado.
public_bandwidth	Integer	Indica a largura de banda da rede pública.
kafka_manager_user	String	Indica o nome de usuário para fazer logon no Kafka Manager.
enable_log_collection	Boolean	Indica se a coleta de log está ativada.
cross_vpc_info	String	Indica informações de acesso entre VPCs.
ipv6_enable	Boolean	Indica se o IPv6 está ativado.
ipv6_connect_addresses	Array of strings	Indica o endereço de conexão IPv6.
rest_enable	Boolean	Indica se a função REST do Kafka está ativada.
rest_connect_address	String	Indica o endereço REST do Kafka.
message_query_instance_enable	Boolean	Indica se a consulta de mensagens está ativada.
vpc_client_plain	Boolean	Indica se o acesso de texto não criptografado dentro da VPC está ativado.
support_features	String	Indica a lista de recursos suportados pela instância de Kafka.
trace_enable	Boolean	Indica se o rastreamento de mensagens está ativado.

Parâmetro	Tipo	Descrição
pod_connect_address	String	Indica o endereço de conexão no lado do locatário.
disk_encrypted	Boolean	Indica se a criptografia de disco está ativada.
kafka_private_connect_address	String	Indica o endereço de conexão privada de uma instância do Kafka.
ces_version	String	Indica a versão do Cloud Eye.
tags	Array<Object>	Indica a lista de tags.

Tabela 7-23 tags

Parâmetro	Tipo	Descrição
key	String	Indica a chave da tag.
value	String	Indica o valor da tag.

Exemplo de respostas

```
{
  "instances": [
    {
      "name": "kafka-100230526",
      "engine": "kafka",
      "port": 9092,
      "status": "RUNNING",
      "type": "cluster",
      "specification": "100MB",
      "engine_version": "XXX",
      "connect_address": "192.168.1.116,192.168.1.152,192.168.1.78",
      "instance_id": "ef84dd5f-3ece-4336-8c99-987defd62e3a",
      "resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
      "charging_mode": 1,
      "vpc_id": "2477879f-aebf-496f-a08a-67812885ce9b",
      "vpc_name": "vpc-y00502467",
      "created_at": "1568797295209",
      "product_id": "00300-30308-0--0",
      "security_group_id": "008a08e2-10cc-4d9b-90ab-3f3b8f6c3333",
      "security_group_name": "z00417080-cce-node-na7j",
      "subnet_id": "5ca08fb7-7522-4d95-9fa5-ff6b3592a29d",
      "available_zones": [
        "ae04cf9d61544df3806a3feeb401b204"
      ],
      "user_id": "2b4af4428ec840dfa1f0f1a32e965567",
      "user_name": "laiyh",
      "kafka_manager_user": "root",
      "maintain_begin": "22:00",
      "maintain_end": "02:00",
      "storage_space": 492,
      "total_storage_space": 600,
      "used_storage_space": 25,
      "partition_num": "300",
      "ssl_enable": false,
      "management_connect_address": "https://192.168.1.116:9999",
    }
  ]
}
```

```

        "storage_resource_id": "81982562-ce8b-490a-95fa-2b225c292271",
        "storage_spec_code": "dms.physical.storage.ultra",
        "service_type": "advanced",
        "storage_type": "hec",
        "enterprise_project_id": "0",
        "is_logical_volume": true,
        "extend_times": 0,
        "retention_policy": "produce_reject",
        "ipv6_enable": false,
        "ipv6_connect_addresses": [],
        "rest_enable": false,
        "rest_connect_address": "",
        "message_query_inst_enable": true,
        "vpc_client_plain": false,
        "support_features":
"feature.physerver.kafka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.physerver.kafka.pulbic.dynamic,feature.physerver.kafka.user.manager",
        "trace_enable": false,
        "agent_enable": false,
        "pod_connect_address":
"100.113.16.105:9100,100.113.5.197:9100,100.113.15.231:9100",
        "disk_encrypted": false,
        "enable_auto_topic": true
    }
],
    "instance_num": 1
}

```

Código de status

Tabela 7-24 descreve o código de status das operações bem-sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-24 Código de status

Código de status	Descrição
200	Todas as instâncias são consultadas com sucesso.

7.1.1.7 Criação de um tópico em uma instância do Kafka

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Criação de um tópico para uma instância de Kafka](#).

Função

Essa API é usada para criar um tópico em uma instância do Kafka.

URI

POST /v1.0/{project_id}/instances/{instance_id}/topics

Tabela 7-25 descreve os parâmetros.

Tabela 7-25 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica o ID de um projeto.
instance_id	String	Sim	Indica o ID da instância.

Solicitação

Parâmetros de solicitação

Tabela 7-26 descreve os parâmetros.

Tabela 7-26 Parâmetros de solicitação

Parâmetro	Tipo	Obrigatório	Descrição
id	String	Sim	Indica o nome de um tópico. O nome de um tópico tem de 4 a 64 caracteres, começa com uma letra e contém apenas letras, hifens (-), sublinhados (_) e dígitos.
partition	Integer	Não	Indica o número de partições de tópico, que é usado para definir o número de mensagens consumidas simultaneamente. Intervalo de valores: 1–100. Valor padrão: 3 .
replication	Integer	Não	Indica o número de réplicas, que é configurado para garantir a confiabilidade dos dados. Intervalo de valores: 1–3. Valor padrão: 3 .
sync_replication	Boolean	Não	Indica se a replicação síncrona deve ser ativada. Depois que essa função for ativada, o parâmetro acks no cliente do produtor deve ser definido como -1 . Caso contrário, esse parâmetro não terá efeito. Por padrão, a replicação síncrona está desativada.
retention_time	Integer	Não	Indica o período de retenção de uma mensagem. Seu valor padrão é 72 . Intervalo de valores: 1–720. Unidade: hora.
sync_message_flush	Boolean	Não	Indica se a liberação síncrona deve ser ativada. Valor padrão: false . A liberação síncrona compromete o desempenho.

Exemplos de solicitações

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}/topics
{
  "id" : "haha",
  "partition" : 3,
  "replication" : 3,
  "sync_replication" : true,
  "retention_time" : 10,
  "sync_message_flush" : true
}
```

Resposta

Parâmetros de resposta

[Tabela 7-27](#) descreve os parâmetros.

Tabela 7-27 Parâmetros de resposta

Parâmetro	Tipo	Descrição
id	String	Indica o nome de um tópico.

Exemplo de respostas

```
{
  "id": "haha"
}
```

Código de status

[Tabela 7-28](#) descreve o código de status das operações bem-sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-28 Código de status

Código de status	Descrição
200	O tópico foi criado com sucesso.

7.1.1.8 Consulta de um tópico em uma instância de Kafka

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Lista dos tópicos de uma instância de Kafka](#).

Função

Essa API é usada para consultar detalhes sobre um tópico em uma instância de Kafka.

URI

GET /v1.0/{project_id}/instances/{instance_id}/topics

Tabela 7-29 descreve os parâmetros.

Tabela 7-29 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica a ID de um projeto.
instance_id	String	Sim	Indica o ID da instância.

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

```
GET https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}/topics
```

Resposta

Parâmetros de resposta

Tabela 7-30 descreve os parâmetros de resposta.

Tabela 7-30 Parâmetro de resposta

Parâmetro	Tipo	Descrição
total	Integer	Indica o número total de tópicos.
size	Integer	Indica o número máximo de registros a serem exibidos em uma página.
remain_partitions	Integer	Indica o número de partições restantes.
max_partitions	Integer	Indica o número total de partições.
topics	Array	Indica a lista de tópicos.

Tabela 7-31 Descrição do parâmetro

Parâmetro	Tipo	Descrição
policiesOnly	Boolean	Se esta política é a política padrão.
id	String	Indica o nome do tópico.
replication	Integer	Indica o número de réplicas, que é configurado para garantir a confiabilidade dos dados.

Parâmetro	Tipo	Descrição
partition	Integer	Indica o número de partições de tópico, que é usado para definir o número de mensagens consumidas simultaneamente.
retention_time	Integer	Indica o período de retenção de uma mensagem.
sync_replication	Boolean	Indica se a replicação síncrona deve ser ativada. Depois que essa função for ativada, o parâmetro acks no cliente do produtor deve ser definido como -1 . Caso contrário, esse parâmetro não terá efeito. Por predefinição, a replicação síncrona está desativada.
sync_message_flush	Boolean	Indica se o esvaziamento síncrono deve ser ativado. O rubor síncrono compromete o desempenho.
external_configs	Object	Indica a configuração estendida.
topic_type	Integer	Indica o tipo de tópico.

Exemplo de respostas

```
{
  "count": 1,
  "topics": [
    {
      "id": "topic-test",
      "replication": 3,
      "partition": 4,
      "retention_time": 72,
      "sync_replication": "false",
      "sync_message_flush": "false"
    }
  ]
}
```

Código de status

Tabela 7-32 descreve o código de status das operações sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-32 Código de status

Código de status	Descrição
200	As informações foram consultadas com sucesso.

7.1.1.9 Exclusão dos tópicos em uma instância de Kafka em lote

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Exclusão dos tópicos em uma instância de Kafka em lote](#).

Função

Essa API é usada para excluir tópicos em uma instância de Kafka em lotes.

URI

POST /v1.0/{project_id}/instances/{instance_id}/topics/delete

[Tabela 7-33](#) descreve os parâmetros.

Tabela 7-33 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
project_id	String	Sim	Indica a ID de um projeto.
instance_id	String	Sim	Indica o ID da instância.

Solicitação

Parâmetros de solicitação

[Tabela 7-34](#) descreve os parâmetros.

Tabela 7-34 Parâmetro de solicitação

Parâmetro	Tipo	Obrigatório	Descrição
topics	Array	Sim	Indica a lista de tópicos a serem excluídos.

Exemplos de solicitações

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}/topics/delete
{
  "topics" : ["hah", "aabb"]
}
```

Resposta

Parâmetros de resposta

[Tabela 7-35](#) descreve os parâmetros.

Tabela 7-35 Parâmetros de resposta

Parâmetro	Tipo	Descrição
topics	Array	Indica a lista de tópicos.

Tabela 7-36 tópicos descrição do parâmetro

Parâmetro	Tipo	Descrição
id	String	Indica o nome do tópico.
success	Boolean	Indica se os tópicos são excluídos.

Exemplo de respostas

```
{
  "topics" : [{
    "id" : "haha",
    "success" : true
  }, {
    "id" : "aabb",
    "success" : true
  }
]
```

Código de status

Tabela 7-37 descreve o código de status das operações sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-37 Código de status

Código de status	Descrição
200	Os tópicos são excluídos com sucesso.

7.1.2 Outras APIs

7.1.2.1 Consulta de informações da AZ

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Listagem das Informações da AZ](#).

Função

Essa API é usada para consultar o ID da AZ.

URI

GET /v1.0/availableZones

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

```
GET https://{dms_endpoint}/v1.0/availableZones
```

Resposta

Parâmetros de resposta

[Tabela 7-38](#) e [Tabela 7-39](#) descrevem os parâmetros.

Tabela 7-38 Parâmetros de resposta

Parâmetro	Tipo	Descrição
region_id	String	Indica o ID da região.
available_zones	Array	Indica detalhes das AZ. Para mais detalhes, consulte Tabela 7-39 .

Tabela 7-39 available_zones parameter description

Parâmetro	Tipo	Descrição
soldOut	Boolean	Indica se os recursos estão esgotados.
id	String	Indica o ID de uma AZ.
code	String	Indica o código de uma AZ.
name	String	Indica o nome de uma AZ.
port	String	Indica o número da porta de uma AZ.
resource_availability	String	Indica se uma AZ tem recursos disponíveis. <ul style="list-style-type: none">● true: A AZ tem recursos disponíveis.● false: Os recursos da AZ foram esgotados.

Exemplo de respostas

```
{
  regionId: "XXXX",
  available_zones:[
    {
      "id":"1d7b939b382c4c3bb3481a8ca10da768",
      "name":"az10.dc1",
```

```
    "code": "az10.dc1",  
    "port": "8002",  
    "resource_availability": "true"  
  },  
  {  
    "id": "1d7b939b382c4c3bb3481a8ca10da769",  
    "name": "az10.dc2",  
    "code": "az10.dc2",  
    "port": "8002",  
    "resource_availability": "true"  
  }  
]  
}
```

Código de status

Tabela 7-40 descreve o código de status das operações sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-40 Código de status

Código de status	Descrição
200	As informações da AZ são consultadas com êxito.

7.1.2.2 Consulta das especificações do produto

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Consulta de especificações do produto](#).

Função

Essa API é usada para consultar as especificações do produto para configurar o ID do produto.

Por exemplo, para criar uma instância de Kafka de pagamento por uso com largura de banda de 100 MB/s, localize a seção em que o valor de **bandwidth** é de **100MB** em "Hourly" na mensagem de resposta. Em seguida, o valor de **product_id** na mesma seção é o ID do produto que deve ser configurado para a instância de Kafka.

unavailable_zones indica as AZs indisponíveis. Se o valor estiver vazio, todas as AZs estão disponíveis. Caso contrário, as AZs listadas no valor não têm recursos suficientes. Certifique-se de que as AZs nas quais você deseja criar a instância não estejam listadas aqui.

URI

GET /v1.0/products?engine={engine}

Tabela 7-41 descreve os parâmetros.

Tabela 7-41 Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
engine	String	Sim	Indica o mecanismo de mensagens.

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

```
GET https://{dms_endpoint}/v1.0/products?engine={engine}
```

Resposta

Parâmetros de resposta

A [Tabela 7-43](#) descreve os parâmetros de resposta.

Tabela 7-42 Parâmetros

Parâmetro	Tipo	Descrição
Hourly	Array	Indica a lista de produtos de pagamento por uso.
Monthly	Array	Indica a lista de produtos em modo de cobrança anual/mensal. Atualmente, não é possível criar instâncias anuais/mensais do Kafka chamando uma API.

Tabela 7-43 Parâmetros

Parâmetro	Tipo	Descrição
name	String	Indica o mecanismo de mensagens, que é kafka .
version	String	Indica a versão do mecanismo de mensagens. Atualmente, apenas as versões 1.1.0, 2.3.0 e 2.7 são suportadas.
values	Array	Indica as especificações do produto. Para mais detalhes, consulte Tabela 7-44 .

Tabela 7-44 Descrição do parâmetro values

Parâmetro	Tipo	Descrição
detail	Array	Indica os detalhes das especificações. Para mais detalhes, consulte Tabela 7-45 .
name	String	Indica o tipo de instância.
unavailable_zones	Array	Indica as AZs onde os recursos são esgotados.
available_zones	Array	Indica as AZs onde há recursos disponíveis.

Tabela 7-45 Descrição do parâmetro detail

Parâmetro	Tipo	Descrição
tps	String	Indica o número máximo de mensagens por unidade de tempo.
storage	String	Indica o espaço de armazenamento da mensagem.
partition_num	String	Indica o número máximo de tópicos em uma instância do Kafka.
product_id	String	Indica o ID do produto.
spec_code	String	Indica o ID da especificação.
io	Array	Indica as informações de I/O. Para mais detalhes, consulte Tabela 7-46 .
bandwidth	String	Indica a largura de banda de uma instância do Kafka.
available_zones	Array	Indica as AZs onde há recursos disponíveis.
ecs_flavor_id	String	Indica os sabores do ECS correspondente.
arch_type	String	Indica o tipo de arquitetura de instância. No momento, apenas x86 é compatível.

Tabela 7-46 Descrição do parâmetro io

Parâmetro	Tipo	Descrição
io_type	String	Indica o tipo de I/O.
storage_spec_code	String	Indica a especificação de I/O.
available_zones	Array	Indica as AZs onde há recursos de I/O disponíveis.
unavailable_zones	Array of strings	Indica as AZs em que os recursos de I/O estão esgotados.
volume_type	String	Indica o tipo de disco.

Exemplo de respostas

```
{
  "Hourly": [{
    "name": "kafka",
    "version": "XXX",
    "values": [{
      "detail": [{
        "tps": "50000",
        "storage": "600",
        "partition_num": "300",
        "product_id": "00300-30308-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.mini",
        "io": [{
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high",
          "volume_type": "SAS"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra",
          "volume_type": "SSD"
        }
      ]},
      "bandwidth": "100MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.large.2"
    },
    {
      "tps": "100000",
      "storage": "1200",
      "partition_num": "900",
      "product_id": "00300-30310-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.small.2",
      "io": [{
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high",
        "volume_type": "SAS"
      },
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra",
        "volume_type": "SSD"
      }
    ]},
      "bandwidth": "300MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.xlarge.2"
    },
    {
      "tps": "200000",
      "storage": "2400",
      "partition_num": "1800",
      "product_id": "00300-30312-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.middle.2",
      "io": [{
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra",
        "volume_type": "SSD"
      }
    ]},
      "bandwidth": "600MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.2xlarge.2"
    },
    {
      "tps": "300000",
      "storage": "4800",
      "partition_num": "1800",
      "product_id": "00300-30314-0--0",
```

```

        "spec_code": "dms.instance.kafka.cluster.c3.high.2",
        "io": [{
            "io_type": "ultra",
            "storage_spec_code": "dms.physical.storage.ultra",
            "volume_type": "SSD"
        }],
        "bandwidth": "1200MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3ne.2xlarge.2"
    }],
    "name": "cluster",
    "unavailable_zones": []
}]]
}],
"Monthly": [{
    "name": "kafka",
    "version": "XXX",
    "values": [{
        "detail": [{
            "tps": "50000",
            "storage": "600",
            "partition_num": "300",
            "product_id": "00300-30309-0--0",
            "spec_code": "dms.instance.kafka.cluster.c3.mini",
            "io": [{
                "io_type": "high",
                "storage_spec_code": "dms.physical.storage.high",
                "volume_type": "SAS"
            }],
            {
                "io_type": "ultra",
                "storage_spec_code": "dms.physical.storage.ultra",
                "volume_type": "SSD"
            }
        ]],
        "bandwidth": "100MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3.large.2"
    }],
    {
        "tps": "100000",
        "storage": "1200",
        "partition_num": "900",
        "product_id": "00300-30311-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.small.2",
        "io": [{
            "io_type": "high",
            "storage_spec_code": "dms.physical.storage.high",
            "volume_type": "SAS"
        }],
        {
            "io_type": "ultra",
            "storage_spec_code": "dms.physical.storage.ultra",
            "volume_type": "SSD"
        }
    ]],
        "bandwidth": "300MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3.xlarge.2"
    }],
    {
        "tps": "200000",
        "storage": "2400",
        "partition_num": "1800",
        "product_id": "00300-30313-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.middle.2",
        "io": [{
            "io_type": "ultra",
            "storage_spec_code": "dms.physical.storage.ultra",
            "volume_type": "SSD"
        }],
    }],
}]]

```

```

        "bandwidth": "600MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3.2xlarge.2"
    },
    {
        "tps": "300000",
        "storage": "4800",
        "partition_num": "1800",
        "product_id": "00300-30315-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.high.2",
        "io": [{
            "io_type": "ultra",
            "storage_spec_code": "dms.physical.storage.ultra",
            "volume_type": "SSD"
        }],
        "bandwidth": "1200MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3ne.2xlarge.2"
    },
    "name": "cluster",
    "unavailable_zones": []
}
}
}

```

Código de status

[Tabela 7-47](#) descreve o código de status das operações bem-sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-47 Código de status

Código de status	Descrição
200	As especificações do produto foram consultadas com sucesso.

7.1.2.3 Consultando janelas de tempo de manutenção

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Listagem da janela de tempo de manutenção](#).

Função

Essa API é usada para consultar o horário de início e término da janela de manutenção.

URI

GET /v1.0/instances/maintain-windows

Solicitação

Parâmetros de solicitação

Nenhum.

Exemplos de solicitações

GET https://{dms_endpoint}/v1.0/instances/maintain-windows

Resposta

Parâmetros de resposta

[Tabela 7-48](#) e [Tabela 7-49](#) descrevem os parâmetros de resposta.

Tabela 7-48 Parâmetros de resposta

Parâmetro	Tipo	Descrição
maintain_windows	Array	Indica uma lista de janelas de tempo de manutenção suportadas.

Tabela 7-49 descrição do parâmetro maintain_windows

Parâmetro	Tipo	Descrição
seq	Integer	Indica o número sequencial de uma janela de tempo de manutenção.
begin	String	Indica a hora em que uma janela de tempo de manutenção é iniciada.
end	String	Indica a hora em que uma janela de tempo de manutenção termina.
default	Boolean	Indica se uma janela de tempo de manutenção está definida para o segmento de tempo padrão.

Exemplo de respostas

```
{
  "maintain_windows": [{
    "default": false,
    "seq": 1,
    "begin": "22:00:00",
    "end": "02:00:00"
  },
  {
    "default": true,
    "seq": 2,
    "begin": "02:00:00",
    "end": "06:00:00"
  },
  {
    "default": false,
    "seq": 3,
    "begin": "06:00:00",
    "end": "10:00:00"
  },
  {
    "default": false,
    "seq": 4,
    "begin": "10:00:00",
```



```
    "end": "14:00:00"
  },
  {
    "default": false,
    "seq": 5,
    "begin": "14:00:00",
    "end": "18:00:00"
  },
  {
    "default": false,
    "seq": 6,
    "begin": "18:00:00",
    "end": "22:00:00"
  }
}]
}
```

Código de status

Tabela 7-50 descreve o código de status das operações sucedidas. Para obter detalhes sobre outros códigos de status, consulte [Código de status](#).

Tabela 7-50 Código de status

Código de status	Descrição
200	As janelas de tempo de manutenção são consultadas com sucesso.

7.2 API V2

7.2.1 Adição de partições a um tópico para uma instância do Kafka

Função

Essa API é usada para adicionar partições a um tópico de uma instância do Kafka.

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Modificação de tópicos de uma instância do Kafka](#).

URI

POST /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions-reassignment

Tabela 7-51 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes sobre como obter um ID de projeto, consulte Obtenção de um ID de projeto.
instance_id	Sim	String	ID da instância.
topic	Sim	String	Nome do tópico.

Solicitação

Tabela 7-52 Parâmetro de solicitação

Parâmetro	Obrigatório	Tipo	Descrição
partition	Não	Integer	Número total de partições após a adição. O valor deve ser maior do que o número atual de partições. Valor máximo: 100 .

Resposta

Nenhuma.

Exemplo de solicitação

Adição de partições a um tópico.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions-reassignment
{
  "partition" : 3
}
```

Exemplo de resposta

Nenhum.

Código de status

Código de status	Descrição
204	Partições adicionadas com sucesso.

Código de erro

Para obter detalhes, consulte [Códigos de erro](#).

7.2.2 Redefinição do deslocamento do grupo de consumidores para a posição especificada

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Redefinição do deslocamento do grupo de consumidores para a posição especificada](#).

Função

As instâncias do Kafka não oferecem suporte à redefinição on-line do deslocamento do consumidor. Antes de redefinir, pare o cliente para o qual o deslocamento deve ser reiniciado. Depois que um cliente é interrompido, o servidor considera o cliente off-line somente após o período especificado em `ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG` (1000 ms por padrão).

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

POST `/v2/{project_id}/instances/{instance_id}/management/groups/{group}/reset-message-offset`

Tabela 7-53 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto .
instance_id	Sim	String	ID da instância.
group	Sim	String	Nome do grupo de consumidores.

Parâmetros de solicitação

Tabela 7-54 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
topic	Não	String	Nome do tópico.

Parâmetro	Obrigatório	Tipo	Descrição
partition	Sim	Integer	Número da partição. O valor padrão é -1 , indicando que todas as partições são redefinidas.
message_offset	Não	Long	<p>Esse parâmetro redefine o deslocamento do grupo de consumidores para a posição especificada.</p> <ul style="list-style-type: none"> ● Se esta posição for anterior ao deslocamento mais antigo atual, o deslocamento será redefinido para a posição mais antiga. ● Se essa posição for posterior ao último deslocamento atual, o deslocamento será redefinido para a posição mais recente. <p>Deve ser especificado message_offset ou timestamp.</p>
timestamp	Não	Long	<p>Este parâmetro redefine o deslocamento do grupo de consumidores para o tempo especificado. O valor é um carimbo de data/hora UNIX, em milissegundos.</p> <ul style="list-style-type: none"> ● Se esse carimbo de data/hora for anterior ao carimbo de data/hora mais antigo atual, o deslocamento será redefinido para o carimbo de data/hora mais antigo. ● Se esse carimbo de data/hora for posterior ao carimbo de data/hora mais recente atual, o deslocamento será redefinido para o carimbo de data/hora mais recente. <p>Deve ser especificado message_offset ou timestamp.</p>

Parâmetros de resposta

Nenhum

Exemplo de solicitações

- Um deslocamento de grupo de consumidores é redefinido para a posição especificada.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/  
groups/{group}/reset-message-offset
```

```
{  
  "topic" : "test",  
  "partition" : 0,  
  "message_offset" : 10  
}
```

- Um deslocamento de grupo de consumidores é redefinido para o tempo especificado.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/  
groups/{group}/reset-message-offset
```

```
{  
  "topic" : "test",  
  "partition" : 0,  
  "timestamp" : 1571812144000  
}
```

Exemplo de resposta

Nenhum

Exemplos de código do SDK

A seguir estão exemplos de código do SDK.

Java

- Um deslocamento de grupo de consumidores é redefinido para a posição especificada.

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ResetMessageOffsetSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in  
        plaintext, which has great security risks. It is recommended that the AK and  
        SK be stored in ciphertext in configuration files or environment variables  
        and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for  
        authentication. Before running this example, set environment variables  
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ResetMessageOffsetRequest request = new ResetMessageOffsetRequest();  
        ResetMessageOffsetReq body = new ResetMessageOffsetReq();  
        body.withMessageOffset(10L);  
        body.withPartition(0);  
    }  
}
```

```
        body.withTopic("test");
        request.withBody(body);
        try {
            ResetMessageOffsetResponse response =
client.resetMessageOffset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Um deslocamento de grupo de consumidores é redefinido para o tempo especificado.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and
SK be stored in ciphertext in configuration files or environment variables
and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetMessageOffsetRequest request = new ResetMessageOffsetRequest();
        ResetMessageOffsetReq body = new ResetMessageOffsetReq();
        body.withTimestamp(1571812144000L);
        body.withPartition(0);
        body.withTopic("test");
        request.withBody(body);
        try {
            ResetMessageOffsetResponse response =
client.resetMessageOffset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- Um deslocamento de grupo de consumidores é redefinido para a posição especificada.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetMessageOffsetRequest()
        request.body = ResetMessageOffsetReq(
            message_offset=10,
            partition=0,
            topic="test"
        )
        response = client.reset_message_offset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Um deslocamento de grupo de consumidores é redefinido para o tempo especificado.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
```

```
credentials = BasicCredentials(ak, sk) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ResetMessageOffsetRequest()
    request.body = ResetMessageOffsetReq(
        timestamp=1571812144000,
        partition=0,
        topic="test"
    )
    response = client.reset_message_offset(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Um deslocamento de grupo de consumidores é redefinido para a posição especificada.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetMessageOffsetRequest{
        messageOffsetResetMessageOffsetReq:= int64(10)
        topicResetMessageOffsetReq:= "test"
        request.Body = &model.ResetMessageOffsetReq{
            MessageOffset: &messageOffsetResetMessageOffsetReq,
            Partition: int32(0),
            Topic: &topicResetMessageOffsetReq,
        }
    }
    response, err := client.ResetMessageOffset(request)
    if err == nil {
```



```

        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- Um deslocamento de grupo de consumidores é redefinido para o tempo especificado.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and
    SK be stored in ciphertext in configuration files or environment variables
    and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetMessageOffsetRequest{
        timestampResetMessageOffsetReq:= int64(1571812144000)
        topicResetMessageOffsetReq:= "test"
        request.Body = &model.ResetMessageOffsetReq{
            Timestamp: &timestampResetMessageOffsetReq,
            Partition: int32(0),
            Topic: &topicResetMessageOffsetReq,
        }
    }
    response, err := client.ResetMessageOffset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

Códigos de status

Código de status	Descrição
204	Redefinição bem-sucedida do deslocamento do grupo de consumidores para a posição especificada.

Códigos de erro

Consulte [Códigos de erro](#).

7.2.3 Modificação das cotas de tarefas de despejo

Função

Essa API é usada para modificar cotas de tarefas de despejo.

Você pode chamar essa API para aumentar a cota de tarefa de despejo para instâncias criadas antes de setembro de 2022. Para instâncias criadas em setembro de 2022 ou posterior, a cota de tarefa de despejo já é o máximo por padrão. Um erro é relatado quando você chama essa API para modificar a cota de tarefa de despejo, porque a cota de tarefa de despejo não pode ser reduzida.

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

PUT /v2/{project_id}/connectors/{connector_id}/sink-tasks

Tabela 7-55 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
connector_id	Sim	String	ID de despejo da instância. Para obter detalhes, consulte Consulta de IDs de instância.

Parâmetros de solicitação

Tabela 7-56 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
sink_max_tasks	Sim	Integer	Número total de tarefas de despejo.

Parâmetros de resposta

Nenhum

Exemplo de solicitações

Modificação das cotas de tarefas de despejo.

```
PUT https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks
{
  "sink_max_tasks" : 9
}
```

Exemplo de resposta

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

Modificação das cotas de tarefas de despejo.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateSinkTaskQuotaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateSinkTaskQuotaRequest request = new UpdateSinkTaskQuotaRequest();
        UpdateSinkTaskQuotaReq body = new UpdateSinkTaskQuotaReq();
        body.withSinkMaxTasks(9);
        request.withBody(body);
        try {
            UpdateSinkTaskQuotaResponse response =
            client.updateSinkTaskQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

Modificação das cotas de tarefas de despejo.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSinkTaskQuotaRequest()
        request.body = UpdateSinkTaskQuotaReq(
            sink_max_tasks=9
        )
        response = client.update_sink_task_quota(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Modificação das cotas de tarefas de despejo.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateSinkTaskQuotaRequest{}
request.Body = &model.UpdateSinkTaskQuotaReq{
    SinkMaxTasks: int32(9),
}
response, err := client.UpdateSinkTaskQuota(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Códigos de status

Código de status	Descrição
204	As cotas de tarefas de despejo são modificadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

7.2.4 Criação de um pedido para desativar os conectores de despejo de uma instância

Função

Essa API é usada para criar um pedido para desativar os conectores de despejo para uma instância.

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Desativação do Smart Connect \(instância de pagamento por uso\)](#).

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/delete-connector-order

Tabela 7-57 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto. Comprimento mínimo: 1 Comprimento máximo: 100
instance_id	Sim	String	ID da instância. Comprimento mínimo: 1 Comprimento máximo: 100

Parâmetros de solicitação

Tabela 7-58 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
instance_id	Sim	String	ID da instância cujo conector você deseja desativar, que é o mesmo que o do caminho da solicitação.
url	Não	String	Página exibida após o envio do pedido de desativação do conector.

Parâmetros de resposta

Código de status: 200

Tabela 7-59 Parâmetro do corpo da resposta

Parâmetro	Tipo	Descrição
order_id	String	ID do pedido gerado pelo CBC.

Exemplo de solicitação

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/delete-connector-order
```

```
{
  "instance_id" : "20c6b355-5d95-45ef-b476-e38bccce0d7",
  "url" : "https://console.xxx.xxx.com/dms/?engine=all&region=xxx&locale=xxx#/queue/manager/newKafkaList"
}
```

Exemplo de resposta

Código de status: 200

Um pedido criado para desativar conectores de despejo para uma instância

```
{
  "order_id" : "CS2304180947HLABU"
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateDeleteConnectorOrderSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        // plaintext, which has great security risks. It is recommended that the AK and SK
        // be stored in ciphertext in configuration files or environment variables and
        // decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        // authentication. Before running this example, set environment variables
        // CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateDeleteConnectorOrderRequest request = new
        CreateDeleteConnectorOrderRequest();
        ConnectorOrderRequestBody body = new ConnectorOrderRequestBody();
        body.withUrl("https://console.xxx.xxx.com/dms/?
engine=all&region=xxx&locale=xxx#/queue/manager/newKafkaList");
        body.withInstanceId("20c6b355-5d95-45ef-b476-e38bccce0d7");
        request.withBody(body);
        try {
            CreateDeleteConnectorOrderResponse response =
            client.createDeleteConnectorOrder(request);
            System.out.println(response.toString());
        }
    }
}
```

```
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDeleteConnectorOrderRequest()
        request.body = ConnectorOrderRequestBody(
            url="https://console.xxx.xxx.com/dms/?
engine=all&region=xxx&locale=xxx#/queue/manager/newKafkaList",
            instance_id="20c6b355-5d95-45ef-b476-e38bccce0d7"
        )
        response = client.create_delete_connector_order(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
```



```
// The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateDeleteConnectorOrderRequest{}
urlConnectorOrderRequestBody:= "https://console.xxx.xxx.com/dms/?
engine=all&region=xxx&locale=xxx#/queue/manager/newKafkaList"
request.Body = &model.ConnectorOrderRequestBody{
    Url: &urlConnectorOrderRequestBody,
    InstanceId: "20c6b355-5d95-45ef-b476-e38bccccce0d7",
}
response, err := client.CreateDeleteConnectorOrder(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

Código de status

Código de status	Descrição
200	É criado um pedido para desativar os conectores de despejo de uma instância.

Códigos de erro

Consulte [Códigos de erro](#).

7.2.5 Criação de uma tarefa de despejo

Função

Essa API é usada para criar uma tarefa de despejo.

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Ativação do Smart Connect \(instância de pagamento por uso\)](#).

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

POST /v2/{project_id}/connectors/{connector_id}/sink-tasks

Tabela 7-60 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
connector_id	Sim	String	ID de despejo da instância. Para obter detalhes, consulte Consulta de IDs de instância.

Parâmetros de solicitação

Tabela 7-61 Parâmetros do corpo da solicitação

Parâmetro	Obrigatório	Tipo	Descrição
source_type	Sim	String	Tipo de dados de origem. Atualmente, apenas BLOB é suportado.
task_name	Sim	String	Nome de uma tarefa de despejo.
destination_type	Sim	String	Tipo de armazenamento. Atualmente, apenas OBS é suportado.
obs_destination_descriptor	Sim	ObsDestinationDescriptor object	Descrição do despejo.

Tabela 7-62 ObsDestinationDescriptor

Parâmetro	Obrigatório	Tipo	Descrição
topics	Sim	String	Nomes dos tópicos a serem despejados. Vários tópicos podem ser separados por vírgulas (.). Expressões regulares são suportadas.

Parâmetro	Obrigatório	Tipo	Descrição
topics_regex	Não	String	Expressão regular dos tópicos a serem descartados. Defina o parâmetro topics_regex ou o parâmetro topics . Não defina ambos. Não defina como ".*".
consumer_strategy	Sim	String	Deslocamento. <ul style="list-style-type: none"> ● latest: as mensagens são consumidas a partir do final do tópico. ● earliest: as mensagens são consumidas desde o início do tópico. O valor padrão é latest .
destination_file_type	Sim	String	Formato do arquivo de despejo. Atualmente, apenas arquivos TXT são suportados.
access_key	Sim	String	ID da chave de acesso (AK).
secret_key	Sim	String	Chave de acesso secreta (SK).
obs_bucket_name	Sim	String	Nome do bucket do OBS usado para armazenar os dados.
obs_path	Não	String	Caminho do OBS. Este parâmetro pode ser deixado em branco por padrão. O valor pode conter no máximo 64 caracteres, incluindo letras, dígitos, sublinhados (_), hifens (-) e barras (/). Este parâmetro é deixado vazio por padrão.

Parâmetro	Obrigatório	Tipo	Descrição
partition_format	Sim	String	<p>Estrutura de diretório do arquivo de objeto gravado no OBS. A estrutura de diretórios está no formato de yyyy/MM/dd/HH/mm (hora em que a tarefa de despejo foi criada).</p> <ul style="list-style-type: none"> ● yyyy: ano. ● yyyy/MM: ano e mês. ● yyyy/MM/dd: ano, mês e dia. ● yyyy/MM/dd/HH: ano, mês, dia e hora. ● yyyy/MM/dd/HH/mm: ano, mês, dia, hora e minuto. Por exemplo, 2017/11/10/14/49 significa que a estrutura de diretórios é 2017 > 11 > 10 > 14 > 49, onde 2017 é a pasta mais externa. <p>NOTA Depois que os dados são despejados com sucesso, a estrutura do diretório de armazenamento é obs_bucket_path/file_prefix/partition_format. O fuso horário padrão é GMT+08:00.</p>
record_delimiter	Não	String	<p>Delimitador para o arquivo de despejo, que é usado para separar os dados do usuário que são gravados no arquivo de despejo.</p> <p>Intervalo de valores:</p> <ul style="list-style-type: none"> ● Comma (,) ● Semicolon (;) ● Vertical bar () ● Newline (\n) ● NULL <p>Valor padrão: newline (\n).</p>
deliver_time_interval	Sim	Integer	<p>Nenhum arquivo de pacote será gerado se não houver dados dentro de um segmento de tempo.</p> <p>Intervalo de valores: 30 a 900</p> <p>Unidade: segundo.</p> <p>NOTA Esse parâmetro é obrigatório se os dados de streaming forem despejados no OBS.</p>


```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateSinkTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSinkTaskRequest request = new CreateSinkTaskRequest();
        CreateSinkTaskReq body = new CreateSinkTaskReq();
        ObsDestinationDescriptor obsDestinationDescriptorbody = new
ObsDestinationDescriptor();
        obsDestinationDescriptorbody.withTopics("topic-test")
            .withConsumerStrategy(ObsDestinationDescriptor.ConsumerStrategyEnum.fr
omValue("earliest"))
            .withDestinationFileType(ObsDestinationDescriptor.DestinationFileTypeE
num.fromValue("TEXT"))
            .withAccessKey("XXXXXXXXXXXXXXXXXXXXXXXX")
            .withSecretKey("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
            .withObsBucketName("6666")
            .withObsPath("obsTransfer-1122976956")
            .withPartitionFormat(ObsDestinationDescriptor.PartitionFormatEnum.from
Value("yyyy/MM/dd/HH/mm"))
            .withRecordDelimiter("")
            .withDeliverTimeInterval(300);
        body.withObsDestinationDescriptor(obsDestinationDescriptorbody);

body.withDestinationType(CreateSinkTaskReq.DestinationTypeEnum.fromValue("OBS"));
body.withTaskName("obsTransfer-1122976956");
body.withSourceType(CreateSinkTaskReq.SourceTypeEnum.fromValue("BLOB"));
request.withBody(body);
        try {
            CreateSinkTaskResponse response = client.createSinkTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Criação de uma tarefa de despejo para despejar dados em topic-test para OBS

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    # plaintext, which has great security risks. It is recommended that the AK and SK
    # be stored in ciphertext in configuration files or environment variables and
    # decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    # authentication. Before running this example, set environment variables
    # CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSinkTaskRequest()
        obsDestinationDescriptorbody = ObsDestinationDescriptor(
            topics="topic-test",
            consumer_strategy="earliest",
            destination_file_type="TEXT",
            access_key="XXXXXXXXXXXXXXXXXXXX",
            secret_key="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
            obs_bucket_name="6666",
            obs_path="obsTransfer-1122976956",
            partition_format="yyy/MM/dd/HH/mm",
            record_delimiter="",
            deliver_time_interval=300
        )
        request.body = CreateSinkTaskReq(
            obs_destination_descriptor=obsDestinationDescriptorbody,
            destination_type="OBS",
            task_name="obsTransfer-1122976956",
            source_type="BLOB"
        )
        response = client.create_sink_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Criação de uma tarefa de despejo para despejar dados em topic-test para OBS

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
```

```

    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
plaintext, which has great security risks. It is recommended that the AK and SK
be stored in ciphertext in configuration files or environment variables and
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
authentication. Before running this example, set environment variables
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSinkTaskRequest{
        obsPathObsDestinationDescriptor:= "obsTransfer-1122976956"
        recordDelimiterObsDestinationDescriptor:= ""
        obsDestinationDescriptorbody := &model.ObsDestinationDescriptor{
            Topics: "topic-test",
            ConsumerStrategy:
model.GetObsDestinationDescriptorConsumerStrategyEnum().EARLIEST,
            DestinationFileType:
model.GetObsDestinationDescriptorDestinationFileTypeEnum().TEXT,
            AccessKey: "XXXXXXXXXXXXXXXXXXXXXXXX",
            SecretKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
            ObsBucketName: "6666",
            ObsPath: &obsPathObsDestinationDescriptor,
            PartitionFormat:
model.GetObsDestinationDescriptorPartitionFormatEnum().YYYY_MM_DD_HH_MM,
            RecordDelimiter: &recordDelimiterObsDestinationDescriptor,
            DeliverTimeInterval: int32(300),
        }
        request.Body = &model.CreateSinkTaskReq{
            ObsDestinationDescriptor: obsDestinationDescriptorbody,
            DestinationType: model.GetCreateSinkTaskReqDestinationTypeEnum().OBS,
            TaskName: "obsTransfer-1122976956",
            SourceType: model.GetCreateSinkTaskReqSourceTypeEnum().BLOB,
        }
        response, err := client.CreateSinkTask(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

Código de status

Código de status	Descrição
200	Uma tarefa de despejo foi criada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

7.2.6 Listagem de tarefas de despejo

Função

Essa API é usada para listar tarefas de despejo.

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Consulta de tarefas do Smart Connect](#).

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

GET /v2/{project_id}/connectors/{connector_id}/sink-tasks

Tabela 7-64 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
connector_id	Sim	String	ID de despejo da instância. Para obter detalhes, consulte Consulta de IDs de instância.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 7-65 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
tasks	Array of tasks objects	Lista de tarefas de despejo.
total_number	Integer	Número total de tarefas de despejo.

Parâmetro	Tipo	Descrição
max_tasks	Integer	Número máximo de tarefas.
quota_tasks	Integer	Cotas de tarefas.

Tabela 7-66 tasks

Parâmetro	Tipo	Descrição
task_id	String	ID da tarefa.
task_name	String	Nome de uma tarefa de despejo.
destination_type	String	Tipo da tarefa de despejo.
create_time	Long	Hora em que a tarefa de despejo é criada.
status	String	Status da tarefa de despejo.
topics	String	Lista de tópicos ou expressão regular de tópicos da tarefa de despejo.

Exemplo de solicitação

Listagem de tarefas de despejo

```
GET https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks
```

Exemplo de resposta

Código de status: 200

Listagem bem-sucedida de tarefas de despejo

```
{
  "tasks" : [ {
    "task_id" : "2e148bed-3038-4617-8ade-b52e84a33eeb",
    "task_name" : "obsTransfer-1122976956",
    "destination_type" : "OBS",
    "create_time" : 1592309487621,
    "status" : "RUNNING",
    "topics" : "topic-test"
  } ],
  "total_number" : 1,
  "max_tasks" : 9,
  "quota_tasks" : 10
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListSinkTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSinkTasksRequest request = new ListSinkTasksRequest();
        try {
            ListSinkTasksResponse response = client.listSinkTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
```

```
CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSinkTasksRequest()
    response = client.list_sink_tasks(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSinkTasksRequest{}
    response, err := client.ListSinkTasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Código de status

Código de status	Descrição
200	As tarefas de despejo são listadas com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

7.2.7 Consulta de uma tarefa de despejo

Função

Essa API é usada para consultar uma tarefa de despejo.

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Consulta de tarefas do Smart Connect](#).

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

GET /v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}

Tabela 7-67 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
connector_id	Sim	String	ID de despejo da instância. Para obter detalhes, consulte Consulta de IDs de instância.
task_id	Sim	String	ID da tarefa de despejo.

Tabela 7-68 Parâmetros de consulta

Parâmetro	Obrigatório	Tipo	Descrição
topic-info	Não	String	Se a informação do tópico está contida. O valor padrão é false .

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Código de status: 200

Tabela 7-69 Parâmetros do corpo de resposta

Parâmetro	Tipo	Descrição
task_name	String	Nome de uma tarefa de despejo.
destination_type	String	Tipo da tarefa de despejo.
create_time	Long	Hora em que a tarefa de despejo é criada.
status	String	Status da tarefa de despejo.
topics	String	Lista de tópicos ou expressão regular de tópicos da tarefa de despejo.
obs_destination_descriptor	obs_destination_descriptor object	Descrição do despejo.
topics_info	Array of topics_info objects	Informação do tópico.

Tabela 7-70 obs_destination_descriptor

Parâmetro	Tipo	Descrição
consumer_strategy	String	Política de consumo de mensagens: <ul style="list-style-type: none"> ● latest: as mensagens são consumidas a partir do final do tópico. ● earliest: as mensagens são consumidas desde o início do tópico. O valor padrão é latest .
destination_file_type	String	Formato do arquivo de despejo. Somente text é suportado.
obs_bucket_name	String	Nome do bucket do OBS usado para armazenar os dados.
obs_path	String	Caminho do OBS.

Parâmetro	Tipo	Descrição
partition_format	String	<p>Estrutura de diretório do arquivo de objeto gravado no OBS. A estrutura de diretórios está no formato yyyy/MM/dd/HH/mm (hora em que a tarefa de despejo foi criada).</p> <ul style="list-style-type: none"> ● yyyy: ano. ● yyyy/MM: ano e mês. ● yyyy/MM/dd: ano, mês e dia. ● yyyy/MM/dd/HH: ano, mês, dia e hora. ● yyyy/MM/dd/HH/mm: ano, mês, dia, hora e minuto. Por exemplo, 2017/11/10/14/49 significa que a estrutura de diretórios é 2017 > 11 > 10 > 14 > 49, onde 2017 é a pasta mais externa. <p>NOTA Depois que os dados são despejados com sucesso, a estrutura do diretório de armazenamento é obs_bucket_path/file_prefix/partition_format. O fuso horário padrão é GMT+08:00.</p>
record_delimiter	String	<p>Delimitador para o arquivo de despejo, que é usado para separar os dados do usuário que são gravados no arquivo de despejo.</p> <p>Intervalo de valores:</p> <ul style="list-style-type: none"> ● Comma (,) ● Semicolon (;) ● Vertical bar () ● Newline (\n) ● NULL <p>Valor padrão: newline (\n).</p>
deliver_time_interval	Integer	<p>Nenhum arquivo de pacote será gerado se não houver dados dentro de um segmento de tempo.</p> <p>Intervalo de valores: 30-900. Valor padrão: 300. Unidade: segundo.</p> <p>NOTA Esse parâmetro é obrigatório se os dados de streaming forem despejados no OBS.</p>
obs_part_size	Long	<p>Tamanho (em bytes) de cada arquivo a ser carregado.</p> <p>Valor padrão: 5242880.</p>

Tabela 7-71 topics_info

Parâmetro	Tipo	Descrição
topic	String	Nome do tópico.

Parâmetro	Tipo	Descrição
partitions	Array of partitions objects	Lista de partições.

Tabela 7-72 partitions

Parâmetro	Tipo	Descrição
partition_id	String	ID da partição.
status	String	Status de execução.
last_transfer_of_fset	String	Deslocamento de mensagem despejada.
log_end_offset	String	Deslocamento da mensagem.
lag	String	Número de mensagens empilhadas.

Exemplo de solicitação

Consulta de detalhes da tarefa de despejo especificada

```
GET https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}?topic-info=true
```

Exemplo de resposta

Código de status: 200

Consulta bem-sucedida de uma tarefa de despejo

```
{
  "task_name" : "obsTransfer-56997523",
  "destination_type" : "OBS",
  "create_time" : 1628126621283,
  "status" : "RUNNING",
  "topics" : "topic-sdk-no-delete",
  "obs_destination_descriptor" : {
    "consumer_strategy" : "earliest",
    "destination_file_type" : "TEXT",
    "obs_bucket_name" : "testobs",
    "obs_path" : "obsTransfer-56997523",
    "partition_format" : "yyyy/MM/dd/HH/mm",
    "record_delimiter" : "",
    "deliver_time_interval" : 300,
    "obs_part_size" : 5242880
  },
  "topics_info" : [ {
    "topic" : "topic-sdk-no-delete",
    "partitions" : [ {
      "partition_id" : "2",
      "status" : "RUNNING",
      "last_transfer_offset" : "3",
      "log_end_offset" : "3",
      "lag" : "0"
    }
  ]
}
]
```



```
    }, {
      "partition_id" : "1",
      "status" : "RUNNING",
      "last_transfer_offset" : "3",
      "log_end_offset" : "3",
      "lag" : "0"
    }, {
      "partition_id" : "0",
      "status" : "RUNNING",
      "last_transfer_offset" : "3",
      "log_end_offset" : "3",
      "lag" : "0"
    } ]
  } ]
}
```

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowSinkTaskDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
        decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for
        authentication. Before running this example, set environment variables
        CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowSinkTaskDetailRequest request = new ShowSinkTaskDetailRequest();

        request.withTopicInfo(ShowSinkTaskDetailRequest.TopicInfoEnum.fromValue("<topic-
        info>"));
        try {
            ShowSinkTaskDetailResponse response =
            client.showSinkTaskDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
```

```
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSinkTaskDetailRequest()
        request.topic_info = "<topic-info>"
        response = client.show_sink_task_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowSinkTaskDetailRequest{
    topicInfoRequest:=
model.GetShowSinkTaskDetailRequestTopicInfoEnum().<TOPIC_INFO>
    request.TopicInfo = &topicInfoRequest
    response, err := client.ShowSinkTaskDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

Código de status

Código de status	Descrição
200	A tarefa de despejo é consultada com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

7.2.8 Exclusão de uma tarefa de despejo

Função

Essa API é usada para excluir uma tarefa de despejo.

NOTA

Esta API está desatualizada e pode não ser mantida no futuro. Use a API descrita em [Exclusão de uma tarefa do Smart Connect](#).

Método de chamada

Para obter detalhes, consulte [Como chamar uma API](#).

URI

DELETE /v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}

Tabela 7-73 Parâmetros de URI

Parâmetro	Obrigatório	Tipo	Descrição
project_id	Sim	String	ID do projeto. Para obter detalhes, consulte Obtenção de um ID de projeto.
connector_id	Sim	String	ID de despejo da instância. Para obter detalhes, consulte Consulta de IDs de instância.
task_id	Sim	String	ID da tarefa de despejo.

Parâmetros de solicitação

Nenhum

Parâmetros de resposta

Nenhum

Exemplo de solicitação

Exclusão de uma tarefa de despejo especificada

```
DELETE https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}
```

Exemplo de resposta

Nenhum

Código de exemplo do SDK

O código de exemplo do SDK é o seguinte.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteSinkTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in
        plaintext, which has great security risks. It is recommended that the AK and SK
        be stored in ciphertext in configuration files or environment variables and
```

```
decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");

    ICredential auth = new BasicCredentials()
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    DeleteSinkTaskRequest request = new DeleteSinkTaskRequest();
    try {
        DeleteSinkTaskResponse response = client.deleteSinkTask(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSinkTaskRequest()
        response = client.delete_sink_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in
    plaintext, which has great security risks. It is recommended that the AK and SK
    be stored in ciphertext in configuration files or environment variables and
    decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for
    authentication. Before running this example, set environment variables
    CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSinkTaskRequest{}
    response, err := client.DeleteSinkTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Código de status

Código de status	Descrição
204	A tarefa de despejo foi excluída com sucesso.

Códigos de erro

Consulte [Códigos de erro](#).

8 Apêndice

8.1 Código de status

Tabela 8-1 lista os códigos de status.

Tabela 8-1 Códigos de status

Código de status	Nome	Descrição
100	Continue	O servidor recebeu a parte inicial da solicitação e o cliente deve continuar enviando a parte restante.
101	Switching Protocols	O solicitante pediu ao servidor para mudar de protocolos e o servidor concordou em fazê-lo. O protocolo de destino deve ser mais avançado do que o protocolo de origem. Por exemplo, o protocolo HTTP atual é mudado para uma versão posterior do HTTP.
200	OK	Pedido enviado com sucesso.
201	Created	A solicitação foi atendida, resultando na criação de um novo recurso.
202	Accepted	A solicitação foi aceita para processamento, mas o processamento não foi concluído.
203	Non-Authoritative Information	O pedido foi cumprido.
204	NoContent	O servidor processou a solicitação com sucesso, mas não está retornando nenhum corpo de resposta. O código de status é retornado em resposta a uma solicitação HTTP OPTIONS.
205	Reset Content	O servidor cumpriu a solicitação, mas o solicitante precisa redefinir o conteúdo.

Código de status	Nome	Descrição
206	Partial Content	O servidor processou com sucesso uma parte da solicitação GET.
300	Multiple Choices	Existem várias opções para o recurso solicitado. Por exemplo, esse código pode ser usado para apresentar uma lista de características de recursos e endereços a partir dos quais o cliente, como um navegador, pode escolher.
301	Moved Permanently	Este e todos os pedidos futuros foram permanentemente transferidos para o determinado URI indicado na presente resposta.
302	Found	O recurso solicitado foi movido temporariamente.
303	See Other	A resposta à solicitação pode ser encontrada em outro URI usando um método GET ou POST.
304	Not Modified	O recurso solicitado não foi modificado. Quando o servidor retorna esse código de status, ele não retorna nenhum recurso.
305	Use Proxy	O recurso solicitado está disponível apenas através de um proxy.
306	Unused	Esse código de status HTTP não é mais usado.
400	BadRequest	Pedido inválido. O cliente deve modificar a solicitação em vez de reiniciá-la.
401	Unauthorized	As informações de autorização fornecidas pelo cliente estão incorretas ou inválidas.
402	Payment Required	Reservado para uso futuro.
403	Forbidden	O servidor recebeu a solicitação e a entendeu, mas o servidor está se recusando a responder a ela. O cliente deve modificar a solicitação em vez de reiniciá-la.
404	NotFound	O recurso solicitado não pode ser encontrado. O cliente deve modificar a solicitação em vez de reiniciá-la.
405	MethodNotAllowed	Um método de solicitação não é suportado para o recurso solicitado. O cliente deve modificar a solicitação em vez de reiniciá-la.
406	Not Acceptable	O servidor não pode atender a solicitação com base nas características de conteúdo da solicitação.

Código de status	Nome	Descrição
407	Proxy Authentication Required	Esse código é semelhante ao 401, mas indica que o cliente deve primeiro se autenticar com o proxy.
408	Request Time-out	O servidor expirou o tempo limite ao aguardar a solicitação. O cliente pode reiniciar o pedido sem qualquer modificação a qualquer momento.
409	Conflict	A solicitação não pode ser processada devido a um conflito, como um conflito de edição entre várias atualizações simultâneas ou o recurso que o cliente tenta criar já sai.
410	Gone	O recurso solicitado foi excluído permanentemente e não estará disponível novamente.
411	Length Required	O servidor se recusou a processar a solicitação porque a solicitação não especifica o comprimento de seu conteúdo.
412	Precondition Failed	O servidor não atende a uma das pré-condições que o solicitante coloca na solicitação.
413	Request Entity Too Large	O servidor se recusa a processar uma solicitação porque a solicitação é muito grande. O servidor pode fechar a conexão para impedir que o cliente continue a solicitação. Se o servidor não puder processar a solicitação temporariamente, a resposta conterá um campo Retry-After.
414	Request-URI Too Large	O URI fornecido era muito longo para o servidor processar.
415	Unsupported Media Type	O servidor não suporta o tipo de mídia na solicitação.
416	Requested range not satisfiable	O intervalo de solicitado é inválido.
417	Expectation Failed	O servidor não cumpre os requisitos do campo Esperar pedido-cabeçalho.
422	UnprocessableEntity	A solicitação é bem formada, mas não pode ser processada devido a erros semânticos.
429	TooManyRequests	O cliente enviou mais solicitações do que seu limite de taxa é permitido dentro de um determinado período de tempo, ou o servidor recebeu mais solicitações do que é capaz de processar dentro de um determinado período de tempo. Neste caso, o cliente deve reiniciar pedidos após o tempo especificado no cabeçalho Retry-After da resposta expirar.

Código de status	Nome	Descrição
500	InternalServerError	O servidor é capaz de receber a solicitação, mas não conseguiu entender a solicitação.
501	Not Implemented	O servidor não suporta a função solicitada.
502	Bad Gateway	O servidor estava agindo como um gateway ou proxy e recebeu uma solicitação inválida de um servidor remoto.
503	ServiceUnavailable	O serviço solicitado é inválido. O cliente deve modificar a solicitação em vez de reiniciá-la.
504	ServerTimeout	O pedido não pode ser atendido dentro de um determinado prazo. A resposta chegará ao cliente somente se a solicitação carregar o parâmetro timeout .
505	HTTP Version not supported	O servidor não suporta a versão do protocolo HTTP usada na solicitação.

8.2 Códigos de erro

Se um código de erro que começa com APIGW for retornado depois que você chamar uma API, corrija a falha consultando as instruções fornecidas em [Códigos de erro do API Gateway](#).

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400002	The project ID format is invalid.	ID de projeto inválido.	Verifique o formato do ID do projeto.
400	DMS.00400004	The request body is empty.	O corpo da solicitação está vazio.	Verifique o corpo da solicitação.
400	DMS.00400005	The message body is not in JSON format or contains invalid characters.	Verifique o formato do ID do projeto.	Verifique o formato do corpo da mensagem.
400	DMS.00400007	Unsupported type.	Tipo não suportado.	Verifique o tipo.
400	DMS.00400008	Unsupported version.	Versão não suportada.	Verifique a versão.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400009	Invalid product_id.	product_id inválido na solicitação.	Verifique o parâmetro product_id.
400	DMS.00400010	Invalid instance name. The name must be 4 to 64 characters long. Only letters, digits, underscores (_), and hyphens (-) are allowed.	Nome de instância inválido. O nome deve ter de 4 a 64 caracteres. Apenas letras, dígitos, sublinhados (_) e hifens (-) são permitidos.	Verifique o nome da instância.
400	DMS.00400011	The instance description can contain a maximum of 1024 characters.	A descrição da instância pode conter um máximo de 1024 caracteres.	Verifique a descrição da instância.
400	DMS.00400012	The password does not meet the complexity requirements. An instance password must be a string consisting of 8 to 32 characters. Contain at least two of the following character types: Lowercase letters, Uppercase letters, Digits, Special characters. ~!@#\$%^&*()-_+=+ [{}];:','<>/?	A senha não atende aos requisitos de complexidade. Uma senha: - Pode conter de 8 a 32 caracteres. - Deve conter pelo menos três dos seguintes tipos de caracteres: letras, dígitos e caracteres especiais ~!@#\$%^&*()-_+=+ [{}];:','<>/? - Não pode ser uma senha fraca.	Verifique se a senha atende aos requisitos.
400	DMS.00400013	vpc_id in the request is empty.	O parâmetro de solicitação vpc_id está vazio.	Verifique o parâmetro vpc_id.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400014	security_group_id in the request is empty.	O parâmetro de solicitação security_group_id está vazio.	Verifique o parâmetro security_group_id.
400	DMS.00400015	Invalid username. A username must be 4 to 64 characters long and consist of only letters, digits, and hyphens (-).	Nome de usuário inválido. Um nome de usuário deve ter de 4 a 64 caracteres e consistir apenas de letras, dígitos e hífens (-).	Verifique o nome de usuário.
400	DMS.00400016	subnet_id in the request is empty.	O parâmetro de solicitação subnet_id está vazio.	Verifique o parâmetro subnet_id.
400	DMS.00400017	This DMS instance job task is still running.	Uma tarefa em segundo plano vinculada a esta instância está em execução.	Tente novamente mais tarde.
400	DMS.00400018	This subnet must exist in the VPC.	A sub-rede deve existir na VPC.	Verifique a sub-rede.
400	DMS.00400019	The password does not meet the complexity requirements.	A senha não atende aos requisitos de complexidade.	Verifique se a senha atende aos requisitos.
400	DMS.00400020	DHCP must be enabled for this subnet.	DHCP deve ser ativado para a sub-rede.	Verifique o status do DHCP.
400	DMS.00400021	The isAutoRenew parameter in the request must be either 0 or 1.	O isAutoRenew é inválido na solicitação.	Verifique o parâmetro isAutoRenew.
400	DMS.00400022	Engine does not match the product id.	Os parâmetros engine e product ID não correspondem.	Verifique o parâmetro engine.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400026	This operation is not allowed due to the instance status.	Esta operação não é permitida quando a instância está no estado atual.	Verifique o status da instância.
400	DMS.00400028	Query advanced product, specCode not exists.	O specCode não existe durante a consulta de recurso avançada.	Verifique o parâmetro origin_spec_code.
400	DMS.00400029	Query advanced product failed, can not find product for request.	O specCode não existe durante a consulta de recurso avançada.	Verifique o parâmetro origin_spec_code.
400	DMS.00400030	Invalid DMS instance id. The id must be a uuid.	ID de instância inválido.	Verifique o parâmetro id.
400	DMS.00400035	DMS instance quota of the tenant is insufficient.	Cota da instância insuficiente.	Solicite uma cota maior.
400	DMS.00400037	The instanceParams parameter in the request contains invalid characters or is not in JSON format.	O parâmetro de solicitação instanceParams não está no formato JSON ou contém caracteres inválidos.	Verifique o parâmetro de solicitação.
400	DMS.00400038	The periodNum parameter in the request must be an integer.	O parâmetro periodNum da solicitação deve ser um número inteiro.	Verifique o parâmetro periodNum.
400	DMS.00400039	The quota limit has been reached.	O limite da cota foi atingido.	Solicite uma cota maior.
400	DMS.00400042	The AZ does not exist.	A AZ não existe.	Verifique a AZ.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400045	The instance is not frozen and cannot be unfrozen.	A instância não pode ser descongelada porque não está congelada.	Verifique o status da instância.
400	DMS.00400046	This security group does not exist.	O grupo de segurança não existe.	Verifique o grupo de segurança.
400	DMS.00400047	The periodType parameter in the request must be either 2 or 3.	O periodType inválido na solicitação.	Verifique o parâmetro periodType.
400	DMS.00400048	Invalid security group rules. Ensure that rules with the protocol being ANY are configured for both the inbound and outbound directions.	O grupo de segurança deve ter regras de saída e entrada com protocolos definidos como ANY.	Verifique as regras do grupo de segurança.
400	DMS.00400049	The availability zone does not support ipv6.	A AZ não suporta IPv6.	Selecione outra AZ.
400	DMS.00400051	not found the new setup version tar to upgrade instance.	O pacote para atualizar a instância para a versão de destino não foi encontrado.	Selecione outra versão de destino.
400	DMS.00400052	only the instance at running status can upgrade.	Somente instâncias em execução podem ser atualizadas.	Tente novamente mais tarde.
400	DMS.00400053	the upgrade instance version equals to current version.	A versão de destino é a mesma que a versão atual.	Selecione outra versão de destino.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400055	Resource sold out.	Recursos, como ECS e recursos de volume, são insuficientes.	Tente novamente mais tarde.
400	DMS.00400060	This instance name already exists.	O nome da instância já existe.	Verifique o nome da instância.
400	DMS.00400061	Invalid instance ID format.	ID de instância inválido.	Verifique o ID da instância.
400	DMS.00400062	Invalid request parameter.	Parâmetros de solicitação inválidos.	Verifique os parâmetros da solicitação.
400	DMS.00400063	Invalid configuration parameter {0}.	Parâmetro de configuração {0} inválido.	Verifique o parâmetro.
400	DMS.00400064	The action parameter in the request must be delete or restart.	O parâmetro action na solicitação deve ser delete ou restart.	Verifique o parâmetro action.
400	DMS.00400065	The instances parameter in the request is empty.	O parâmetro instances na solicitação está vazio.	Verifique o parâmetro instances.
400	DMS.00400066	Parâmetro de configuração {0} inválido.	Parâmetro de configuração {0} inválido.	Verifique o parâmetro.
400	DMS.00400067	The available_zones parameter in the request must be an array that contains only one AZ ID.	O parâmetro de solicitação available_zones deve ser uma matriz que contém apenas um ID de AZ.	Verifique o parâmetro available_zones.
400	DMS.00400068	The VPC does not exist.	A VPC não existe.	Verifique a VPC.
400	DMS.00400070	Invalid task ID format.	ID de tarefa inválido.	Verifique o ID da tarefa.
400	DMS.00400081	Duplicate instance name.	O nome da instância já existe.	Verifique o nome da instância.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400082	Instance id is repeated.	O ID da instância já existe.	Verifique o ID da instância.
400	DMS.00400085	The message body contains invalid characters or is not in JSON format. The error key is <key>.	O corpo da mensagem não está no formato JSON ou contém caracteres inválidos.	Verifique o corpo da mensagem.
400	DMS.00400099	The following instances in the Creating, Starting, Stopping, or Restarting state cannot be deleted.	As instâncias ({} no estado Creating, Starting, Stopping ou Restarting não podem ser excluídas.	Verifique o status da instância.
400	DMS.00400100	The instances array can contain a maximum of 50 instance IDs.	A matriz de instâncias pode conter um máximo de 50 IDs de instância.	Verifique a quantidade da instância.
400	DMS.00400101	The name of a Kafka topic must be 4 to 64 characters long and start with a letter. Only letters, digits, underscores (_), and hyphens (-) are allowed.	O nome de um tópico em uma instância do Kafka deve ter de 4 a 64 caracteres e começar com uma letra. Apenas letras, dígitos, sublinhados (_) e hifens (-) são permitidos.	Verifique o nome do tópico.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400102	The number of partitions created for a Kafka topic must be within the range of 1-20.	O número de partições criadas para um tópico em uma instância do Kafka deve estar dentro do intervalo de 1 a 20.	Verifique o número de partições do tópico.
400	DMS.00400103	The number of replicas created for a Kafka topic must be within the range of 1-20.	O número de réplicas criadas para um tópico em uma instância do Kafka deve estar dentro do intervalo de 1 a 20.	Verifique o número de réplicas do tópico.
400	DMS.00400105	The message retention period of a Kafka topic must be within the range of 1-168.	O tempo de envelhecimento de um tópico em uma instância do Kafka deve estar no intervalo de 1 a 168.	Verifique o tempo de envelhecimento do tópico.
400	DMS.00400106	Invalid maintenance time window.	Janela de tempo de manutenção inválida.	Verifique o parâmetro da janela de tempo de manutenção.
400	DMS.00400107	The instance exists for unpaid scale up orders. Please process non payment orders first.	Um pedido de modificação de especificação para a instância está pendente.	Processe o pedido primeiro.
400	DMS.00400108	The Instance exists for processing scale up order. Please try again later.	Um pedido de modificação de especificação para a instância está sendo processado.	Tente novamente mais tarde.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400124	The maximum number of disk expansion times has been reached.	O número máximo de tempos de expansão do disco foi atingido.	Verifique o número máximo de tempos de expansão do disco.
400	DMS.00400125	Invalid SPEC_CODE.	SPEC_CODE inválido.	Verifique SPEC_CODE.
400	DMS.00400126	Invalid period time.	Período inválido para cobrança anual/mensal.	Verifique o período de tempo para a cobrança anual/mensal.
400	DMS.00400127	Instance not support to change retention_policy.	A instância não suporta alterações na política de retenção.	Entre em contato com o suporte técnico.
400	DMS.00400128	Invalid public access parameters.	Parâmetros de acesso público inválidos.	Verifique os parâmetros de acesso público.
400	DMS.00400129	Current instance version is less than required.	A versão da instância não suporta esta operação.	Entre em contato com o suporte técnico.
400	DMS.00400133	Sink task quota for connector invalid.	Cota de tarefa do connector inválida.	Entre em contato com o suporte técnico.
400	DMS.00400134	There is another order need to pay first.	Existe um pedido não pago.	Pague o pedido primeiro.
400	DMS.00400135	Not support disk encrypted.	Não há suporte para a criptografia de disco.	Não ative a criptografia de disco.
400	DMS.00400136	Disk encrypted key is null.	A chave de criptografia de disco está vazia.	Verifique a chave de criptografia do disco.
400	DMS.00400137	Disk encrypted key state is not enabled.	A chave de criptografia de disco não está ativada.	Ative a chave de criptografia do disco.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400142	Timestamp is invalid.	O carimbo de data/hora é inválido.	Insira um carimbo de data/hora correto.
400	DMS.00400500	Invalid disk space.	Espaço em disco inválido.	Verifique o espaço em disco.
400	DMS.00400800	Invalid request parameter. Check the request parameter.	Parâmetro de solicitação inválido.	Verifique os parâmetros da solicitação.
400	DMS.00400861	Replication factor larger than available brokers.	O número de réplicas no tópico a ser criado é maior do que o número de corretores disponíveis.	Entre em contato com o suporte técnico.
400	DMS.00400867	Failed to create the Smart Connect task.	Falha ao criar a tarefa do Smart Connect.	Entre em contato com o suporte técnico.
400	DMS.00400868	Failed to stop the Smart Connect task.	Falha ao parar a tarefa do Smart Connect.	Tente novamente mais tarde.
400	DMS.00400869	Failed to start the Smart Connect task.	Falha ao iniciar a tarefa do Smart Connect.	Tente novamente mais tarde.
400	DMS.00400870	Failed to verify the Smart Connect task.	Falha ao verificar a tarefa do Smart Connect.	Tente novamente mais tarde.
400	DMS.00400872	Failed to restart the Smart Connect task.	Falha ao reiniciar a tarefa do Smart Connect.	Tente novamente mais tarde.
400	DMS.00400873	Failed to modify the Smart Connect task.	Falha ao modificar a tarefa do Smart Connect.	Entre em contato com o suporte técnico.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.00400874	The topic has been used in another Smart Connect task.	O tópico foi usado em outra tarefa do Smart Connect.	Verifique o tópico e tente novamente.
400	DMS.00400875	Inconsistent source and target Redis instance types in the Smart Connect task.	Tipos de instância do Redis de origem e destino inconsistentes na tarefa do Smart Connect.	Altere os tipos de instância do Redis de origem e destino e tente novamente.
400	DMS.00400876	The topic does not exist.	O tópico não existe.	Verifique o tópico e tente novamente.
400	DMS.00400970	RabbitMQ plugin is not exist	Nome de plug-in inválido	Verifique a lista de plug-ins.
400	DMS.00400971	The instance ssl is off.	O ssl da instância está desativado.	Visualize os detalhes da instância e verifique se o SSL está ativado.
400	DMS.00400975	Failed to query topics.	Falha ao consultar tópicos.	Verifique se o tópico existe.
400	DMS.00404033	Does not support extend rabbitMQ disk space.	O espaço em disco do RabbitMQ não pode ser expandido.	Expanda o cluster do RabbitMQ.
400	DMS.00500960	Invalid user AK/SK.	AK/SK de usuário inválida.	Corrija a AK/SK do usuário e tente novamente.
400	DMS.00500986	Your account has been restricted.	Sua conta está suspensa.	Entre em contato com a central de cobrança.
400	DMS.00500987	Balance is not enough	Saldo insuficiente.	Recarregue sua conta e tente novamente mais tarde.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.10240002	The number of queried queues exceeds the upper limit.	O número máximo de filas consultadas foi atingido.	Verifique a quantidade da fila.
400	DMS.10240004	The tag name is invalid.	Nome da tag inválido.	Verifique o nome da tag.
400	DMS.10240005	The project ID format is invalid.	ID de projeto inválido.	Verifique o formato do ID do projeto.
400	DMS.10240007	The name contains invalid characters.	O nome contém caracteres inválidos.	Verifique o nome.
400	DMS.10240009	The message body is not in JSON format or contains invalid characters.	O corpo da mensagem não está no formato JSON ou contém caracteres inválidos.	Verifique o corpo da mensagem.
400	DMS.10240010	The description contains invalid characters.	A descrição contém caracteres inválidos.	Verifique a descrição.
400	DMS.10240011	The name length must be 1 to 64 characters.	O nome pode conter de 1 a 64 caracteres.	Verifique o comprimento do nome.
400	DMS.10240012	The name length must be 1 to 32 characters.	O nome pode conter de 1 a 32 caracteres.	Verifique o comprimento do nome.
400	DMS.10240013	The description length must not exceed 160 characters.	A descrição pode conter um máximo de 160 caracteres.	Verifique o comprimento da descrição.
400	DMS.10240014	The number of consumable messages exceeds the maximum limit.	O número de mensagens consumíveis não está dentro do intervalo permitido.	Verifique o número de mensagens consumíveis.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.10240015	The queue ID format is invalid.	ID de fila inválido.	Verifique o ID da fila.
400	DMS.10240016	The group ID format is invalid.	ID de grupo inválido.	Verifique o ID do grupo.
400	DMS.10240017	The queue already exists.	A fila já existe.	Verifique se a fila existe.
400	DMS.10240018	The consumer group already exists.	O grupo de consumidores já existe.	Verifique se o grupo de consumidores existe.
400	DMS.10240019	The number of consumer groups exceeds the upper limit.	O número de grupos de consumidores excede o limite superior.	Verifique o número de grupos de consumidores.
400	DMS.10240020	The quota is insufficient.	Cota insuficiente.	Verifique a cota.
400	DMS.10240021	The value of time_wait is not within the value range of 1-60.	O valor de time_wait não está dentro do intervalo de 1 a 60.	Verifique o valor de time_wait.
400	DMS.10240022	The value of max Consume Count must be within the range of 1-100.	O valor de max Consume Count deve estar dentro do intervalo de 1 a 100.	Verifique o valor de max Consume Count.
400	DMS.10240027	The value of retention_hours must be an integer in the range of 1-72.	O valor de retention_hours deve ser um número inteiro no intervalo de 1 a 72.	Verifique o valor de retention_hours.
400	DMS.10240028	Non-kafka queues do not support retention_hours.	As filas não kafka não suportam retention_hours.	Verifique se a fila é uma fila do Kafka. Caso contrário, não defina retention_hours.
400	DMS.10240032	The queue is being created.	A fila está sendo criada.	Verifique se a fila está sendo criada.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.10240035	The tag key is empty or too long.	A chave de tag da fila está vazia ou muito longa.	Verifique a chave de tag da fila.
400	DMS.10240036	The tag key contains invalid characters.	A chave de tag da fila contém caracteres inválidos.	Verifique a chave de tag da fila.
400	DMS.10240038	The tag value is too long.	O valor da tag é muito longo.	Verifique o valor da tag da fila.
400	DMS.10240039	The tag value contains invalid characters.	O valor da tag contém caracteres inválidos.	Verifique o valor da tag da fila.
400	DMS.10240040	You can only create or delete tags.	Você só pode criar ou excluir tags.	Verifique se a operação atende aos requisitos.
400	DMS.10240041	You can only filter or count tags.	Você só pode filtrar ou contar tags.	Verifique se a operação atende aos requisitos.
400	DMS.10240042	The number of records on each page for pagination query exceeds the upper limit.	O número de registros em cada página para consulta de paginação excede o limite superior.	Verifique o tamanho da página.
400	DMS.10240043	The number of skipped records for pagination query exceeds the upper limit.	O deslocamento de consulta de paginação excede o limite superior.	Verifique o deslocamento de paginação.
400	DMS.10240044	A maximum of 10 tags can be created.	É possível criar no máximo 10 tags.	Verifique a quantidade de tags.
400	DMS.10240045	The tag key has been used.	A chave de tag foi usada.	Verifique se a chave de tag foi usada.
400	DMS.10540001	The message body contains invalid fields.	O corpo da mensagem contém campos inválidos.	Verifique o corpo da mensagem.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.10540003	Message ack status must be either 'success' or 'fail'. It should not be '{status}'.	O status de confirmação da mensagem deve ser de sucesso ou falha. Não deve ser {status}.	Verifique se o status atende aos requisitos.
400	DMS.10540004	Request error	Erro na solicitação. O queue ou group name não corresponde ao handler.	Verifique se o queue ou group name corresponde ao handler.
400	DMS.10540010	The request format is incorrect	O formato da solicitação está incorreto. {Descrição do erro}	Verifique o formato da solicitação.
400	DMS.10540011	The message size is {message size}, larger than the size limit {max allowed size}.	O tamanho da mensagem é {tamanho da mensagem}, maior que o limite de tamanho {tamanho máximo permitido}.	Verifique o tamanho do corpo da solicitação.
400	DMS.10540012	The message body is not in JSON format or contains invalid characters.	O corpo da mensagem não está no formato JSON ou contém caracteres inválidos.	Verifique o formato do corpo da mensagem.
400	DMS.10540014	The URL contains invalid parameters.	O URL contém parâmetros inválidos.	Verifique os parâmetros de URL.
400	DMS.10540202	The request format is incorrect	O formato da solicitação está incorreto. {Descrição do erro}	Verifique o formato da solicitação.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.10542204	Failed to consume messages due to {desc}.	Falha ao consumir mensagens. {Descrição do erro}	Verifique as informações de erro e corrija a falha de acordo.
400	DMS.10542205	Failed to obtain the consumption instance because the handler does not exist. This may be because the consumer instance is released 1 minute after the message is consumed. As a result, the consumer instance fails to be obtained from the handler.	Falha ao obter a instância de consumo porque o handler não existe. Isso pode ocorrer porque a instância do consumidor é liberada 1 minuto depois que a mensagem é consumida.	Verifique o handler.
400	DMS.10542206	The value of ack_wait must be within the range of 15-300.	O valor de ack_wait deve estar no intervalo de 15 a 300.	Verifique o valor de ack_wait.
400	DMS.10542209	The handler does not exist because the handler fails to be parsed, the message consumption times out, or the message consumption is repeatedly acknowledged.	O handler não existe porque o handler falha ao ser analisado, o consumo de mensagens expira ou o consumo de mensagens é repetidamente reconhecido.	Verifique se o handler ou o reconhecimento de consumo excede o tempo limite.
400	DMS.10542214	The request format is incorrect	O formato da solicitação está incorreto. {Descrição do erro}	Verifique o formato da solicitação.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
400	DMS.50050004	The consumer group is offline.	O grupo de consumidores está off-line.	Inicie a instância do consumidor no grupo de consumidores.
401	DMS.10240101	Invalid token.	Token inválido.	Verifique se o token é válido.
401	DMS.10240102	Expired token.	O token expirou.	Verifique se o token expirou.
401	DMS.10240103	Missing token.	O token está faltando.	Verifique se o token está faltando.
401	DMS.10240104	The project ID and token do not match.	O ID do projeto e o token não correspondem.	Verifique se o ID do projeto corresponde ao token.
403	DMS.00403002	A tenant has the read-only permission and cannot perform operations on DMS.	Não é possível executar operações no DMS porque você só tem permissões de leitura.	Verifique a permissão do locatário.
403	DMS.00403003	This role does not have the permissions to perform this operation.	Esta função não tem as permissões para executar esta operação.	Verifique a permissão de função.
403	DMS.10240304	Change the quota of a queue or consumer group to a value smaller than the used quota.	A cota de uma fila ou grupo de consumidores não pode ser menor do que a quantidade usada.	Verifique a cota.
403	DMS.10240306	The tenant has been frozen. You cannot perform operations on DMS.	O locatário foi congelado. Não é possível executar operações no DMS.	Verifique o status do locatário.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
403	DMS.10240307	The consumer group quota must be within the range of 1-10.	A cota do grupo de consumidores deve estar dentro do intervalo de 1 a 10.	Verifique se o número de grupos de consumidores excede a cota.
403	DMS.10240308	The queue quota must be within the range of 1-20.	A cota da fila deve estar dentro do intervalo de 1 a 20.	Verifique se o número de filas excede a cota.
403	DMS.10240309	Access denied. You cannot perform operations on DMS.	Acesso negado. Não é possível executar operações no DMS.	Verifique se você tem a permissão necessária para realizar esta operação.
403	DMS.10240310	A tenant has the read-only permission and cannot perform operations on DMS.	O locatário tem permissões somente leitura e não pode executar operações no DMS.	Verifique a permissão do locatário.
403	DMS.10240311	This role does not have the permissions to perform this operation.	Essa função não tem as permissões necessárias para executar operações no DMS.	Verifique a permissão de função.
403	DMS.10240312	The tenant is restricted and cannot perform operations on DMS.	O locatário é restrito e não pode realizar operações no DMS.	Verifique a permissão de função.
404	DMS.00404001	The requested URL does not exist.	O URL solicitado não existe.	Verifique o URL.
404	DMS.00404022	This instance does not exist.	A instância não existe.	Verifique se a instância existe.
404	DMS.00404024	Connector does not exist.	O connector não existe.	Verifique o connector.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
404	DMS.00404026	The dumping task does not exist.	A tarefa de despejo não existe.	Verifique a tarefa de despejo.
404	DMS.00404027	Connector already exists.	O connector já existe.	Verifique o connector.
404	DMS.00404029	The dumping task quota has been reached.	A cota da tarefa de despejo foi atingida.	Verifique a cota de tarefa de despejo.
404	DMS.10240401	The queue ID is incorrect or not found.	O ID da fila está incorreto ou não foi encontrado.	Verifique se o ID da fila existe e se está correto.
404	DMS.10240405	The consumption group ID is incorrect or not found.	O ID do grupo de consumo está incorreto ou não foi encontrado.	Verifique se o ID do grupo de consumidores existe e está correto.
404	DMS.10240406	The URL or endpoint does not exist.	O URL ou ponto de extremidade não existe.	Verifique se o URL ou ponto de extremidade existe e está correto.
404	DMS.10240407	The request is too frequent. Flow control is being performed. Please try again later.	A solicitação é enviada com muita frequência e o controle de fluxo está sendo executado. Tente novamente mais tarde.	Tente novamente mais tarde.
404	DMS.10240426	No tag containing this key exists.	Não existem tags que contenham esta chave.	Verifique a tag.
404	DMS.10540401	The queue name does not exist.	O nome da fila não existe.	Verifique se o nome da fila existe.
405	DMS.00405001	This request method is not allowed.	O método de solicitação não é permitido.	Verifique o método de solicitação.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
408	DMS.111501024	Query timed out	Tempo limite de consulta de mensagem	Por favor, consulte mais tarde
500	DMS.00500000	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500006	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500017	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500024	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500025	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500041	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500052	Internal service error.	Falha ao enviar o trabalho de atualização da instância.	Entre em contato com o suporte técnico.
500	DMS.00500053	Internal service error.	O nó de instância especificado não foi encontrado.	Entre em contato com o suporte técnico.
500	DMS.00500054	Internal service error.	Falha ao gerar a senha.	Entre em contato com o suporte técnico.
500	DMS.00500070	Internal service error.	Falha ao configurar a instância.	Entre em contato com o suporte técnico.
500	DMS.00500071	Internal service error.	Falha ao criar a política de backup da instância.	Entre em contato com o suporte técnico.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
500	DMS.00500094	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500106	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.00500990	Failed to update topics.	Falha ao atualizar tópicos.	Entre em contato com o suporte técnico.
500	DMS.00501000	Failed to create agency, may be you do not have the agency permission.	Falha ao criar agência	Verifique se o usuário atual tem a permissão de agência.
500	DMS.00501001	Failed to get agency roleId.	Falha ao obter o roleId da agência.	Tente novamente a solicitação mais tarde.
500	DMS.00501002	Failed to query agency roleId.	Falha ao consultar o ID da função com base no nome da função.	Verifique se o nome da função na solicitação está correto.
500	DMS.00501003	Failed to grant role to agency.	Falha ao conceder função à agência.	Tente novamente mais tarde ou entre em contato com o suporte técnico
500	DMS.00501010	The product specification does not exist.	A especificação do produto não existe.	Entre em contato com o suporte técnico.
500	DMS.00501011	Failed to query the product ID from CBC.	Falha ao consultar o ID do produto do CBC.	Entre em contato com o suporte técnico.
500	DMS.00501012	Smart Connect tasks exist.	As tarefas do Smart Connect existem.	Exclua todas as tarefas do Smart Connect.
500	DMS.10250002	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.10250003	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.

Código de status	Códigos de erro	Mensagem de erro	Descrição	Solução
500	DMS.10250004	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.10250005	Internal communication error.	Erro de comunicação interna.	Entre em contato com o suporte técnico.
500	DMS.10250006	Internal service error.	Erro de serviço interno.	Entre em contato com o suporte técnico.
500	DMS.10550035	tag_type must be either or or and.	tag_type deve ser 'or' ou 'and'.	Verifique tag_type.
501	DMS.11150102 6	Query reach maximum byte	O número total de bytes na consulta excede o limite superior.	Reduza o intervalo de tempo para garantir que o número de bytes consultados não exceda o limite ou use outros métodos para consultar dados.
503	DMS.11150102 5	Query Busy. Please try again later.	Consulta de mensagem ocupada	Por favor, consulte mais tarde

8.3 Status da instância

Tabela 8-2 Descrição do status da instância

Status	Descrição
Creating	A instância está sendo criada.
Running	A instância está sendo executada corretamente. Instâncias neste estado podem fornecer serviços.
Faulty	A instância não está sendo executada corretamente.
Restarting	A instância está sendo reiniciada.
Starting	O status entre Frozen e Running .
Changing	As especificações da instância estão sendo alteradas.
Change failed	As especificações da instância não foram alteradas.

Status	Descrição
Frozen	A instância foi congelada devido a saldo insuficiente da conta. Você pode descongelar a instância completando sua conta em My Order .
Freezing	O status entre Running e Frozen .
Upgrading	A instância está sendo atualizada.
Rolling back	A instância está sendo revertida.

8.4 Obtenção de um ID de projeto

Cenário

Um ID de projeto é necessário para alguns URLs quando uma API é chamada. Obtenha um ID de projeto usando um dos seguintes métodos:

- [Obtenção de um ID de projeto chamando uma API](#)
- [Obtenção de um ID de projeto no console](#)

Obtenção de um ID de projeto chamando uma API

Você pode obter um ID de projeto chamando a API usada para [consultar projetos com base em critérios especificados](#).

A API usada para obter um ID de projeto é **GET <https://{Endpoint}/v3/projects>**, onde *{Endpoint}* indica o ponto de extremidade do IAM. Você pode obter o ponto de extremidade do IAM em [Regiões e pontos de extremidade](#). Para obter detalhes sobre a autenticação de chamadas de API, consulte [Autenticação](#).

O seguinte é um exemplo de resposta. O valor de **id** na seção de **projects** é o ID do projeto:

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxx-xxx-xxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```


Obtenção de um ID de projeto no console

Um ID de projeto é necessário para alguns URLs quando uma API é chamada. Você pode obter um ID de projeto no console.

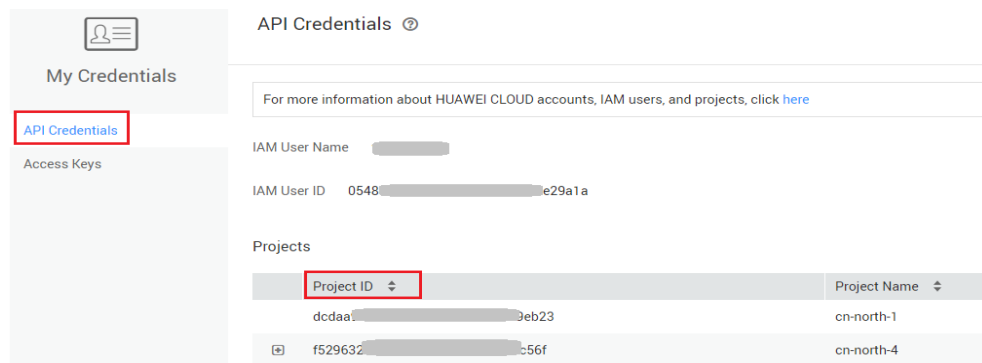
O procedimento a seguir descreve como obter um ID de projeto:

Passo 1 Faça logon no console de gerenciamento.

Passo 2 Passe o ponteiro do mouse sobre o nome de usuário no canto superior direito e escolha **My Credentials** na lista suspensa.

Na página **API Credentials**, exiba o ID do projeto na lista de projetos.

Figura 8-1 Visualização de um ID de projeto



----Fim

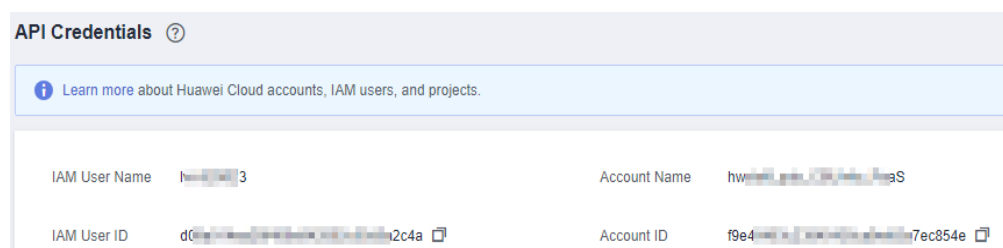
8.5 Obtenção do nome e ID da conta

O nome da conta e o ID da conta são necessários para alguns URLs quando uma API é chamada. O procedimento a seguir descreve como obter o domínio e o ID de domínio:

1. Faça logon no console de gerenciamento.
2. Passe o ponteiro do mouse sobre o nome de usuário no canto superior direito e escolha **My Credentials** na lista suspensa.

Veja o nome da conta e o ID da conta.

Figura 8-2 Visualização do nome e ID da conta



A Histórico de alterações

Lançado em	Descrição
06/03/2024	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição de APIs do Smart Connect.● Mudança das APIs de despejo para APIs desatualizadas.
26/02/2024	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição da API para Produção de mensagens.
10/11/2023	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição das APIs de Obtenção de configurações de instância e Modificação das configurações de instância.
26/07/2023	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição das APIs de Consulta da lista de partições de um tópico e Consulta da lista de produtores atuais de um tópico.
06/06/2023	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição da API para criar um grupo de consumidores.
23/04/2023	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição da API para excluir grupos de consumidores em uma instância do Kafka em lotes.
03/02/2023	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Atualização dos URIs em Criação de uma instância, Modificação de especificações de instância e Consulta de informações do produto para modificação de especificações de instância para suportar novas especificações de instância.
30/06/2022	Modificação do seguinte conteúdo: <ul style="list-style-type: none">● Adição da descrição de novas especificações ao parâmetro specification em Criação de uma instância.● Adição do parâmetro broker_num em Criação de uma instância.

Lançado em	Descrição
14/12/2021	Modificação do seguinte conteúdo: <ul style="list-style-type: none"><li data-bbox="660 338 1362 405">● Alteração das APIs de V1 para V2 em Permissões e ações suportadas.
16/11/2021	Modificação do seguinte conteúdo: <ul style="list-style-type: none"><li data-bbox="660 472 927 506">● Adição de APIs V2.
13/10/2020	Esta edição é o 1º lançamento oficial.