

配置审计

SDK 参考

文档版本 01
发布日期 2025-03-26



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 SDK 概述..... 1

2 使用 SDK 为 Config 创建合规规则..... 2

3 使用 SDK 为 Config 查询资源详情、资源关系和资源历史..... 4

1 SDK 概述

本文介绍了Config服务提供的SDK语言版本，列举了最新版本SDK的获取地址。

在线生成 SDK 代码

[API Explorer](#)能根据需要动态生成SDK代码功能，降低您使用SDK的难度，推荐使用。

SDK 列表

[表1-1](#)提供了Config服务支持的SDK列表，您可以在GitHub仓库查看SDK更新历史、获取安装包以及查看指导文档。

表 1-1 SDK 列表

编程语言	Github地址	参考文档
Java	huaweicloud-sdk-java-v3	Java SDK使用指导
Python	huaweicloud-sdk-python-v3	Python SDK使用指导
Go	huaweicloud-sdk-go-v3	Go SDK使用指导
NodeJs	huaweicloud-sdk-nodejs-v3	NodeJs SDK使用指导
PHP	huaweicloud-sdk-php-v3	PHP SDK使用指导

2 使用 SDK 为 Config 创建合规规则

示例简介

该示例展示了如何通过Java版本SDK为Config服务创建合规规则和查询合规规则。

合规规则通过指定合规策略和合规策略所应用的范围（如：在某一区域的某些资源）来构成合规规则。

开发前准备

1. 获取华为云开发工具包（SDK），您也可以查看安装JAVA SDK。
2. 您需要拥有华为云账号以及该账号对应的 Access Key（AK）和 Secret Access Key（SK），请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的 AK/SK。具体请参见[访问密钥](#)。
3. **配置审计 Config SDK**，支持 Java JDK 1.8 及其以上版本。

安装 SDK

您可以通过Maven方式获取和安装SDK，您只需要在Java项目的pom.xml文件中加入相应的依赖项即可。具体的SDK版本号请参见[SDK开发中心](#)。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-config</artifactId>
  <version>{sdk-version}</version>
</dependency>
```

代码示例

```
public class CreatePolicyAssignmentDemo {
    public static void main(String[] args) {
        // 认证用的ak和sk直接写到代码中有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");
        String regionId = "<region id>";
        HttpConfig config = HttpConfig.getDefaultHttpConfig();
        config.withIgnoreSSLVerification(true);

        ICredential auth = new GlobalCredentials().withAk(ak).withSk(sk);
        ConfigClient client = ConfigClient.newBuilder().withHttpConfig(config).withCredential(auth)
            .withRegion(ConfigRegion.valueOf(regionId)).build();
```

```
        CreatePolicyAssignmentsRequest createRequest = new CreatePolicyAssignmentsRequest()
            .withBody(new
PolicyAssignmentRequestBody().withPolicyAssignmentType(PolicyAssignmentRequestBody.PolicyAssignment
TypeEnum.BUILTIN)
            .withName("<Your policy_assignment_name>").withDescription("<Your
policy_assignment_description>")
            .withPolicyFilter(new PolicyFilterDefinition()).withPolicyDefinitionId("<Your
policy_definition_id>"));

        try {
            CreatePolicyAssignmentsResponse createResponse = client.createPolicyAssignments(createRequest);
            System.out.println(createResponse.toString());
            ShowPolicyAssignmentRequest showPolicyAssignmentRequest = new
ShowPolicyAssignmentRequest()
                .withPolicyAssignmentId(createResponse.getId());
            ShowPolicyAssignmentResponse showResponse =
client.showPolicyAssignment(showPolicyAssignmentRequest);
            System.out.println(showResponse.toString());
        } catch (ConnectionException | RequestTimeoutException | ServiceResponseException ex) {
            System.out.println(ex);
        }
    }
}
```

返回结果示例

```
class CreatePolicyAssignmentsResponse {
    policyAssignmentType: "policyAssignmentType"
    id: "id"
    name: "name"
    description: "description"
    policyFilter: class PolicyFilterDefinition {}
    period: "period"
    state: "state"
    created: "created"
    updated: "updated"
    policyDefinitionId: "policyDefinitionId"
    customPolicy: "customPolicy"
    parameters: {}
    createdBy: "createdBy"
}

class ShowPolicyAssignmentResponse {
    policyAssignmentType: "policyAssignmentType"
    id: "id"
    name: "name"
    description: "description"
    policyFilter: class PolicyFilterDefinition {}
    period: "period"
    state: "state"
    created: "created"
    updated: "updated"
    policyDefinitionId: "policyDefinitionId"
    customPolicy: "customPolicy"
    parameters: {}
    createdBy: "createdBy"
}
```

参考

更多信息请参考[资源合规概述](#)。

3

使用 SDK 为 Config 查询资源详情、资源关系和资源历史

示例简介

该示例展示了如何通过Java版本SDK查询资源详情、资源关系和资源历史。

1. **资源清单** 默认展示资源的部分属性，如果您需要查看某个资源的资源详情，可按如下操作查看。
2. **资源关系** 记录了您在华为云上的不同资源之间的关联情况。
3. **资源历史** 是过去某段时间内资源不同状态的集合。对接服务上报Config的资源属性和资源关系的变化，都会在资源时间线中生成一条记录，该记录会包含资源变更情况的详细信息，默认的保存期限为7年。

开发前准备

1. 获取华为云开发工具包（SDK），您也可以查看安装JAVA SDK。
2. 您需要拥有华为云账号以及该账号对应的 Access Key（AK）和 Secret Access Key（SK），请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见[访问密钥](#)。
3. **配置审计 Config SDK**，支持 Java JDK 1.8 及其以上版本。

安装 SDK

您可以通过Maven方式获取和安装SDK，您只需要在Java项目的pom.xml文件中加入相应的依赖项即可。具体的SDK版本号请参见[SDK开发中心](#)。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-config</artifactId>
  <version>{sdk-version}</version>
</dependency>
```

代码示例

```
public class ShowResourceRelationDemo {
    public static void main(String[] args) {
        // 认证用的ak和sk直接写到代码中有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
```

```
String ak = System.getenv("HUAWEICLOUD_SDK_AK");
String sk = System.getenv("HUAWEICLOUD_SDK_SK");
String regionId = "<region id>";
HttpConfig config = HttpConfig.getDefaultHttpConfig();
config.withIgnoreSSLVerification(true);

ICredential auth = new GlobalCredentials().withAk(ak).withSk(sk);
ConfigClient client = ConfigClient.newBuilder().withHttpConfig(config).withCredential(auth)
    .withRegion(ConfigRegion.valueOf(regionId)).build();

try {
    String resourceid = "<resource id>";
    // 查询资源详情
    ShowResourceDetailRequest resourceDetailRequest = new ShowResourceDetailRequest()
        .withResourceid(resourceid);
    System.out.println(client.showResourceDetail(resourceDetailRequest));
    // 查询资源关系
    ShowResourceRelationsRequest resourceRelationsRequest = new ShowResourceRelationsRequest()
        .withResourceid(resourceid)
        .withDirection(ShowResourceRelationsRequest.DirectionEnum.IN);
    System.out.println(client.showResourceRelations(resourceRelationsRequest).toString());
    // 查询资源历史
    ShowResourceHistoryRequest resourceHistoryRequest = new ShowResourceHistoryRequest()
        .withResourceid(resourceid);
    System.out.println(client.showResourceHistory(resourceHistoryRequest).toString());
} catch (ConnectionException | RequestTimeoutException | ServiceResponseException ex) {
    System.out.println(ex);
}
}
```

返回结果示例

```
class ShowResourceDetailResponse {
    id: 81fa****a864
    name: zh****ng
    provider: iam
    type: users
    regionId: global
    projectId:
    projectName:
    epId: 0
    epName: default
    checksum: 522u****e689
    created: 2023-09-18T12:56:30.000Z
    updated: 2023-09-18T12:56:30.000Z
    provisioningState: Succeeded
    state: Normal
    tags: {}
    properties: {pwd_status=false, pwd_strength=high, group_list=[f588****54c5], role_list=[],
last_login_time=2023-09-18T12:57:45Z, virtual_mfa_device=false, login_protect={enabled=false},
credentials=[], policy_list=[], access_mode=default, is_root_user=false, enabled=true}
}

class ShowResourceRelationsResponse {
    relations: [class ResourceRelation {
        relationType: contains
        fromResourceType: iam.groups
        toResourceType: iam.users
        fromResourceid: f587****54c5
        toResourceid: 81fa****a864
    }]
    pageInfo: class PageInfo {
        currentCount: 1
        nextMarker: null
    }
}

class ShowResourceHistoryResponse {
```



```
items: [class HistoryItem {
  domainId: 39f4****ea39
  resourceId: 81fa****a864
  resourceType: iam.users
  captureTime: 2023-09-21T15:39:27.632Z
  status: ResourceChanged.CREATE
  relations: [class ResourceRelation {
    relationType: isContainedIn
    fromResourceType: iam.users
    toResourceType: iam.groups
    fromResourceId: 81fa****a864
    toResourceId: b04e****8dd2
  }]
  resource: class ResourceEntity {
    id: 81fa****a864
    name: zh****ng
    provider: iam
    type: users
    regionId: global
    projectId:
    projectName:
    epId: 0
    epName: default
    checksum: 00ce****f053
    created: 2023-09-18T12:56:30Z
    updated: 2023-09-18T12:56:30Z
    provisioningState: Succeeded
    state: null
    tags: {}
    properties: {pwd_status=false, pwd_strength=high, group_list=[b04e****8dd2], role_list=[],
virtual_mfa_device=false, login_protect={enabled=false}, credentials=[], policy_list=[], access_mode=default,
enabled=true}
  }
}]
pageInfo: class PageInfo {
  currentCount: 1
  nextMarker: null
}
```

参考

更多信息请参考[查看资源历史](#)。