SoftWare Repository for Container

SDK Reference (Enterprise Edition)

Issue	01
Date	2026-11-26



HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Copyright © Huawei Technologies Co., Ltd. 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 SDK Overview....1
2 Obtaining SDK Sample Code....2
3 Java SDK Usage Example....4
4 Python SDK Usage Example....7
5 Go SDK Usage Example....9
6 PHP SDK Usage Example....11
7 Node.js SDK Usage Example....13
8 .NET SDK Usage Example....15

 Copyright © Huawei Technologies Co., Ltd.

1 SDK Overview

This document lists the Software Development Kits (SDKs) provided by SoftWare Repository for Container (SWR) and the addresses for obtaining the latest versions of these SDKs.

For details about the SDK privacy data processing policies, see **Huawei Cloud SDK Privacy Statement**.

SDKs

Before you start, make sure you have installed the latest SDKs. Outdated versions may cause compatibility issues or lack the latest functions. You can query the version information in the **SDK Center**.

Table 1-1 lists the SDKs supported by each cloud service. You can view the SDK update history, obtain the SDK installation packages, and view SDK user guides on GitHub.

NOTE

If you cannot access GitHub, check whether you have access to the Internet.

Table 1-1 SDKs

Programming Language	GitHub Address	Reference Document
Java	**huaweicloud-sdk-java-v3**	**Java SDK User Guide**
Python	**huaweicloud-sdk-python-v3**	**Python SDK User Guide**
Go	**huaweicloud-sdk-go-v3**	**Go SDK User Guide**
PHP	**huaweicloud-sdk-php-v3**	**Php SDK User Guide**
Node.js	**huaweicloud-sdk-nodejs-v3**	**Node.js SDK User Guide**
.NET	**huaweicloud-sdk-net-v3**	**.Net SDK User Guide**

 Copyright © Huawei Technologies Co., Ltd.

2 Obtaining SDK Sample Code

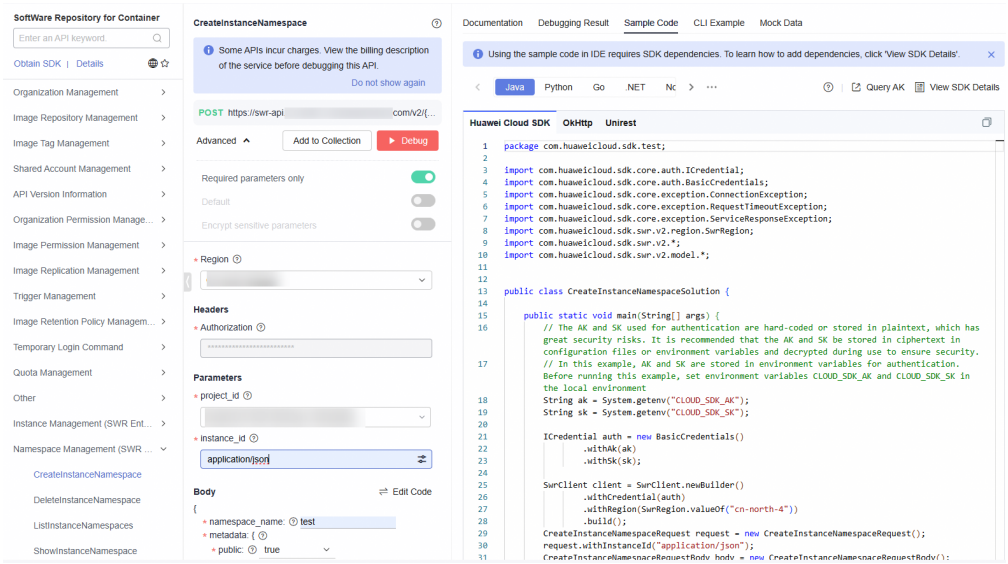
Huawei Cloud allows you to obtain the SDK sample API code in the following ways to simplify SDK usage and improve development efficiency.

API Explorer can dynamically generate SDK sample code as needed and supports multiple programming languages. You are advised to use **API Explorer** to obtain the SDK sample code.

Obtaining SDK Sample Code Through API Explorer

1. On the **API Explorer** page of SWR, select the API for which you want to generate SDK sample code.
2. In the parameter configuration interface in the middle, enter the parameters required to call the API. For details about parameter settings, click the **Documentation** tab on the right.
3. On the right of the page, click the **Sample Code** tab to view and obtain the SDK sample code in the corresponding programming language, as shown in **Figure 2-1**.

Figure 2-1 Obtaining sample code through API Explorer



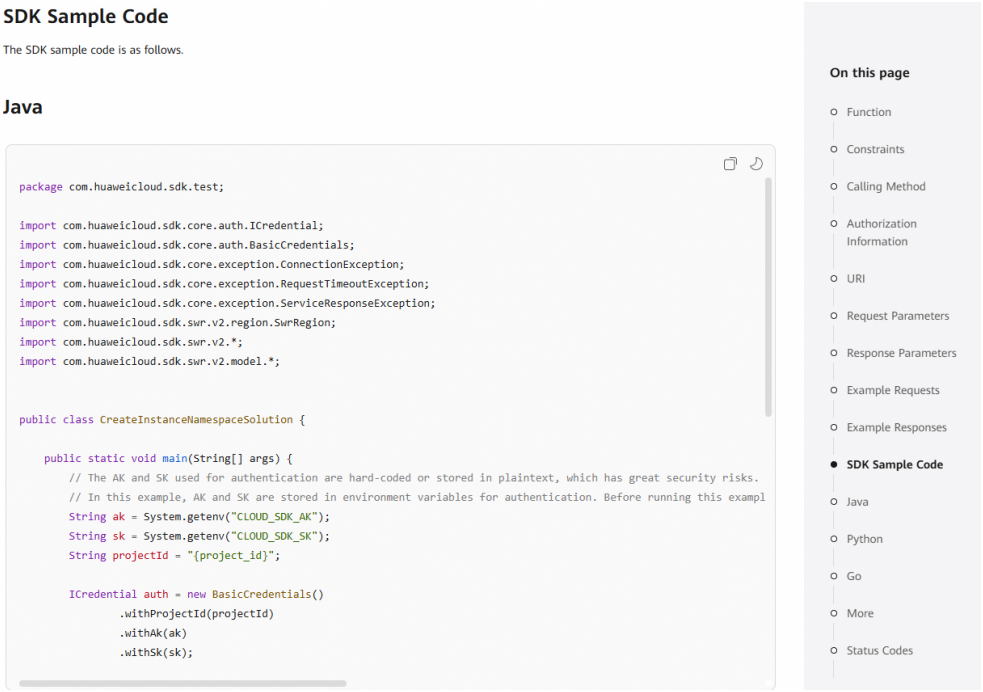
 Copyright © Huawei Technologies Co., Ltd.

Obtaining SDK Sample Code from the API Reference

1. In the **API Reference (Enterprise Edition)** of SWR, locate the API page from which you want to obtain the SDK sample code.
2. In the navigation pane on the right, click **SDK Sample Code** to view the SDK sample code, as shown in **Figure 2-2**.

   The current document provides only the SDK sample code in Java, Python, and Go. The sample code is generated based on the example request values. If you need sample code in other programming languages or want to set custom parameters in the sample code, see **Obtaining SDK Sample Code Through API Explorer**.

Figure 2-2 Obtaining sample code from the document



 Copyright © Huawei Technologies Co., Ltd.

3 Java SDK Usage Example

This section describes how to use the Huawei Cloud Java SDK by calling the CreateInstanceNamespace API of SWR.

Prerequisites

- You have obtained the access key (AK) and secret access key (SK) corresponding to your Huawei Cloud account. For details, see **Creating Access Keys**.
- **JDK 1.8** or a later version has been installed in the environment where the SDK runs, and the **Maven** environment has also been installed and configured.

Installing the SDK

You are advised to install the Huawei Cloud Java SDK using Maven. The following uses Maven as an example.

Open the **pom.xml** file of the Maven project and add the following dependency configuration under the **<dependencies>** node. Maven will automatically download the required SDK package.

The Java SDK 3.1.219 is used as an example. You can obtain the latest SDK version from **SDK Center**.

- **Installing the SDK package of a single cloud service**

```
<dependency>
    <groupId>com.huaweicloud.sdk</groupId>
    <artifactId>huaweicloud-sdk-swr</artifactId>
    <version>3.1.219</version>
</dependency>
```

- **Installing the SDK package of a cloud service collection**

```
<dependency>
    <groupId>com.huaweicloud.sdk</groupId>
    <artifactId>huaweicloud-sdk-all</artifactId>
    <version>3.1.219</version>
</dependency>
```

- **Installing the SDK bundle package of a cloud service collection**

  If there is a third-party library conflict, you can import the bundle package (3.0.40-rc or later). It contains all supported services and redirects the third-party software that the SDK depends on, preventing conflicts with the libraries that the service itself depends on.

 Copyright © Huawei Technologies Co., Ltd.

```
<dependency>
    <groupId>com.huaweicloud.sdk</groupId>
    <artifactId>huaweicloud-sdk-bundle</artifactId>
    <version>3.1.219</version>
</dependency>
```

CAUTION

The bundle package already contains the core package and the SDK package of all cloud services. There is no need to import the core package and service package separately. Due to the Maven dependency resolution order, adding the core package and the service package may prevent the bundle package from taking effect.

Using the SDK

The following is a complete example demonstrating how to use the Huawei Cloud Java SDK to call the CreateInstanceNamespace API.

- To obtain the SDK example code for other APIs, see **Obtaining SDK Sample Code**.
- This example provides only the basic API parameter settings. For more advanced settings, see the **User Manual** of the SDK.
- There will be huge security risks if the AK and SK used for authentication are written into the code. You are advised to encrypt the AK and SK and store them in a configuration file or environment variables, and decrypt them when needed to ensure security. In this example, the AK and SK are stored as the environment variables **CLOUD_SDK_AK** and **CLOUD_SDK_SK**. For details, see **Environment Variables**.

```
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.swr.v2.region.SwrRegion;
import com.huaweicloud.sdk.swr.v2.*;
import com.huaweicloud.sdk.swr.v2.model.*;


public class CreateInstanceNamespaceSolution {

    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);

        SwrClient client = SwrClient.newBuilder()
                .withCredential(auth)
                .withRegion(SwrRegion.valueOf("cn-north-4"))
```

 Copyright © Huawei Technologies Co., Ltd.

```
            .build();
        CreateInstanceNamespaceRequest request = new CreateInstanceNamespaceRequest();
        request.withInstanceId("instance_id");
        CreateInstanceNamespaceRequestBody body = new CreateInstanceNamespaceRequestBody();
        NamespaceMetadata metadatabody = new NamespaceMetadata();
        metadatabody.withPublic(NamespaceMetadata.PublicEnum.fromValue("true"));
        body.withMetadata(metadatabody);
        body.withNamespaceName("test");
        request.withBody(body);
        try {
            CreateInstanceNamespaceResponse response = client.createInstanceNamespace(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());
        }
    }
}
```

 Copyright © Huawei Technologies Co., Ltd.

4 Python SDK Usage Example

This section describes how to use the Huawei Cloud Python SDK by calling the CreateInstanceNamespace API of SWR.

Prerequisites

- You have obtained the AK and SK corresponding to your Huawei Cloud account. For details, see **Creating Access Keys**.
- **Python 3.6** or a later version has been installed in the environment where the SDK runs.

Installing the SDK

Run the command below in the CLI to install the Huawei Cloud Python SDK. The Python SDK 3.1.217 is used as an example. You can obtain the latest SDK version from **SDK Center**.

- **Installing the SDK package of a single cloud service**
  ```
  pip install huaweicloudsdkswr==3.1.217
  ```
- **Installing the SDK package of a cloud service collection**
  ```
  pip install huaweicloudsdkall==3.1.217
  ```

Using the SDK

The following is a complete example demonstrating how to use the Huawei Cloud Python SDK to call the CreateInstanceNamespace API.

- To obtain the SDK example code for other APIs, see **Obtaining SDK Sample Code**.
- This example provides only the basic API parameter settings. For more advanced settings, see the **User Manual** of the SDK.
- There will be huge security risks if the AK and SK used for authentication are written into the code. You are advised to encrypt the AK and SK and store them in a configuration file or environment variables, and decrypt them when needed to ensure security. In this example, the AK and SK are stored as the environment variables **CLOUD_SDK_AK** and **CLOUD_SDK_SK**. For details, see **Environment Variables**.
  ```
  # coding: utf-8
  ```

 Copyright © Huawei Technologies Co., Ltd.

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkswr.v2.region.swr_region import SwrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkswr.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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = SwrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(SwrRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateInstanceNamespaceRequest()
        request.instance_id = "instance_id"
        metadatabody = NamespaceMetadata(
            public="true"
        )
        request.body = CreateInstanceNamespaceRequestBody(
            metadata=metadatabody,
            namespace_name="test"
        )
        response = client.create_instance_namespace(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

 Copyright © Huawei Technologies Co., Ltd.

5 Go SDK Usage Example

This section describes how to use the Huawei Cloud Go SDK by calling the CreateInstanceNamespace API of SWR.

Prerequisites

- You have obtained the AK and SK corresponding to your Huawei Cloud account. For details, see **Creating Access Keys**.
- **Go 1.14** or a later version has been installed in the environment where the SDK runs.

Installing the SDK

Run the command below in the CLI to install the Huawei Cloud Go SDK. The Go SDK 0.1.218 is used as an example. You can obtain the latest SDK version from **SDK Center**.

```
go get github.com/huaweicloud/huaweicloud-sdk-go-v3@v0.1.218
```

Using the SDK

The following is a complete example demonstrating how to use the Huawei Cloud Go SDK to call the CreateInstanceNamespace API.

- To obtain the SDK example code for other APIs, see **Obtaining SDK Sample Code**.
- This example provides only the basic API parameter settings. For more advanced settings, see the **User Manual** of the SDK.
- There will be huge security risks if the AK and SK used for authentication are written into the code. You are advised to encrypt the AK and SK and store them in a configuration file or environment variables, and decrypt them when needed to ensure security. In this example, the AK and SK are stored as the environment variables **CLOUD_SDK_AK** and **CLOUD_SDK_SK**. For details, see **Environment Variables**.

```
package main

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

 Copyright © Huawei Technologies Co., Ltd.

```
    swr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/swr/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/swr/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/swr/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, err := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    hcClient, err := swr.SwrClientBuilder().
         WithRegion(region.ValueOf("cn-north-4")).
         WithCredential(auth).
         SafeBuild()


    if err != nil {
        fmt.Println(err)
        return
    }

    client := swr.NewSwrClient(hcClient)

    request := &model.CreateInstanceNamespaceRequest{}
    request.InstanceId = "instance_id"
    metadatabody := &model.NamespaceMetadata{
        Public: model.GetNamespaceMetadataPublicEnum().TRUE,
    }
    request.Body = &model.CreateInstanceNamespaceRequestBody{
        Metadata: metadatabody,
        NamespaceName: "test",
    }
    response, err := client.CreateInstanceNamespace(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

 Copyright © Huawei Technologies Co., Ltd.

6 PHP SDK Usage Example

This section describes how to use the Huawei Cloud PHP SDK by calling the CreateInstanceNamespace API of SWR.

Prerequisites

- You have obtained the AK and SK corresponding to your Huawei Cloud account. For details, see **Creating Access Keys**.
- **PHP 5.6** or a later version has been installed in the environment where the SDK runs.

Installing the SDK

You are advised to use Composer to install the Huawei Cloud PHP SDK. Composer is the dependency management tool for PHP. It allows you to declare and install dependencies in a project.

Run the commands below in the CLI to install Composer and Huawei Cloud PHP SDK. The PHP SDK 3.1.203 is used as an example. For details about the latest SDK version, visit **SDK Center**.

```
# Install Composer. If Composer has been installed, skip this step.
curl -sS https://getcomposer.org/installer | php

# Install the PHP SDK.
composer require huaweicloud/huaweicloud-sdk-php:3.1.203
```

Using the SDK

The following is a complete example demonstrating how to use the Huawei Cloud PHP SDK to call the CreateInstanceNamespace API.

- To obtain the SDK example code for other APIs, see **Obtaining SDK Sample Code**.
- This example provides only the basic API parameter settings. For more advanced settings, see the **User Manual** of the SDK.
- There will be huge security risks if the AK and SK used for authentication are written into the code. You are advised to encrypt the AK and SK and store them in a configuration file or environment variables, and decrypt them when needed to ensure security. In this example, the AK and SK are stored as the

 Copyright © Huawei Technologies Co., Ltd.

environment variables **CLOUD_SDK_AK** and **CLOUD_SDK_SK**. For details, see **Environment Variables**.

```
<?php
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
use HuaweiCloud\SDK\Swr\V2\SwrClient;
use HuaweiCloud\SDK\Swr\V2\Model\CreateInstanceNamespaceRequest;
use HuaweiCloud\SDK\Swr\V2\Model\CreateInstanceNamespaceRequestBody;
use HuaweiCloud\SDK\Swr\V2\Model\NamespaceMetadata;

// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are 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 = getenv('CLOUD_SDK_AK');
$sk = getenv('CLOUD_SDK_SK');
$endpoint = "https://swr-api.cn-north-4.myhuaweicloud.com";
$projectId = "067cf8aecf3845d1ad7a5d308322f13b";
$credentials = new BasicCredentials($ak,$sk,$projectId);
$config = HttpConfig::getDefaultConfig();
$config->setIgnoreSslVerification(true);

$client = SwrClient::newBuilder(new SwrClient)
  ->withHttpConfig($config)
  ->withEndpoint($endpoint)
  ->withCredentials($credentials)
  ->build();
$request = new CreateInstanceNamespaceRequest();
$request->setInstanceId("instance_id");

$body = new CreateInstanceNamespaceRequestBody();
$metadatabody = new NamespaceMetadata();
$metadatabody->setPublic("true");
$body->setMetadata($metadatabody);
$body->setNamespaceName("test");
$request->setBody($body);
try {
  $response = $client->CreateInstanceNamespace($request);
  echo "\n";
  echo $response;
} catch (ConnectionException $e) {
  $msg = $e->getMessage();
  echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
  $msg = $e->getMessage();
  echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
  echo "\n";
  echo $e->getHttpStatusCode(). "\n";
  echo $e->getRequestId(). "\n";
  echo $e->getErrorCode() . "\n";
  echo $e->getErrorMsg() . "\n";
}
```

 Copyright © Huawei Technologies Co., Ltd.

7 Node.js SDK Usage Example

This section describes how to use the Huawei Cloud Node.js SDK by calling the CreateInstanceNamespace API of SWR.

Prerequisites

- You have obtained the AK and SK corresponding to your Huawei Cloud account. For details, see **Creating Access Keys**.
- **Node.js 10.16.1** or a later version has been installed in the environment where the SDK runs.

Installing the SDK

Run the commands below in the CLI to install the Huawei Cloud Node.js SDK. The Node.js SDK 3.1.217 is used as an example. You can obtain the latest SDK version from **SDK Center**.

- **Installing the SDK package of a single cloud service**
```
npm i @huaweicloud/huaweicloud-sdk-core@3.1.217
npm i @huaweicloud/huaweicloud-sdk-swr@3.1.217
```
- **Installing the SDK package of a cloud service collection**
```
npm i @huaweicloud/huaweicloud-sdk-all@3.1.217
```

Using the SDK

The following is a complete example demonstrating how to use the Huawei Cloud Node.js SDK to call the CreateInstanceNamespace API.

- To obtain the SDK example code for other APIs, see **Obtaining SDK Sample Code**.
- This example provides only the basic API parameter settings. For more advanced settings, see the **User Manual** of the SDK.
- There will be huge security risks if the AK and SK used for authentication are written into the code. You are advised to encrypt the AK and SK and store them in a configuration file or environment variables, and decrypt them when needed to ensure security. In this example, the AK and SK are stored as the environment variables **CLOUD_SDK_AK** and **CLOUD_SDK_SK**. For details, see **Environment Variables**.
```
const core = require('@huaweicloud/huaweicloud-sdk-core');
const swr = require("@huaweicloud/huaweicloud-sdk-swr/v2/public-api");
```

 Copyright © Huawei Technologies Co., Ltd.

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
const ak = process.env.CLOUD_SDK_AK;
const sk = process.env.CLOUD_SDK_SK;
const endpoint = "https://swr-api.cn-north-4.myhuaweicloud.com";
const project_id = "067cf8aecf3845d1ad7a5d308322f13b";

const credentials = new core.BasicCredentials()
                    .withAk(ak)
                    .withSk(sk)
                    .withProjectId(project_id)
const client = swr.SwrClient.newBuilder()
                        .withCredential(credentials)
                        .withEndpoint(endpoint)
                        .build();
const request = new swr.CreateInstanceNamespaceRequest();
request.instanceId = "instance_id";
const body = new swr.CreateInstanceNamespaceRequestBody();
const metadatabody = new swr.NamespaceMetadata();
metadatabody.withPublic("true");
body.withMetadata(metadatabody);
body.withNamespaceName("test");
request.withBody(body);
const result = client.createInstanceNamespace(request);
result.then(result => {
    console.log("JSON.stringify(result)::" + JSON.stringify(result));
}).catch(ex => {
    console.log("exception:" + JSON.stringify(ex));
});
```

 Copyright © Huawei Technologies Co., Ltd.

8 .NET SDK Usage Example

This section describes how to use the Huawei Cloud .NET SDK by calling the CreateInstanceNamespace API of SWR.

Prerequisites

- You have obtained the AK and SK corresponding to your Huawei Cloud account. For details, see **Creating Access Keys**.
- **.NET Framework 4.7.2** or a later version, or **.NET 8.0** or a later version has been installed in the environment where the SDK runs.

Installing the SDK

You can install the Huawei Cloud .NET SDK using the following methods. .NET SDK 3.1.213 is used as an example. You can obtain the latest SDK version from **SDK Center**.

- **Use .NET CLI to install the SDK.**
```
dotnet add package HuaweiCloud.SDK.Core --version 3.1.213
dotnet add package HuaweiCloud.SDK.Swr --version 3.1.213
```
- **Use Package Manager to install the SDK.**
```
Install-Package HuaweiCloud.SDK.Core -Version 3.1.213
Install-Package HuaweiCloud.SDK.Swr -Version 3.1.213
```

Using the SDK

The following is a complete example demonstrating how to use the Huawei Cloud .NET SDK to call the CreateInstanceNamespace API.

- To obtain the SDK example code for other APIs, see **Obtaining SDK Sample Code**.
- This example provides only the basic API parameter settings. For more advanced settings, see the **User Manual** of the SDK.
- There will be huge security risks if the AK and SK used for authentication are written into the code. You are advised to encrypt the AK and SK and store them in a configuration file or environment variables, and decrypt them when needed to ensure security. In this example, the AK and SK are stored as the environment variables **CLOUD_SDK_AK** and **CLOUD_SDK_SK**. For details, see **Environment Variables**.

 Copyright © Huawei Technologies Co., Ltd.

```
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.Swr;
using HuaweiCloud.SDK.Swr.V2;
using HuaweiCloud.SDK.Swr.V2.Model;

namespace CreateInstanceNamespaceSolution
{
    class Program
    {
        static async Task 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 = Environment.GetEnvironmentVariable("CLOUD_SDK_AK");
            string sk = Environment.GetEnvironmentVariable("CLOUD_SDK_SK");

            var config = HttpConfig.GetDefaultConfig();
            config.IgnoreSslVerification = true;
            var auth = new BasicCredentials(ak, sk);

            var client = SwrAsyncClient.NewBuilder()
                  .WithCredential(auth)
                  .WithRegion(SwrRegion.ValueOf("cn-north-4"))
                  .WithHttpConfig(config)
                  .Build();

            var req = new CreateInstanceNamespaceRequest
            {
                InstanceId = "instance_id"
            };
            NamespaceMetadata metadatabody = new NamespaceMetadata()
            {
                Public = NamespaceMetadata.PublicEnum.FromValue("true")
            };
            req.Body = new CreateInstanceNamespaceRequestBody()
            {
                Metadata = metadatabody,
                NamespaceName = "test"
            };

            try
            {
                var resp = await client.CreateInstanceNamespace(req);
                var respStatusCode = resp.HttpStatusCode;
                Console.WriteLine(respStatusCode);
            }
            catch (RequestTimeoutException requestTimeoutException)
            {
                Console.WriteLine(requestTimeoutException.ErrorMessage);
            }
            catch (ServiceResponseException clientRequestException)
            {
                Console.WriteLine(clientRequestException.HttpStatusCode);
                Console.WriteLine(clientRequestException.RequestId);
                Console.WriteLine(clientRequestException.ErrorCode);
                Console.WriteLine(clientRequestException.ErrorMsg);
            }
            catch (ConnectionException connectionException)
            {
                Console.WriteLine(connectionException.ErrorMessage);
            }
        }
```

 Copyright © Huawei Technologies Co., Ltd.

```
    }
}
```