

# ImageSearch SDK

**Issue** 01  
**Date** 2023-03-02



**Copyright © Huawei Technologies Co., Ltd. 2023. 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**



HUAWEI 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.

## **Huawei Technologies Co., Ltd.**

Address: Huawei Industrial Base  
Bantian, Longgang  
Shenzhen 518129  
People's Republic of China

Website: <https://www.huawei.com>

Email: [support@huawei.com](mailto:support@huawei.com)

---

# Contents

---

<b>1 Introduction to ImageSearch SDK.....</b>	<b>1</b>
<b>2 Obtaining Authentication Information.....</b>	<b>3</b>
<b>3 Preparing the Environment.....</b>	<b>4</b>
<b>4 Using the SDK (Java).....</b>	<b>5</b>
4.1 Preparing a Java Development Environment.....	5
4.2 Obtaining and Installing the Java SDK.....	5
4.3 Creating an Instance.....	6
4.4 Adding an Image.....	7
4.5 Searching for Images.....	8
4.6 Querying an Image.....	10
4.7 Deleting an Image.....	11
4.8 Modifying Image Information.....	12
4.9 Deleting an Instance.....	13
4.10 Querying Details About an Instance.....	14
<b>5 Using the Python SDK.....</b>	<b>16</b>
5.1 Preparing a Python Development Environment.....	16
5.2 Obtaining and Installing the Python SDK.....	16
5.3 Creating an Instance.....	17
5.4 Adding an Image.....	17
5.5 Searching for Images.....	18
5.6 Querying an Image.....	19
5.7 Deleting an Image.....	20
5.8 Modifying Image Information.....	21
5.9 Deleting an Instance.....	21
5.10 Querying Details About an Instance.....	22
<b>6 Using the .NET SDK.....</b>	<b>24</b>
6.1 Preparing a .NET Development Environment.....	24
6.2 Obtaining and Installing the .Net SDK.....	24
6.3 Creating an Instance.....	24
6.4 Adding an Image.....	26
6.5 Searching for Images.....	27

---

6.6 Querying an Image.....	28
6.7 Deleting an Image.....	29
6.8 Modifying Image Information.....	31
6.9 Deleting an Instance.....	32
6.10 Querying Details About an Instance.....	33
<b>7 Using the Go SDK.....</b>	<b>35</b>
7.1 Preparing a Go Development Environment.....	35
7.2 Obtaining and Installing the Go SDK.....	35
7.3 Creating an Instance.....	35
7.4 Adding an Image.....	36
7.5 Searching for Images.....	37
7.6 Querying an Image.....	38
7.7 Deleting an Image.....	39
7.8 Modifying Image Information.....	40
7.9 Deleting an Instance.....	41
7.10 Querying Details About an Instance.....	41

# 1 Introduction to ImageSearch SDK

## Overview of ImageSearch

ImageSearch leverages cutting-edge deep learning and image recognition technologies to deliver service- and industry-specific feature vectorization and search capabilities that help you search images from a specified gallery using the exact or fuzzy match option.

ImageSearch is accessible through open application programming interfaces (APIs). You can obtain image search results in real time by calling an API.

## SDK Overview

The ImageSearch software development kit (SDK) encapsulates the RESTful APIs provided by ImageSearch to simplify application development. You can directly call API functions provided by the ImageSearch SDK to obtain the ImageSearch service capabilities.

### NOTE

ImageSearch SDK does not support access retry. If an access error occurs, the SDK directly returns the request result. Therefore, you need to establish a retry mechanism on the client.

## Mappings Between Services and APIs

[Table 1-1](#) lists the mappings between the ImageSearch services and APIs.

**Table 1-1** Mappings between services and APIs

Service	API
Creating an instance	POST /v1/{projectId}/service
Adding an image	POST /v1/{projectId}/{instanceName}/image
Searching for images	POST /v1/{projectId}/{instanceName}/image/search

Service	API
Querying an image	POST /v1/{projectId}/{instanceName}/image/check
Deleting an image	DELETE /v1/{projectId}/{instanceName}/image
Modifying image information	PUT /v1/{projectId}/{instanceName}/image
Deleting an instance	DELETE /v1/{projectId}/service/{instanceName}
Querying details about an instance	GET /v1/{projectId}/service/{instanceName}

# 2 Obtaining Authentication Information

---

Authentication is required to use service APIs. You are advised to use AK/SK authentication. The procedure is as follows:

- Step 1** Register an account and log in to the HUAWEI CLOUD management console.
- Step 2** Hover the cursor over the username and select **My Credential** from the drop-down list box.
- Step 3** Choose **Access Keys** and click **Create Access Key**.
- Step 4** Enter the verification code sent to your mail or mobile phone.
- Step 5** Click **OK** to download the AK/SK of your account. The AK/SK is saved in a local file. Keep the file secure.

----End

# 3 Preparing the Environment

**Table 3-1** lists the environments you need to prepare before using ImageSearch SDKs that use different programming languages.

**Table 3-1** Development environment

Programming Language	Item	Description
JAVA	Installing JDK	An environment used for developing Java applications. Java JDK 1.8 and later versions are supported. You are advised to use the Java SDK by installing dependencies through Apache Maven.
PYTHON	Installing Python	Python 3.3 and later versions are supported.
.NET	Installing .NET	.NET Standard 2.0 and later versions, and C# 4.0 and later versions are supported.
GO	Installing Go	Go 1.14 and later versions are supported.



# 4 Using the SDK (Java)

## 4.1 Preparing a Java Development Environment

Image Search Java SDK uses Java SE Development Kit 8 (JDK 8) or later. The following uses JDK 8 (Windows x64) running on Windows 7 as an example. If you have downloaded the JDK and configured the environment, skip this section.

- Step 1** Download the [JDK file](#).
- Step 2** After the JDK file is downloaded, install the JDK as prompted. For example, install the JDK to the `C:\Program Files\Java\jdk1.8.0_131` directory on the local PC.
- Step 3** Right-click **Computer**, choose **Properties > Advanced System Settings > Environment Variables**, and perform the following operations to configure Java environment variables:
- Create system variable **JAVA\_HOME** whose value is the JDK installation path.
  - Add `%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin` to **Path**. Separate multiple values with semicolons (;).
  - Create system variable **CLASSPATH** whose value is `%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar`.
- Step 4** Open the command line interface (CLI) and run `java -version`. If the information shown in [Figure 4-1](#) is displayed, the configuration is successful.

**Figure 4-1** Java version information

```
C:\>java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

----End

## 4.2 Obtaining and Installing the Java SDK

You are advised to use the Java SDK for ImageSearch by installing dependencies through Apache Maven.

- Step 1** **Download** and **install** Apache Maven on your operating system.
- Step 2** After installing and configuring Maven, run the **mvn -v** command. Maven is installed successfully if the following information is displayed.

```
D:\>mvn -v
D:\
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: D:\maven\apache-maven-3.3.9\bin\..
Java version: 1.8.0_262, vendor: Huawei Technologies Co., Ltd
Java home: D:\develop\jdk_1.8\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 10", version: "10.0", arch: "amd64", family: "dos"
```

- Step 3** Add dependencies to an existing Maven project or use an integrated development environment (IDE) to create a Maven project.

For example, to create a Maven project using IntelliJ IDEA, perform the following steps (skip them if you already have a Maven project):

1. Start IntelliJ IDEA.
2. Choose **File > New > project...**
3. In the displayed **New Project** dialog box, click **Maven** and then **Next**.
4. Specify **GroupId** and **ArtifactId**, and click **Next**.
5. Specify **Project name** and **Project location**, and click **Finish**.

- Step 4** Add dependency items to the **pom.xml** file of the Maven project.

The following are dependencies for ImageSearch SDK 3.0.55:

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-imagesearch</artifactId>
  <version>3.0.67</version>
</dependency>
```

The following sample code uses the fastjson dependency (ignore and delete it if it is not required):

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.70</version>
</dependency>
```

----End

## 4.3 Creating an Instance

This section describes how to create an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the **setName** and **setModel** methods of the **CreateInstanceReq** class to configure the instance name and model. After the configuration is complete, run the code.

1. The sample code for creating an instance is as follows:

```
import com.alibaba.fastjson.JSON;
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.imagesearch.v1.ImageSearchClient;
import com.huaweicloud.sdk.imagesearch.v1.model.RunCreateInstanceRequest;
```

```

import com.huaweicloud.sdk.imagesearch.v1.model.CreateInstanceReq;
import com.huaweicloud.sdk.imagesearch.v1.model.RunCreateInstanceResponse;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;
import java.util.Arrays;
import java.util.List;

/**
 * Create an instance. An index library is created in the instance to store image features.
 */
public class RunCreateInstanceSolution {
    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";
        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
        RunCreateInstanceRequest request = new RunCreateInstanceRequest();
        CreateInstanceReq body = new CreateInstanceReq();
        //Set the instance name.
        body.setName("instance-name");
        //Set the model.
        body.setModel("common-search");
        List<String> tags = Arrays.asList("animal", "plant");
        body.setTags(tags);
        request.setBody(body);
        try {
            RunCreateInstanceResponse response = client.runCreateInstance(request);
            System.out.println(JSON.toJSONString(response));
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}

```

2. Execute the sample code. If **200** is displayed on the console, the code is successfully executed. For details, see [Creating an Instance](#).

```

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"desc":"","domain":"general image search","expiredDate":-1,"httpStatusCode":
200,"instanceName":"instance-name","level":30000000,"registerDate":1636700045229,"status":
{"value":"NORMAL"},"tags":["animal","plant"]}
Process finished with exit code 0

```

## 4.4 Adding an Image

This section describes how to add an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the **setPath** method of the **AddPictureRequestReq** class to specify the URL of the image you want to add. After the configuration is complete, run the code.

1. The sample code for adding an image is as follows:

```
import com.alibaba.fastjson.JSON;
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.imagesearch.v1.*;
import com.huaweicloud.sdk.imagesearch.v1.model.*;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

/**
 * Add an image.
 */
public class RunAddPictureSolution {

    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);
        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
        RunAddPictureRequest request = new RunAddPictureRequest();
        request.setInstanceName("instance-name");
        AddPictureRequestReq req = new AddPictureRequestReq();
        //Add an image. Use setPath to configure the image URL.
        req.setPath("https://bucketname.obs.cn-north-4.myhuaweicloud.com/image/test1.jpg");

        request.setBody(req);
        try {
            RunAddPictureResponse response = client.runAddPicture(request);
            System.out.println(JSON.toJSONString(response));
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. Execute the sample code. If the image adding result is displayed on the console, the code is successfully executed. For details, see [Adding an image](#).

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"httpStatusCode":200,"result":"Success."}
```

```
Process finished with exit code 0
```

## 4.5 Searching for Images

This section describes how to use an image to search for similar images using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the **setPath** method of the **SearchPictureReq** class to specify the URL of the image you want to use to search for similar images. After the configuration is complete, run the code.

1. The sample code for image searching is as follows:

```
import com.alibaba.fastjson.JSON;
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.imagesearch.v1.*;
import com.huaweicloud.sdk.imagesearch.v1.model.*;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

/**
 * Search for similar images from the image library.
 */
public class RunSearchPictureSolution {

    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";

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

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
        RunSearchPictureRequest request = new RunSearchPictureRequest();
        //Set the instance name.
        request.setInstanceName("instance-name");
        SearchPictureReq body = new SearchPictureReq();
        //Set the image URL.
        body.setPath("https://bucketname.obs.cn-north-4.myhuaweicloud.com/search-test.jpg");
        request.setBody(body);
        try {
            RunSearchPictureResponse response = client.runSearchPicture(request);
            System.out.println(JSON.toJSONString(response));
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. Execute the sample code. If the image search result is displayed on the console, the code is successfully executed. For details, see [Searching for Images](#).

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"count":1,"httpStatusCode":200,"result":[{"path":"https://bucketName.obs.cn-north-4.myhuaweicloud.com/search-test.jpg","sim":1.0,"tags":{}}]}
Process finished with exit code 0
```

## 4.6 Querying an Image

This section describes how to query an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the `setPath` method of the `CheckPictureReq` class to specify the URL of the image you want to query. After the configuration is complete, run the code.

1. The sample code for querying an image is as follows:

```
import com.alibaba.fastjson.JSON;

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.imagesearch.v1.*;
import com.huaweicloud.sdk.imagesearch.v1.model.*;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

/**
 * Check whether the image exists in the index library by searching for it in the specified path.
 */
public class RunCheckPictureSolution {

    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";

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

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
        RunCheckPictureRequest request = new RunCheckPictureRequest();
        //Set the instance name.
        request.setInstanceName("instance-name");
        DeletePictureReq body = new DeletePictureReq();
        //Image URL, which is used as the ID of an image in the image library.
        body.setPath("https://bucketname.obs.cn-north4.myhuaweicloud.com/searchtest.jpg");
        request.setBody(body);

        try {
            RunCheckPictureResponse response = client.runCheckPicture(request);
            System.out.println(JSON.toJSONString(response));
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. Execute the sample code. If the image query result is displayed on the console, the code is successfully executed. For details, see [Querying an Image](#).

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"exist":"true","statusCode":200}
```

```
Process finished with exit code 0
```

## 4.7 Deleting an Image

This section describes how to delete an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the `setPath` method of the `DeletePictureReq` class to specify the URL of the image you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an image is as follows:

```
import com.alibaba.fastjson.JSON;

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.imagesearch.v1.*;
import com.huaweicloud.sdk.imagesearch.v1.model.*;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

/**
 * Delete an image from the index library by specifying the image path.
 */
public class RunDeletePictureSolution {

    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";

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

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
        RunDeletePictureRequest request = new RunDeletePictureRequest();
        //Set the instance name.
        request.setInstanceName("instance-name");
        DeletePictureReq body= new DeletePictureReq();
        //Image URL, which is used as the ID of an image in the image library.
        body.setPath("https://bucketname.obs.cn-north-4.myhuaweicloud.com/search-test.jpg");
        request.setBody(body);
        try {
            RunDeletePictureResponse response = client.runDeletePicture(request);
            System.out.println(JSON.toJSONString(response));
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```

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

```

2. Execute the sample code. If the image deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Image](#).

```

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"httpStatusCode":200,"result":"Success."}

```

```
Process finished with exit code 0
```

## 4.8 Modifying Image Information

This section describes how to modify an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the **setPath** or **setTags** methods of the **RunModifyPictureReq** class to change the path or tags of an image. After the configuration is complete, run the code.

1. The sample code for modifying image information is as follows:

```

import com.alibaba.fastjson.JSON;

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.imagesearch.v1.*;
import com.huaweicloud.sdk.imagesearch.v1.model.*;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

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

/**
 * Modify the information about an existing image in the image library.
 */
public class RunModifyPictureSolution {

    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";

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

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
        RunModifyPictureRequest request = new RunModifyPictureRequest();
        //Set the instance name.
        request.setInstanceName("instance-name");
        RunModifyPictureReq body = new RunModifyPictureReq();
        //Image URL, which is used as the ID of an image in the image library.
        body.setPath("https://bucketname.obs.cn-north-4.myhuaweicloud.com/search-test.jpg");
        //Customize tags in the form of a key-value pair.
        Map<String, String> tags = new HashMap<>();
        tags.put("animal", "dog");
        body.setTags(tags);
    }
}

```



```

request.setBody(body);

try {
    RunModifyPictureResponse response = client.runModifyPicture(request);
    System.out.println(JSON.toJSONString(response));
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

2. Execute the sample code. If the image information modification result is displayed on the console, the code is successfully executed. For details, see [Modifying Image Information](#).

```

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"httpStatusCode":200,"result":"Success."}

```

Process finished with exit code 0

## 4.9 Deleting an Instance

This section describes how to delete an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the `setInstanceName` method of the `RunDeleteInstanceRequest` class to specify the name of the instance you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an instance is as follows:

```

import com.alibaba.fastjson.JSON;
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.imagesearch.v1.ImageSearchClient;
import com.huaweicloud.sdk.imagesearch.v1.model.RunDeleteInstanceRequest;
import com.huaweicloud.sdk.imagesearch.v1.model.RunDeleteInstanceResponse;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

/**
 * Delete an existing instance.
 */
public class RunDeleteInstanceDemo {
    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";
        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.
            .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
            .build();
    }
}

```

```

RunDeleteInstanceRequest request = new RunDeleteInstanceRequest();
//Set the instance name.
request.setInstanceName("instance-name");
try {
    RunDeleteInstanceResponse response = client.runDeleteInstance(request);
    System.out.println(JSON.toJSONString(response));
} catch (ConnectionException e) {
    System.out.println(e.getMessage());
} catch (RequestTimeoutException e) {
    System.out.println(e.getMessage());
} catch (ServiceResponseException e) {
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

2. Execute the sample code. If the instance deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Instance](#).

```

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"httpStatusCode":200,"result":"success."}

```

Process finished with exit code 0

## 4.10 Querying Details About an Instance

This section describes how to query an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the **setInstanceName** method of the **RunQueryInstanceRequest** class to specify the name of the instance you want to query. After the configuration is complete, run the code.

1. The sample code for querying an instance is as follows:

```

import com.alibaba.fastjson.JSON;

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.imagesearch.v1.*;
import com.huaweicloud.sdk.imagesearch.v1.model.*;
import com.huaweicloud.sdk.imagesearch.v1.region.ImageSearchRegion;

/**
 * Display details about an instance.
 */
public class RunQueryInstanceSolution {

    public static void main(String[] args) {
        //Set the obtained AK and SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";

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

        ImageSearchClient client = ImageSearchClient.newBuilder()
            .withCredential(auth)
            //Set the region. In this example, CN North-Beijing4 is used.

```

```
        .withRegion(ImageSearchRegion.valueOf("cn-north-4"))
        .build();
RunQueryInstanceRequest request = new RunQueryInstanceRequest();
//Set the instance name.
request.setInstanceName("instance-name");
try {
    RunQueryInstanceResponse response = client.runQueryInstance(request);
    System.out.println(JSON.toJSONString(response));
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

2. Execute the sample code. If the instance querying result is displayed on the console, the code is successfully executed. For details, see [Querying the Details About an Instance](#).

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"desc":"","domain":"general image search","expiredDate":-1,"httpStatusCode":
200,"instanceName":"instance-name","level":30000000,"registerDate":1637230766199,"status":
{"value":"NORMAL"},"tags":["animal","plant"]}
Process finished with exit code 0
```

# 5 Using the Python SDK

## 5.1 Preparing a Python Development Environment

To use the Python SDK of ImageSearch, you need to configure the Python development environment. The procedure is as follows:

- Step 1** Download Python of a proper version from [Python's official website](#) and install it. Python3.3 or later is recommended. This section uses Python 3.7 as an example.
- Step 2** Download the latest version from the [PyCharm official website](#).
- Step 3** Start the PyCharm development tool and choose **File > Settings > Project Interpreter** to configure the Python environment.
- Step 4** Select the Python installation path in the upper part of the page. After selecting the target Python, click **Apply** at the bottom of the page to complete the configuration.

----End

## 5.2 Obtaining and Installing the Python SDK

To obtain and install the Python SDK, you are advised to use pip commands or PyCharm. You need to install the **huaweicloudsdkcore** and **huaweicloudsdkimagesearch** packages as follows:

Install the Python SDK using pip commands:

```
# Install the huaweicloudsdkcore library.  
pip install huaweicloudsdkcore
```

```
# Install the huaweicloudsdkimagesearch library.  
pip install huaweicloudsdkimagesearch
```

To install the Python SDK using PyCharm, perform the following steps:

1. Start PyCharm and choose **File > Settings...**
2. Click **Python Interpreter** and then **+**.
3. Click **+**, search for **huaweicloudsdkcore** and **huaweicloudsdkimagesearch**, and click **Install Package** in the lower left corner.

## 5.3 Creating an Instance

This section describes how to create an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **name** and **model** parameters in the initialized **CreateInstanceReq** command to configure the instance name and model. After the configuration is complete, run the code.

1. The sample code for creating an instance is as follows:

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    # Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = RunCreateInstanceRequest()
        # Set parameters to create an instance.
        request.body = CreateInstanceReq(
            name="instance-name", # Name
            model="common-search", # Model
            tags=["animal", "plant"] # Tag
        )

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

2. Execute the sample code. If **200** is displayed on the console, the code is successfully executed. For details, see [Creating an Instance](#).

```
200
{"domain": "general image search", "desc": "", "registerDate": 1637548249353, "expiredDate": -1,
"level": 30000000, "tags": ["animal", "plant"], "status": "NORMAL", "instanceName": "instance-name"}
Process finished with exit code 0
```

## 5.4 Adding an Image

This section describes how to add an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Use the **setPath** method of the initialized **AddPictureRequestReq** command to specify the URL of the image you want to add. After the configuration is complete, run the code.

1. The sample code for adding an image is as follows:

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    # Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)
    client = ImageSearchClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageSearchRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = RunAddPictureRequest()
        # Set the instance name.
        request.instance_name="instance-name"
        # Set the image path.
        request.body = AddPictureRequestReq(
            path="https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch-test.jpg"
        )
        response = client.run_add_picture(request)
        print(response.status_code)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

2. Execute the sample code. If the image adding result is displayed on the console, the code is successfully executed. For details, see [Adding an image](#).

```
200
{"result": "Success."}
```

## 5.5 Searching for Images

This section describes how to use an image to search for similar images using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **path** parameter in the initialized **SearchPictureReq** command to specify the URL of the image you want to use to search for similar images. After the configuration is complete, run the code.

1. The sample code for image searching is as follows:

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    # Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)
```

```

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

try:
    request = RunSearchPictureRequest()
    # Set the instance name.
    request.instance_name="instance-name"
    # Set the image path.
    request.body=SearchPictureReq(
        path="https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch-test.jpg"
    )
    response = client.run_search_picture(request)
    print(response.status_code)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

2. Execute the sample code. If the image search result is displayed on the console, the code is successfully executed. For details, see [Searching for Images](#).

```

200
{"count": 1, "result": [{"path": "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch-test.jpg", "sim": 1.0, "tags": {}}]}

```

## 5.6 Querying an Image

This section describes how to query an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **path** parameter in the initialized **DeletePictureReq** command to specify the URL of the image you want to query. After the configuration is complete, run the code.

1. The sample code for querying an image is as follows:

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    # Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = RunCheckPictureRequest()
        # Set the instance name.
        request.instance_name="instance-name"
        # Set the image path.
        request.body=DeletePictureReq(
            path="https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch-test.jpg"
        )
    
```

```

response = client.run_check_picture(request)
print(response.status_code)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

2. Execute the sample code. If the image query result is displayed on the console, the code is successfully executed. For details, see [Querying an Image](#).

```

200
{
  "exist": "true"
}

```

## 5.7 Deleting an Image

This section describes how to delete an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set either the **url** or **image** parameter in the initialized **CelebrityRecognitionReq** to specify the information of the image you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an image is as follows:

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)
    client = ImageSearchClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageSearchRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = RunDeletePictureRequest()
        # Set the instance name.
        request.instance_name="instance-name"
        # Set the image path.
        request.body = DeletePictureReq(
            path="https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch-test.jpg"
        )
        response = client.run_delete_picture(request)
        print(response.status_code)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

2. Execute the sample code. If the image deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Image](#).

```

200
{

```



```
"result": "success."
}
```

## 5.8 Modifying Image Information

This section describes how to modify an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **path** and **tags** parameters in the initialized **RunModifyPictureReq** command to change the path and tags of an image. After the configuration is complete, run the code.

1. The sample code for modifying image information is as follows:

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    # Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = RunModifyPictureRequest()
        # Set the instance name.
        request.instance_name="instance-name"
        # Set the image URL and tags.
        request.body=RunModifyPictureReq(
            path="https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch-test.jpg",
            tags={"animal": "dog"}
        )
        response = client.run_modify_picture(request)
        print(response.status_code)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

2. Execute the sample code. If the image information modification result is displayed on the console, the code is successfully executed. For details, see [Modifying Image Information](#).

```
200
{
  "result": "Success."
}
```

## 5.9 Deleting an Instance

This section describes how to delete an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **instance\_name** parameter in the **RunDeleteInstanceRequest** command to

specify the name of the instance you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an instance is as follows:

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    # Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = RunDeleteInstanceRequest()
        #Set the instance name.
        request.instance_name = "instance-name"
        response = client.run_delete_instance(request)
        print(response.status_code)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

2. Execute the sample code. If the instance deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Instance](#).

```
200
{
  "result": "Success."
}
```

## 5.10 Querying Details About an Instance

This section describes how to query an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **instance\_name** parameter in the **RunQueryInstanceRequest** command to specify the name of the instance you want to query. After the configuration is complete, run the code.

1. The sample code for querying an instance is as follows:

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimagesearch.v1 import *
from huaweicloudsdkimagesearch.v1.region.imagesearch_region import ImageSearchRegion

if __name__ == "__main__":
    #Set the obtained AK and SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"
```

```
credentials = BasicCredentials(ak, sk)

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

try:
    request = RunQueryInstanceRequest()
    #Set the instance name.
    request.instance_name = "instance-name"
    response = client.run_query_instance(request)
    print(response.status_code)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

2. Execute the sample code. If the instance querying result is displayed on the console, the code is successfully executed. For details, see [Querying the Details About an Instance](#).

```
200
{"domain": "general image search", "desc": "", "registerDate": 1637548249353, "expiredDate": -1,
"level": 30000000, "tags": ["animal", "plant"], "status": "NORMAL", "instanceName": "instance-name"}
```

# 6 Using the .NET SDK

## 6.1 Preparing a .NET Development Environment

To use the .NET SDK of ImageSearch, you need to configure the .NET development environment. The procedure is as follows:

1. Download the latest Visual Studio from the [Visual Studio official website](#).
2. Click the .exe file to start installation.
3. Choose **Workloads > ASP.NET and Web Development**. After the installation is complete, you can run it.

## 6.2 Obtaining and Installing the .Net SDK

You can install the .Net SDK using commands or Visual Studio.

```
Use .NET CLI to install the .Net SDK.  
dotnet add package HuaweiCloud.SDK.Core  
dotnet add package HuaweiCloud.SDK.ImageSearch
```

```
Use Package Manager to install the .Net SDK.  
Install-Package HuaweiCloud.SDK.Core  
Install-Package HuaweiCloud.SDK.ImageSearch
```

If you use Visual Studio, choose **Tools > NuGet Package Manager > Manage NuGet Packages for Solution...**, and search for and install **HuaweiCloud.SDK.Core** and **HuaweiCloud.SDK.ImageSearch**.

## 6.3 Creating an Instance

This section describes how to create an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Name** and **Model** parameters in the initialized **CreateInstanceReq** command to configure the instance name and model. After the configuration is complete, run the code.

1. The sample code for creating an instance is as follows:  

```
using System;  
using System.Collections.Generic;
```

```

using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.ImageSearch;
using HuaweiCloud.SDK.ImageSearch.V1;
using HuaweiCloud.SDK.ImageSearch.V1.Model;
using Newtonsoft.Json;

namespace RunCreateInstanceSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            const string ak = "<YOUR AK>";
            const string sk = "<YOUR SK>";

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

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

            var req = new RunCreateInstanceRequest
            {
            };
            req.Body = new CreateInstanceReq()
            {
                Name = "instance-name",
                Model = "common-search",
                Tags = new List<string>{"animal", "plant"}
            };

            try
            {
                var resp = client.RunCreateInstance(req);
                var respStatusCode = resp.HttpStatusCode;
                Console.WriteLine(respStatusCode);
                Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
            }
            catch (RequestTimeoutException requestTimeoutException)
            {
                Console.WriteLine(requestTimeoutException.ErrorMessage);
            }
            catch (ServiceResponseException clientRequestException)
            {
                Console.WriteLine(clientRequestException.HttpStatusCode);
                Console.WriteLine(clientRequestException.ErrorCode);
                Console.WriteLine(clientRequestException.ErrorMessage);
            }
            catch (ConnectionException connectionException)
            {
                Console.WriteLine(connectionException.ErrorMessage);
            }
        }
    }
}

```

- Execute the sample code. If **200** is displayed on the console, the code is successfully executed. For details, see [Creating an Instance](#).

```

200
{
  "instanceName": "instance-name",
  "level": 30000000,
  "expiredDate": -1,
  "domain": "general image search",
  "desc": ""
}

```

```
"registerDate": 1638939677331,  
"tags" [  
  "animal",  
  "plant"  
],  
"status": "NORMAL"  
}
```

## 6.4 Adding an Image

This section describes how to add an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the **AddPictureRequestReq** class to specify the URL of the image you want to add. After the configuration is complete, run the code.

1. The sample code for adding an image is as follows:

```
using System;  
using System.Collections.Generic;  
using HuaweiCloud.SDK.Core;  
using HuaweiCloud.SDK.Core.Auth;  
using HuaweiCloud.SDK.ImageSearch;  
using HuaweiCloud.SDK.ImageSearch.V1;  
using HuaweiCloud.SDK.ImageSearch.V1.Model;  
using Newtonsoft.Json;  
  
namespace RunCreateInstanceSolution  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            const string ak = "<YOUR AK>";  
            const string sk = "<YOUR SK>";  
  
            var config = HttpConfig.GetDefaultConfig();  
            config.IgnoreSslVerification = true;  
            var auth = new BasicCredentials(ak, sk);  
  
            var client = ImageSearchClient.NewBuilder()  
                .WithCredential(auth)  
                .WithRegion(ImageSearchRegion.ValueOf("cn-north-4"))  
                .WithHttpConfig(config)  
                .Build();  
  
            var req = new RunAddPictureRequest  
            {  
                InstanceName = "instance-name"  
            };  
            req.Body = new AddPictureRequestReq()  
            {  
                Path = "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"  
            };  
  
            try  
            {  
                var resp = client.RunAddPicture(req);  
                var respStatusCode = resp.HttpStatusCode;  
                Console.WriteLine(respStatusCode);  
                Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));  
            }  
            catch (RequestTimeoutException requestTimeoutException)  
            {  
                Console.WriteLine(requestTimeoutException.ErrorMessage);  
            }  
            catch (ServiceResponseException clientRequestException)  
            {
```

```
        Console.WriteLine(clientRequestException.HttpStatusCode);
        Console.WriteLine(clientRequestException.ErrorCode);
        Console.WriteLine(clientRequestException.ErrorMessage);
    }
    catch (ConnectionException connectionException)
    {
        Console.WriteLine(connectionException.ErrorMessage);
    }
}
}
```

2. Execute the sample code. If the image adding result is displayed on the console, the code is successfully executed. For details, see [Adding an image](#).

```
200
{
  "result": "Success."
}
```

## 6.5 Searching for Images

This section describes how to use an image to search for similar images using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the **SearchPictureReq** class to specify the URL of the image you want to use to search for similar images. After the configuration is complete, run the code.

1. The sample code for image searching is as follows:

```
using System;
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.ImageSearch;
using HuaweiCloud.SDK.ImageSearch.V1;
using HuaweiCloud.SDK.ImageSearch.V1.Model;
using Newtonsoft.Json;

namespace RunSearchPictureSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            const string ak = "<YOUR AK>";
            const string sk = "<YOUR SK>";

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

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

            var req = new RunSearchPictureRequest
            {
                InstanceName = "instance-name"
            };
            req.Body = new SearchPictureReq()
            {
                Path = "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"
            }
        }
    }
}
```

```

};

try
{
    var resp = client.RunSearchPicture(req);
    var respStatusCode = resp.HttpStatusCode;
    Console.WriteLine(respStatusCode);
    Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
}
catch (RequestTimeoutException requestTimeoutException)
{
    Console.WriteLine(requestTimeoutException.ErrorMessage);
}
catch (ServiceResponseException clientRequestException)
{
    Console.WriteLine(clientRequestException.HttpStatusCode);
    Console.WriteLine(clientRequestException.ErrorCode);
    Console.WriteLine(clientRequestException.ErrorMessage);
}
catch (ConnectionException connectionException)
{
    Console.WriteLine(connectionException.ErrorMessage);
}
}
}

```

- Execute the sample code. If the image search result is displayed on the console, the code is successfully executed. For details, see [Searching for Images](#).

```

200
{
  "result": [
    {
      "path": "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg",
      "sim": 1.0,
      "tags": {}
    }
  ],
  "count": 1
}

```

## 6.6 Querying an Image

This section describes how to query an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the **DeletePictureReq** class to specify the URL of the image you want to query. After the configuration is complete, run the code.

- The sample code for querying an image is as follows:

```

using System;
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.ImageSearch;
using HuaweiCloud.SDK.ImageSearch.V1;
using HuaweiCloud.SDK.ImageSearch.V1.Model;
using Newtonsoft.Json;

namespace RunCheckPictureSolution
{
    class Program
    {
        static void Main(string[] args)

```



```
{
    const string ak = "<YOUR AK>";
    const string sk = "<YOUR SK>";

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

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

    var req = new RunCheckPictureRequest
    {
        InstanceName = "instance-name"
    };
    req.Body = new DeletePictureReq()
    {
        Path = "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"
    };

    try
    {
        var resp = client.RunCheckPicture(req);
        var respStatusCode = resp.HttpStatusCode;
        Console.WriteLine(respStatusCode);
        Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
    }
    catch (RequestTimeoutException requestTimeoutException)
    {
        Console.WriteLine(requestTimeoutException.ErrorMessage);
    }
    catch (ServiceResponseException clientRequestException)
    {
        Console.WriteLine(clientRequestException.HttpStatusCode);
        Console.WriteLine(clientRequestException.ErrorCode);
        Console.WriteLine(clientRequestException.ErrorMsg);
    }
    catch (ConnectionException connectionException)
    {
        Console.WriteLine(connectionException.ErrorMessage);
    }
}
}
```

2. Execute the sample code. If the image query result is displayed on the console, the code is successfully executed. For details, see [Querying an Image](#).

```
200
{
  "exist": "true"
}
```

## 6.7 Deleting an Image

This section describes how to delete an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the **DeletePictureReq** class to specify the URL of the image you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an image is as follows:

```

using System;
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.ImageSearch;
using HuaweiCloud.SDK.ImageSearch.V1;
using HuaweiCloud.SDK.ImageSearch.V1.Model;
using Newtonsoft.Json;

namespace RunDeletePictureSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            const string ak = "<YOUR AK>";
            const string sk = "<YOUR SK>";

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

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

            var req = new RunDeletePictureRequest
            {
                InstanceName = "instance-name"
            };
            req.Body = new DeletePictureReq()
            {
                Path = "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"
            };

            try
            {
                var resp = client.RunDeletePicture(req);
                var respStatusCode = resp.HttpStatusCode;
                Console.WriteLine(respStatusCode);
                Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
            }
            catch (RequestTimeoutException requestTimeoutException)
            {
                Console.WriteLine(requestTimeoutException.ErrorMessage);
            }
            catch (ServiceResponseException clientRequestException)
            {
                Console.WriteLine(clientRequestException.HttpStatusCode);
                Console.WriteLine(clientRequestException.ErrorCode);
                Console.WriteLine(clientRequestException.ErrorMsg);
            }
            catch (ConnectionException connectionException)
            {
                Console.WriteLine(connectionException.ErrorMessage);
            }
        }
    }
}

```

2. Execute the sample code. If the image deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Image](#).

```

200
{
  "result": "Success."
}

```

## 6.8 Modifying Image Information

This section describes how to modify an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** and **Tags** parameters in the initialized **RunModifyPictureReq** command to change the path and tags of an image. After the configuration is complete, run the code.

1. The sample code for modifying image information is as follows:

```
using System;
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.ImageSearch;
using HuaweiCloud.SDK.ImageSearch.V1;
using HuaweiCloud.SDK.ImageSearch.V1.Model;
using Newtonsoft.Json;

namespace RunModifyPictureSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            const string ak = "<YOUR AK>";
            const string sk = "<YOUR SK>";

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

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

            var req = new RunModifyPictureRequest
            {
                InstanceName = "instance-name"
            };
            req.Body = new RunModifyPictureReq()
            {
                Path = "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg",
                Tags = new Dictionary<string, string>{{"animal", "dog"}}
            };

            try
            {
                var resp = client.RunModifyPicture(req);
                var respStatusCode = resp.HttpStatusCode;
                Console.WriteLine(respStatusCode);
                Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
            }
            catch (RequestTimeoutException requestTimeoutException)
            {
                Console.WriteLine(requestTimeoutException.ErrorMessage);
            }
            catch (ServiceResponseException clientRequestException)
            {
                Console.WriteLine(clientRequestException.HttpStatusCode);
                Console.WriteLine(clientRequestException.ErrorCode);
                Console.WriteLine(clientRequestException.ErrorMsg);
            }
            catch (ConnectionException connectionException)
            {
            }
        }
    }
}
```

```
        {  
            Console.WriteLine(connectionException.ErrorMessage);  
        }  
    }  
}
```

2. Execute the sample code. If the instance deletion result is displayed on the console, the code is successfully executed. For details, see [Modifying Image Information](#).

```
200  
{  
  "result": "Success."  
}
```

## 6.9 Deleting an Instance

This section describes how to delete an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **InstanceName** parameter in the **RunDeleteInstanceRequest** class to specify the name of the instance you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an instance is as follows:

```
using System;  
using System;  
using System.Collections.Generic;  
using HuaweiCloud.SDK.Core;  
using HuaweiCloud.SDK.Core.Auth;  
using HuaweiCloud.SDK.ImageSearch;  
using HuaweiCloud.SDK.ImageSearch.V1;  
using HuaweiCloud.SDK.ImageSearch.V1.Model;  
using Newtonsoft.Json;  
  
namespace RunDeleteInstanceSolution  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            const string ak = "<YOUR AK>";  
            const string sk = "<YOUR SK>";  
  
            var config = HttpConfig.GetDefaultConfig();  
            config.IgnoreSslVerification = true;  
            var auth = new BasicCredentials(ak, sk);  
  
            var client = ImageSearchClient.NewBuilder()  
                .WithCredential(auth)  
                .WithRegion(ImageSearchRegion.ValueOf("cn-north-4"))  
                .WithHttpConfig(config)  
                .Build();  
            //Set the instance name.  
            var req = new RunDeleteInstanceRequest  
            {  
                InstanceName = "instance-name"  
            };  
  
            try  
            {  
                var resp = client.RunDeleteInstance(req);  
                var respStatusCode = resp.HttpStatusCode;  
                Console.WriteLine(respStatusCode);  
                Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));  
            }  
        }  
    }  
}
```

```
        catch (RequestTimeoutException requestTimeoutException)
        {
            Console.WriteLine(requestTimeoutException.ErrorMessage);
        }
        catch (ServiceResponseException clientRequestException)
        {
            Console.WriteLine(clientRequestException.HttpStatusCode);
            Console.WriteLine(clientRequestException.ErrorCode);
            Console.WriteLine(clientRequestException.ErrorMessage);
        }
        catch (ConnectionException connectionException)
        {
            Console.WriteLine(connectionException.ErrorMessage);
        }
    }
}
```

2. Execute the sample code. If the instance deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Instance](#).

```
200
{
  "result": "Success."
}
```

## 6.10 Querying Details About an Instance

This section describes how to query an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **InstanceName** parameter in the **RunQueryInstanceRequest** class to specify the name of the instance you want to query. After the configuration is complete, run the code.

1. The sample code for querying an instance is as follows:

```
using System;
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.ImageSearch;
using HuaweiCloud.SDK.ImageSearch.V1;
using HuaweiCloud.SDK.ImageSearch.V1.Model;
using Newtonsoft.Json;

namespace RunQueryInstanceSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            const string ak = "<YOUR AK>";
            const string sk = "<YOUR SK>";

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

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

            var req = new RunQueryInstanceRequest
```

```
{
    InstanceName = "instance-name"
};

try
{
    var resp = client.RunQueryInstance(req);
    var respStatusCode = resp.HttpStatusCode;
    Console.WriteLine(respStatusCode);
    Console.WriteLine(JsonConvert.SerializeObject(resp.HttpBody));
}
catch (RequestTimeoutException requestTimeoutException)
{
    Console.WriteLine(requestTimeoutException.ErrorMessage);
}
catch (ServiceResponseException clientRequestException)
{
    Console.WriteLine(clientRequestException.HttpStatusCode);
    Console.WriteLine(clientRequestException.ErrorCode);
    Console.WriteLine(clientRequestException.ErrorMessage);
}
catch (ConnectionException connectionException)
{
    Console.WriteLine(connectionException.ErrorMessage);
}
}
}
```

2. Execute the sample code. If the instance querying result is displayed on the console, the code is successfully executed. For details, see [Querying the Details About an Instance](#).

```
200
{
  "instanceName": "instance-name",
  "level": 30000000,
  "expiredDate": -1,
  "domain": "general image search",
  "domainEn": "General image search",
  "desc": "",
  "registerDate": 1638235929989,
  "tags": [
    "animal",
    "plant"
  ],
  "status": "NORMAL"
}
```

# 7 Using the Go SDK

---

## 7.1 Preparing a Go Development Environment

To use the Go SDK of ImageSearch, you need to configure the Go development environment. If the Go development environment has been installed, skip this section.

1. Download the Go SDK of the required version from the [golang official website](#) and install it. The SDK supports **go 1.14** version or later.
2. Install the downloaded Go in a specified path, for example, **D:\develop\go**.
3. Right-click **Computer**, choose **Properties > Advanced System Settings > Environment Variables**, and perform the following operations:
  - a. Create system variable **GOROOT** whose value is the Go installation path. Add environment variable **GOPATH**, which indicates the directory where the Go project is located.
  - b. Append **;%GOROOT%\bin** (path of the Go compiler) to the **PATH** variable.
  - c. Append **;%GOROOT%\lib** (path of the Go compiler) to the **CLASSPATH** variable.
4. Download the **vs code** of the operating system from the [vs code official website](#).
5. Install the **Go** and **Go Critic** plug-ins.

## 7.2 Obtaining and Installing the Go SDK

Run the following command:

```
go get -u github.com/huaweicloud/huaweicloud-sdk-go-v3
```

## 7.3 Creating an Instance

This section describes how to create an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Name** and **Model** parameters in the initialized **CreateInstanceReq** command to configure the instance name and model. After the configuration is complete, run the code.

1. The sample code for creating an instance is as follows:

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
    // Set the obtained AK and SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
    //Set the request client. In this example, CN North-Beijing4 is used.
    client := imagesearch.NewImageSearchClient(
        imagesearch.ImageSearchClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())
    //Set the instance name and model.
    request := &model.RunCreateInstanceRequest{}
    tags := []string{"animal", "plant"}
    request.Body = &model.CreateInstanceReq{
        Name: "instance-name",
        Model: "common-search",
        Tags: &tags,
    }
    response, err := client.RunCreateInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

2. If the instance creation result is displayed on the console, the code is successfully executed. For details, see [Creating an Instance](#).

```
RunCreateInstanceResponse{"domain": " general image search ","desc":"","registerDate":
1638321711363,"expiredDate":-1,"level":30000000,"status":"NORMAL","instanceName":"instance-
name"}
```

## 7.4 Adding an Image

This section describes how to add an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the initialized **AddPictureRequestReq** command to specify the URL of the image you want to add. After the configuration is complete, run the code.

1. The sample code for adding an image is as follows:

```
package main

import (
```



```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
// Set the obtained AK and SK.
ak := "<YOUR AK>"
sk := "<YOUR SK>"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()
//Set the request client. In this example, CN North-Beijing4 is used.
client := imagesearch.NewImageSearchClient(
    imagesearch.ImageSearchClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.RunAddPictureRequest{
    InstanceName: "instance-name",
}
//In the example, the image is added using URL.
request.Body = &model.AddPictureRequestReq{
    Path: "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"
}
response, err := client.RunAddPicture(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

2. Execute the sample code. If the image adding result is displayed on the console, the code is successfully executed. For details, see [Adding an image](#).  
RunAddPictureResponse {"result":"Success."}

## 7.5 Searching for Images

This section describes how to use an image to search for similar images using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the initialized **SearchPictureReq** command to specify the URL of the image you want to use to search for similar images. After the configuration is complete, run the code.

1. The sample code for image searching is as follows:

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
// Set the obtained AK and SK.
ak := "<YOUR AK>"

```

```

sk := "<YOUR SK>"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()
//Set the request client. In this example, CN North-Beijing4 is used.
client := imagesearch.NewImageSearchClient(
    imagesearch.ImagesearchClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.RunSearchPictureRequest{
    InstanceName: "instance-name",
}
//Use the image URL to search for similar images.
path := "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"
request.Body = &model.SearchPictureReq{
    Path: &path,
}
response, err := client.RunSearchPicture(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

2. Execute the sample code. If the image search result is displayed on the console, the code is successfully executed. For details, see [Searching for Images](#).

```

RunSearchPictureResponse {"count":1,"result":[{"path":"https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg","sim":1,"tags":{}}]}

```

## 7.6 Querying an Image

This section describes how to query an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the initialized **DeletePictureRequestReq** command to specify the URL of the image you want to query. After the configuration is complete, run the code.

1. The sample code for querying an image is as follows:

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
    // Set the obtained AK and SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
    // Set the request client. In this example, CN North-Beijing4 is used.
    client := imagesearch.NewImageSearchClient(

```

```

imagesearch.ImagesearchClientBuilder().
    WithRegion(region.ValueOf("cn-north-4")).
    WithCredential(auth).
    Build()
// Set the instance name.
request := &model.RunCheckPictureRequest{
    InstanceName: "instance-name",
}
//Set the image URL.
path := "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"
request.Body = &model.DeletePictureReq{
    Path: &path,
}
response, err := client.RunCheckPicture(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

2. Execute the sample code. If the image query result is displayed on the console, the code is successfully executed. For details, see [Querying an Image](#).

```
RunCheckPictureResponse {"exist":"true"}
```

## 7.7 Deleting an Image

This section describes how to delete an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** parameter in the initialized **DeletePictureReq** command to specify the URL of the image you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an image is as follows:

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
    // Set the obtained AK and SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

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

    // Set the request client. In this example, CN North-Beijing4 is used.
    client := imagesearch.NewImageSearchClient(
        imagesearch.ImagesearchClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build()

    // Set the instance name.
    request := &model.RunDeletePictureRequest{
        InstanceName: "instance-name",
    }

    // Set the image URL.
    path := "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg"

```

```

request.Body = &model.DeletePictureReq{
    Path: &path,
}
response, err := client.RunDeletePicture(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
}

```

- Execute the sample code. If the image deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Image](#).

```
RunDeletePictureResponse {"result":"Success."}
```

## 7.8 Modifying Image Information

This section describes how to modify an image using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **Path** and **Tags** parameters in the initialized **RunModifyPictureReq** command to change the path and tags of an image. After the configuration is complete, run the code.

- The sample code for modifying image information is as follows:

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
    // Set the obtained AK and SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
    // Set the request client. In this example, CN North-Beijing4 is used.
    client := imagesearch.NewImageSearchClient(
        imagesearch.ImageSearchClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.RunModifyPictureRequest{
        InstanceName: "instance-name",
    }
    // Set the URL and tags of the image.
    var tags interface{}
    tags = map[string]string{"animal": "dog"}
    request.Body = &model.RunModifyPictureReq{
        Path: "https://bucketname.obs.cn-north-4.myhuaweicloud.com/imagesearch/test.jpg",
        Tags: &tags,
    }
    response, err := client.RunModifyPicture(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

```
}
}
```

2. Execute the sample code. If the instance deletion result is displayed on the console, the code is successfully executed. For details, see [Modifying Image Information](#).

```
RunModifyPictureResponse {"result":"Success."}
```

## 7.9 Deleting an Instance

This section describes how to delete an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **InstanceName** parameter in the **RunDeleteInstanceRequest** command to specify the name of the instance you want to delete. After the configuration is complete, run the code.

1. The sample code for deleting an instance is as follows:

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
    // Set the obtained AK and SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
    // Set the request client. In this example, CN North-Beijing4 is used.
    client := imagesearch.NewImageSearchClient(
        imagesearch.ImageSearchClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())
    // Set the instance name.
    request := &model.RunDeleteInstanceRequest{
        InstanceName: "instance-name",
    }
    response, err := client.RunDeleteInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

2. Execute the sample code. If the instance deletion result is displayed on the console, the code is successfully executed. For details, see [Deleting an Instance](#).

```
RunDeleteInstanceResponse {"result":"success"}
```

## 7.10 Querying Details About an Instance

This section describes how to query an instance using AK/SK authentication.

Replace the AK/SK in the sample code with the actual AK/SK. Set the **InstanceName** parameter in the **RunQueryInstanceRequest** command to specify the name of the instance you want to query. After the configuration is complete, run the code.

1. The sample code for querying an instance is as follows:

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    imagesearch "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/imagesearch/v1/region"
)

func main() {
    // Set the obtained AK and SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
    // Set the request client. In this example, CN North-Beijing4 is used.
    client := imagesearch.NewImageSearchClient(
        imagesearch.ImageSearchClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())
    //Set the instance name.
    request := &model.RunQueryInstanceRequest{
        InstanceName: "instance-name",
    }

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

2. Execute the sample code. If the instance querying result is displayed on the console, the code is successfully executed. For details, see [Querying the Details About an Instance](#).

```
RunQueryInstanceResponse {"domain": "general image search", "desc": "", "registerDate":
1638344126274, "expiredDate": -1, "level": 30000000, "tags":
["animal", "plant"], "status": "NORMAL", "instanceName": "instance-name"}
```