Alibaba Cloud
Alibaba Cloud SDK

Java SDK

Issue: 20181024

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade

secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *{stand | slave}*

Contents

1 Getting started

Welcome to use the Alibaba Cloud Software Development Kit (SDK). The Alibaba Cloud SDK for Java allows you to use Alibaba Cloud resources such as Elastic Compute Service (ECS), Server Load Balancer (SLB), and CloudMonitor without complex coding to build your own cloud Java applications. This tutorial provides a step-by-step guidance on setting up a development environment in Java. If you have any problem when using Alibaba Cloud SDK, please join the DingTalk group with ID 11771185 (the official SDK customer service group of Alibaba Cloud) for consultation.

Prerequisites

- An Alibaba Cloud account, and the corresponding Access Key ID and secret. You can create and get your AccessKey on the *Alibaba Cloud console*, or contact your administrator.
- To use Alibaba Cloud Java SDK to call APIs of a product, you must first activate the product on the Alibaba Cloud console if required.
- JDK 1.6 or later.

Install Java SDK

If you use Apache Maven to manage Java projects, you only need to add corresponding dependencies to the *pom.xml* file of the projects. You can download Maven dependencies of different cloud products from *Alibaba Cloud GitHub*.

The core library must be installed no matter which cloud product to use. For example, you must install both the ECS SDK library and the SDK core library if you want to use ECS resources.

If the version of the core library is 3.7.0 and the version of ECS library is 4.11.0, you must declare these two SDK libraries in the *pom.xml* file as follows.

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>3.7.0</version>
</dependency>
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-ecs</artifactId>
    <version>4.11.0</version>
</dependency>
```

Use Java SDK

The following code example shows the three main steps to use Alibaba Cloud Java SDK:

1. Create and initiate the DefaultAcsClient instance.
2. Create an API request and set parameters.
3. Initiate the request and process the response.

```
package com.testprogram;
 import com.aliyuncs.profile.DefaultProfile;
 import com.aliyuncs.DefaultAcsClient;
 import com.aliyuncs.IAcsClient;
 import com.aliyuncs.exceptions.ClientException;
 import com.aliyuncs.exceptions.ServerException;
 import com.aliyuncs.ecs.model.v20140526.*;
 public class Main {
     public static void main(String[] args) {
         // Create and initialize a DefaultAcsClient instance
         DefaultProfile profile = DefaultProfile.getProfile(
             "<your-region-id>", // Region ID
             "<your-access-key-id>", // The AccessKey ID of the RAM
account
             "<your-access-key-secret>"); // The AccessKey Secret of
 the RAM account
         IAcsClient client = new DefaultAcsClient(profile);
         // Initiate the request and handle the response or
exceptions
         DescribeInstancesRequest request = new DescribeInstancesReq
uest();
         request.setPageSize(10);
         request.setConnectTimeout(5000); // Set the connection
timeout to 5000 milliseconds
         request.setReadTimeout(5000); // Set the read timeout to
5000 milliseconds
         // Initiate the request and process the response
         DescribeInstancesResponse response;
         try {
             response = client.getAcsResponse(request);
             for (DescribeInstancesResponse.Instance instance:
response.getInstances()) {
                 System.out.println(instance.getPublicIpAddress());
             }
         } catch (ServerException e) {
             e.printStackTrace();
         } catch (ClientException e) {
             e.printStackTrace();
         }
     }
 }
```

2 Use SDK

2.1 Use Java SDK

This document introduces how to install and use Alibaba Cloud Java SDK.

Install Java SDK

Alibaba Cloud Java SDK supports JDK 1.6 or later. You can install the Java SDK using the following two methods:

- Use Maven (recommended)

  If you use Maven to manage project dependencies, add the following code blocks to the `pom.xml` file to install Java SDK :

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>3.5.0</version>
</dependency>
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-ecs</artifactId>
    <version>3.0.0</version>
</dependency>
```

- Import JARs to the integrated development environment

  Note:
  This installation method will be phased out in the next major version, and will only support the Maven installation method.

  If you are using Eclipse or IntelliJ, you can install the Java SDK directly by importing JAR files. You can download the JAR files of cloud products in *Alibaba Cloud SDK*.

  — Use Eclipse

  Complete the following steps to install Alibaba Cloud Java SDK:

  1. Copy the downloaded `aliyun-java-sdk-xxx.jar` file to your project folder.
  2. In Eclipse, right click the project, and then select **Properties**.
  3. In the displayed dialog box, click**Java Build Path** > **Libraries** > **Add JARs**to add the downloaded JAR files.
  4. Click **Apply and Close**.

  — Use IntelliJ

Complete these steps to install Alibaba Cloud Java SDK using IntelliJ:

1. Copy the downloaded `aliyun-java-sdk-xxx.jar` files to your project folder.
2. Open your project in IntelliJ, and click**File** > **Project Structure**.
3. Click **Apply**, and then click **OK**.

Set up credentials

When using the Alibaba Cloud Java SDK to use Alibaba Cloud resources, you have to provide your identification for authentication.

Java SDK supports the following authentication methods:

Authentication	Description
AccessKey	Use the AccessKey to complete the authentication.
STS Token	Use the STS Token to complete the authentication.
RamRoleArn	Use the AssumeRole of the RAM account to complete the authentication.
EcsRamRole	Use the RAM role of an ECS instance to complete the authentication.
RsaKeyPair	Use the RSA key pair to complete the authentication (supported only on Japanese site)

This document uses AccessKey as an example to illustrate how to set up credentials. To ensure the security of your account, it is recommend using your RAM account instead of the primary account. The primary account has full access to all of your cloud services, while the RAM account has limited access granted by the primary account to the cloud services. Firstly, create an AccessKey as described in ,

and then set up your credentials when initializing AcsClient as follows:

Note:
Do not disclose any code containing your AccessKey, for example, do not commit the code to public GitHub projects. Otherwise, your Alibaba Cloud account may be compromised.

```
DefaultProfile profile = DefaultProfile.getProfile(
    "<your-region-id>", // region ID
    "<your-access-key-id>", //  AccessKey ID of RAM account
```

```
    "<your-access-key-secret>"); // AccessKey Secret of RAM account
```

Initiate a call

This document uses ECS as an example to illustrate how to use Alibaba Cloud Java SDK to make a request:

1. Create an AcsClient client.

```
IAcsClient client = new DefaultAcsClient(profile);
```

2. Create a request.

   The naming convention for requests is `${apiName}Request`. Where **`${apiName}`** is the API name, such as `DescribeInstances`.

   When multiple product SDKs are used, different requests may have the same name. Differentiate the request according to the package.

```
DescribeInstancesRequest request = new DescribeInstancesRequest();
request.setPageSize(10);
request.setConnectTimeout(5000); // Set the connection timeout to
5000 milliseconds
request.setReadTimeout(5000); // Set the read timeout to 5000
milliseconds
```

3. Make a call and handle the response.

```
DescribeInstancesResponse response;
try {
    response = client.getAcsResponse(request);
    for (DescribeInstancesResponse.Instance instance:response.
getInstances()) {
        System.out.println(instance.getPublicIpAddress());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
```

```
}
```

All the returned attributes are deserialized into the response. You can directly call `response.getXXX()` to obtain the response attributes.

```
instanceStatus := response.getStatus()
```

However, if an exception occurs or you want to obtain the original HTTP response, you can use the `doAction()` method to obtain the original response.

```
HttpResponse response = client.doAction(request);
```

2.2 Endpoint management

An endpoint is the service entry of an Alibaba Cloud service. The endpoints of a service vary by regions. For example, the endpoint of an ECS instance in the China (Hangzhou) region is `ecs-cn-hangzhou.aliyuncs.com`, but the endpoint is `ecs.ap-northeast-1.aliyuncs.com` if the ECS instance is located in the Japan (Tokyo) region. Alibaba Cloud SDK has a built-in endpoint addressing module. After a request is sent, Alibaba Cloud SDK will find the endpoint to use according to the region ID and product ID that are specified when creating a client.

Exception handling

If an error occurs, such as`SDK.InvalidRegionID` or `SDK.EndpointResolvingError` , refer to the following solutions to troubleshoot.

Error codes	Error message	Resolution
SDK.InvalidRegionId	Can not find endpoint to access.	The current SDK version is too low. Please upgrade the SDK core library `aliyun-java-sdk-core`to version 2.9.0 or higher.
SDK.EndpointResolvingError	No such region <region-id>. Please check your region ID.	Please check your region ID. You can call the **`DescribeRegions`** API to find the region ID.
SDK.EndpointResolvingError	No endpoint for product <product-id>.	Solution: • The current SDK version is too low. Please upgrade the SDK core library `aliyun-java-sdk-core` to version 2.9.0 or higher, and upgrade the product SDK such as aliyun-java-sdk-ecs to the latest version.

Error codes	Error message	Resolution
		• Configure the endpoint to use directly. For more information, see *Set up an endpoint*.
SDK.EndpointResolvingError	No endpoint in the region < region-id> for product <product -id>.	Cannot find the endpoint of the product in the specified region. Solution: • The product may be not available in the specified region. Please change the region ID accordingly. • Upgrade the SDK core library `aliyun-java-sdk-core` to the latest version to use the latest endpoint addressing configurations. A new version of the SDK core library will be released when the service endpoint is changed. • Configure the endpoint to use directly. For more information, see *Set up an endpoint*.

Set up an endpoint directly

Refer to the following code to set up an endpoint for a request to use. See the API documentation of each product to find available endpoints.

```
DescribeInstancesRequest request = new DescribeInstancesRequest();
// set up an endpoint for this request
configuration.setEndpoint("ecs-cn-hangzhou.aliyuncs.com");
DescribeInstancesResponse response = client.getAcsResponse(request);
```

2.3 Configure STS token

Using your Alibaba Cloud AccessKey directly to develop applications will have potential security risk. To enhance your account security, you can use the Security Token Service (STS) token issued for a subaccount to access Alibaba Cloud services.

Introduction to STS token

Security Token Service is a cloud service that provides short-term access control for Alibaba Cloud accounts (or RAM users). Through STS, you can issue an access credentials with custom time limit and access right to federated users (users managed by your local account system). Federated users can use STS short-term access credentials to call the Alibaba Cloud APIs directly , or log on to the Alibaba Cloud console to manage the authorized resources.

Using the STS token as an access credential has the following advantages:

- Using STS token will reduce the risks of a compromised AccessKey ID and AccessKey Secret, particularly reducing risks for your mobile devices.
- STS token has flexible permission control. You can control the access permission in a finer granularity for products including SLB and ECS, according to the RAM role.

Note:

Make sure that the product you are calling supports STS.

Set up STS token

Two methods are available to set up the STS token:

- Method 1: Use STS token directly

  When using STS token directly, you need to maintain the STS yourself.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.auth.BasicSessionCredentials;
import com.aliyuncs.ecs.model.v20140526. DescribeInstancesRequest;
import com.aliyuncs.ecs.model.v20140526. DescribeInstancesResponse;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.profile.DefaultProfile;
public class SimpleSTSTokenSample {
    public static void main(String[] args) {
        BasicSessionCredentials credentials = new BasicSessi
onCredentials(
            "<your-access-key-id>",
            "<your-access-key-secret>",
            "<your-session-token>"

        DefaultProfile profile = DefaultProfile.getProfile("<your-
region-id>");
        DefaultAcsClient client = new DefaultAcsClient(profile,
credentials);
        DescribeInstancesRequest request = new DescribeInstancesReq
uest();
        try {
            DescribeInstancesResponse response = client.getAcsResp
onse(request);
        } catch (ClientException e) {
            System.err.println(e.toString());
        }
```

  where:

  - **region-id** is the ID of the region that you are using. You can obtain the region ID by calling the **DescribeRegions** API.
  - **sts-access-key-id**, **sts-access-key-secret**, and **sts-session-token** are credentials returned by calling the **AssumeRole** API.
- Method 2: Use SDK to manage STS tokens

You can create a new NewClientWithRamRoleArn object to allow Alibaba Cloud Go SDK to create and maintain STS tokens.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.auth.BasicCredentials;
import com.aliyuncs.auth.STSAssumeRoleSessionCredentialsProvider;
import com.aliyuncs.ecs.model.v20140526. DescribeInstancesRequest;
import com.aliyuncs.ecs.model.v20140526. DescribeInstancesResponse;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.profile.DefaultProfile;
public class UseRoleArnSample {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("<your-
region-id>");
        BasicCredentials basicCredentials = new BasicCredentials(
            "<your-access-key-id>",
            "<your-access-key-secret>"

        STSAssumeRoleSessionCredentialsProvider provider = new
STSAssumeRoleSessionCredentialsProvider(
            basicCredentials,
            "<your-role-arn>",
            profile

        DefaultAcsClient client = new DefaultAcsClient(profile,
provider);
        DescribeInstancesRequest request = new DescribeInstancesReq
uest();
        try {
            DescribeInstancesResponse response = client.getAcsResp
onse(request);
        } catch (ClientException e) {
            System.err.println(e.toString());
```

where:

- **`role-arn`** is the role resource descriptor. You can obtain it on the role details page from the *RAM console*.
- **`role-session-name`** is a temporary role name. You can call the **`AssumeRole`** API to create a temporary identity. After the temporary identity is created, you can use the value set for the **`RoleSessionName`** parameter as the **`role-session-name`** parameter in this method.

2.4 Use CommonRequest

If an Alibaba Cloud product does not provide an SDK for its APIs, you can use the generic calling method (CommonRequest) to call the product APIs. By using the CommonRequest calling method, you can call any API.

Characteristics of CommonRequest calling

The CommonRequest calling method has the following characteristics:

1. Light weight: You only need to download the core package, and do not need to download and install all product SDKs.
2. Easy and convenient: You can use the latest API without updating the SDK.
3. Fast iteration: Updates are made often and are made available quickly.

Call an API using CommonRequest

The APIs of Alibaba Cloud products can be classified into two types, RPC and RESTful styles. The method of making a CommonRequest request varies based on the specific API type.

In general, the API required by the **`Action`** parameter belongs to the RPC type and the API required by the **`PathPattern`** parameter belongs to the RESTful type. In general, all APIs in a product are of the same type. Each API only supports calling of a specific type. Entering wrong codes may call other APIs or receive the error message of `ApiNotFound`.

To make a CommonRequest request, you must obtain the values for the following parameters. You can obtain the values of these parameters from the API documents at the *Document Center*. Besides, you can obtain the parameters of an API through *OpenAPI Explorer*.

- Domain: The universal domain name of a product.
- Version: The version of the API, in the format of YYYY-MM-DD.

  You can get the API version from the common parameters in the API document of a product.
- API information: The name of the API to call.

  — If an API is an RPC API, such as ECS and RDS, you must obtain the value of the **`Action`** parameter and specify the API to call in the form of `request.ApiName = "<Action>"`

  For example, if the Action value of the ECS-RunInstances API is **`RunInstances`**, you can use `request.ApiName = "RunInstances"` to specify the API to call when making a CommonRequest request.

- If an API is a RESTful API, such as Container Service, you must obtain the value of the **`PathPattern`** parameter and specify the RESTful path to call in the form of `request.PathPattern = "<PathPattern>"`.

  For example, the **`PathPattern`** value of the CS-GetClusterList API is `/clusters`, you can use `request.PathPattern = "/clusters"` to specify the RESTful path when making a CommonRequest request.

Example: Call an RPC API

The following code shows how to use the CommonRequest calling method to call the **`DescribeInstanceStatus`** API:

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
public class Sample {
    public static void main(String[] args) {
        //Create a DefaultAcsClient instance and initialize it.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<your-region-id>", // your region ID
            "<your-access-key-id>", // your AccessKey ID
            "<your-access-key-secret>"); // your AccessKey Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and set parameters
        CommonRequest request = new CommonRequest();
        request.setDomain("ecs.aliyuncs.com");
        request.setVersion("2014-05-26");
        request.setAction("DescribeInstanceStatus");
        request.putQueryParameter("PageNumber", "1");
        request.putQueryParameter("PageSize", "30");
        try {
            CommonResponse response = client.getCommonResponse(request
);
            System.out.println(response.getData());
        } catch (ServerException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ClientException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
```

Example: Call a RESTful API

The following code shows how to use commonrequest to call the Container Service's view of all cluster instance interfaces:

```
import com.aliyuncs.CommonRequest;
```

```
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
public class Sample {
    public static void main(String[] args) {
        //Create a DefaultAcsClient instance and initialize it.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<Your-region-ID>", // your region ID
            "<Your-access-key-ID>", // your AccessKey ID
            "<Your-access-key-secret>"); // your AccessKey Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and set parameters
        CommonRequest request = new CommonRequest();
        request.setDomain("cs.aliyuncs.com");
        request.setVersion("2015-12-15");
        request.setUriPattern("/clusters");
        try {
            CommonResponse response = client.getCommonResponse(request
);
            System.out.println(response.getData());
        } catch (ServerException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ClientException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
```

2.5 Handle errors

Alibaba Cloud Java SDK returns corresponding exceptions when an error occurs on the service side or the SDK side. These exceptions contain detailed error information, including the error code (ErrCode) and error message (ErrMsg).

You do not need to handle the exceptions returned by the Alibaba Cloud Java SDK. You only need to resolve the errors returned by the service.

- `ServerException` is the exception returned by the corresponding Alibaba Cloud service.
- `ClientException` is the exception returned by the Alibaba Cloud Java SDK.

For example, when the following error occurs, you have to modify the ID of the AccessKey according to the error message.

```
com.aliyuncs.exceptions.ClientException: InvalidAccessKeyId.NotFound
 : Specified access key is not found.
```

If you want to handle the client errors returned by the Alibaba Cloud Java SDK, refer to the following example to write your codes:

```
try {
```

```
    FooResponse response = client.getAcsResponse(request);
    // Handle the response

}catch (ServerException e){
    // You can add your own error handling logic here
    // For example, print the error message
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
    // If the problem is tricky, you can open a ticket and provide the
 RequestId to us
    System.out.println("ResponseId=" + e.getRequestId());
}catch (ClientException e){
    // You can add your own error handling logic here
    // For example, print the error message
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
```

3 Examples

3.1 ApsaraDB for RDS (Relational Database Service)

This tutorial uses the **`CreateDBInstance`** API of ApsaraDB RDS to show you how to use Alibaba Cloud Java SDK to create an ApsaraDB for RDS instance.

ApsaraDB for RDS (Relational Database Service) is a stable and reliable online database service that supports the elastic scaling function. ApsaraDB for RDS is based on the Apsara distributed system and high-performance storage of ephemeral SSD and supports MySQL, SQL Server, and PostgreSQL engines. ApsaraDB for RDS offers a complete set of solutions for backup, recovery, monitoring, migration, disaster recovery, and troubleshooting database operation and maintenanc e.

Code example

Note:

Running the code in this example will create an ApsaraDB RDS instance and generate fees.

```
import java.util.UUID;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.rds.model.v20140815. CreateDBInstanceRequest;
import com.aliyuncs.rds.model.v20140815. CreateDBInstanceResponse;
public class Demo {
    public static void main(String[] args) {

        //Create the DefaultAcsClient instance and initialize it.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<Your-region-ID>", // your region ID
            "<your-access-key-id>", // your AccessKey ID
            "<your-access-key-secret>"); // your AccessKey Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // Create a request and set parameters

        CreateDBInstanceRequest request = new CreateDBInstanceRequest
();
        request.setEngine("MySQL");
        request.setEngineVersion("5.7");
        request.setDBInstanceClass("mysql.n1.micro. 1");
        request.setDBInstanceStorage(20);
        request.setDBInstanceNetType("Intranet");
        request.setSecurityIPList("0.0.0.0/0");
        request.setPayType("Postpaid");
        request.setDBInstanceDescription("MyRds");
        request.setClientToken(UUID.randomUUID().toString());

        // Initiate a request and handle the response or exceptions
```

```
        CreateDBInstanceResponse response;
        try {
            response = client.getAcsResponse(request);
            String dbInstanceId = response.getDBInstanceId();
            System.out.println("Create dbInstance success, instanceId
 = " + dbInstanceId);
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
```

3.2 Elastic Compute Service (ECS)

3.2.1 Create an ECS Instance

The example shows how to create an ECS instance by calling the **`CreateInstanceRequest`** API.

To create an ECS instance, obtain the following information:

- Image ID

  Obtain the ID of the image to use by calling the **`DescribeImages`** API.

- Instance types

  Select the instance type to use. For more information, see *Instance type families*.

Code example

Note:

Running the code in this example will create an ECS instance and generate fees.

```
import java.util.UUID;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.ecs.model.v20140526. CreateInstanceRequest;
import com.aliyuncs.ecs.model.v20140526. CreateInstanceResponse;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
public class Demo {
    public static void main(String[] args) {
        //Create a DefaultAcsClient instance and initialize it
        DefaultProfile profile = DefaultProfile.getProfile(
            "<your-region-id>", // your region ID
            "<your-access-key-id>", // your AccessKey ID
            "<your-access-key-secret>") // your AccessKey Secret
        IAcsClient client = new DefaultAcsClient(profile);
         // Create an API request and set parameters
        CreateInstanceRequest request = new CreateInstanceRequest();
        request.setImageId("alinux_17_01_64_20G_cloudinit_20171222.vhd
");
```

```
        request.setInstanceName("MyEcsInstance");
        request.setSecurityGroupId("<your-security-group-id>");
        request.setInstanceType("ecs.t1.small");
        request.setClientToken(UUID.randomUUID().toString());
        // Initiate a request and handle the response or exceptions
        CreateInstanceResponse response;
        try {
            response = client.getAcsResponse(request);
            String instanceId = response.getInstanceId();
            System.out.println("Create instance success, instanceId =
 " + instanceId);
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

3.3 Server Load Balancer (SLB)

This tutorial uses the **`CreateLoadBalancer`** API of SLB to show you how to use Alibaba Cloud Java SDK to call SLB APIs.

Alibaba Cloud Server Load Balancer (SLB) is a traffic distribution control service that distributes the incoming traffic among multiple Elastic Compute Service (ECS) instances according to the configured forwarding rules. It expands the service capabilities of the application and increases the availability of the application.

Code example

Note:

Running the code in this example will create an SLB instance and generate fees.

```
import java.util.UUID;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.slb.model.v20140515. CreateLoadBalancerRequest;
import com.aliyuncs.slb.model.v20140515. CreateLoadBalancerResponse;
public class Demo {
    public static void main(String[] args) {
        //Create a DefaultAcsClient instance and initialize it.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<Your-region-ID>", // your region ID
            "<your-access-key-id>", // your AccessKey ID
            "<your-access-key-secret>"); // your AccessKey Secret

        IAcsClient client = new DefaultAcsClient(profile);
        // Create a request and set parameters
        CreateLoadBalancerRequest request = new CreateLoadBalancerRe
quest();
        request.setLoadBalancerName("MyLoadBalancer");
```

```
        request.setAddressType("internet");
        request.setClientToken(UUID.randomUUID().toString());
        // Initiate a request and handle the response or exceptions
        CreateLoadBalancerResponse response;
        try {
            response = client.getAcsResponse(request);
            String loadBalancerId = response.getLoadBalancerId();
            System.out.println("Create loadBalancer success,
loadBalancerId = " + loadBalancerId);
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
```

3.4 Virtual Private Cloud (VPC)

This tutorial uses the **`CreateVpc`** API of VPC to show you how to use Alibaba Cloud Java SDK to call VPC APIs.

Virtual Private Cloud (VPC) is a private network established in Alibaba Cloud. VPCs are logically isolated from other virtual networks in Alibaba Cloud. You have full control over your VPC, such as specifying the IP address range of your VPC, and configuring route tables and network gateways. You can also use Alibaba Cloud resources such as ECS, RDS, and SLB in your own VPC.

Code example

```
import java.util.UUID;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.vpc.model.v20160428. CreateVpcRequest;
import com.aliyuncs.vpc.model.v20160428. CreateVpcResponse;
public class Demo {
    public static void main(String[] args) {
        //Create a DefaultAcsClient instance and initialize it.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<Your-region-ID>", // your region ID
            "<Your-access-key-ID>", // your AccessKey ID
            "<your-access-key-secret>"); // your AccessKey Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // Create a request and set parameters
        CreateVpcRequest request = new CreateVpcRequest();
        request.setVpcName("MyVpc");
        request.setCidrBlock("192.168.0.0/16");
        request.setClientToken(UUID.randomUUID().toString());
        // Initiate a request and handle the response or exceptions
        CreateVpcResponse response;
        try {
            response = client.getAcsResponse(request);
            String vpcId = response.getVpcId();
            System.out.println("Create vpc success, vpcId = " + vpcId
);
        } catch (ServerException e) {
```

```
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
```

4 Errors and troubleshooting

The following are some common errors you may experience when using Alibaba Cloud SDK. Potential causes and resolutions are shown for these commonly experienced errors. Submit your problem to Alibaba Cloud through DingTalk group: 11771185, or *GitHub Issue*.

Error Code	Error information	Cause:	Resolution:
`SDK.CanNotResolveEndpoint`	`Can not resolve endpoint, please check the user guide`	The SDK cannot automatically obtain the endpoint of the called product in the specified region.	Checks whether the provided region ID and endpoint are correct. Run the following codes to set the endpoint: `DefaultProfile.addEndpoint("cn-hangzhou", "cn-hangzhou", "Ecs", "ecs.aliyuncs.com");`
`SDK.JsonUnmarshalError`	`Failed to unmarshal response`	SDK response deserialization fails. In most cases, it is because the response structure received by the SDK does not conform to the API metadata. For example, the fields do not match or the format is incorrect.	You can use the `client.doAction(request)` method to obtain the original HTTP response.
`SDK.TimeoutError`	*`The request timed out 4 times(3 for retry), perhaps we should have the threshold raised a little?`*	The request times out and all retry attempts fail.	• In scenarios such as cross-region calls or low network quality, we recommend that you increase the timeout or the maximum number of retries. • If the problem persists and the network quality is confirmed to be

Error Code	Error information	Cause:	Resolution:
			good, we recommend that you open a ticket.
`SDK.ServerError:InvalidProtocol.NeedSsl`	`Your request is denied as lack of ssl protect .Recommend :https://error-center.aliyun.com/status/search?Keyword=InvalidProtocol.NeedSsl&source=PopGw`	The API only accepts HTTPS request and does not accept HTTP request.	Add the following codes before sending a request: `request.setProtocol(ProtocolType.HTTPS)`