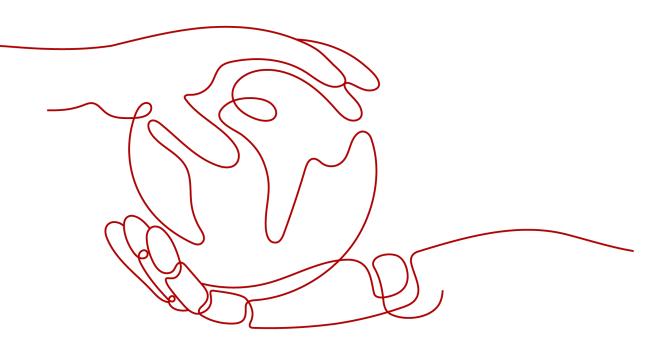
## LakeFormation

# **Development Guide**

 Issue
 01

 Date
 2024-01-31





#### Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. 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 Cloud Computing Technologies Co., Ltd.

#### **Trademarks and Permissions**

NUAWEI and other Huawei trademarks are the property 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 Cloud 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.

# **Contents**

1 About This Document	1
1.1 Application Development	1
1.2 Development Process	
2 Environment Preparation	2
2.1 Overview	
2.2 Preparing a LakeFormation Instance	
2.3 Creating a Client	
2.4 Modifying DNS Information	3
3 Using LakeFormation Java SDK to Develop Programs	5
3.1 Preparing the Development Environment	5
3.2 Reference Example	
3.3 Commissioning Applications	

# About This Document

# **1.1 Application Development**

This document describes the application development process, environment preparation, and program compilation of LakeFormation.

# **1.2 Development Process**

The LakeFormation development process is as follows:

Step	Description	Reference
Preparing the environment	Prepare the development environment.	Environment Preparation
Writing programs based on scenarios	Use LakeFormation SDK to write programs	Using LakeFormation Java SDK to Develop Programs

# **2** Environment Preparation

## 2.1 Overview

**Table 1** describes the environment required for developing lake warehouse building applications.

You also need to prepare an environment for verifying whether an application is running properly.

ltem	Description
Preparing a LakeFormation instance	Create an instance on the LakeFormation console or use an existing instance.
Creating a client	Access the LakeFormation console and create a client on the <b>Clients</b> page.
Modifying DNS information	Access the DNS console and modify the private domain name of the subnet where the Linux environment is located.

#### Table 2-1 Preparation items

# 2.2 Preparing a LakeFormation Instance

- Complete account registration and authorization if you use LakeFormation for the first time. For details, see**Preparations**.
- For how to create a LakeFormation instance, see **Creating an Instance**.
- The AK and SK of the target Huawei Cloud account has been created. For details, see section **Obtaining AK/SK**.

# 2.3 Creating a Client

- **Step 1** Log in to the Huawei Cloud management console.
- **Step 2** In the upper left corner, click and choose **Analytics** > **LakeFormation** to access the LakeFormation console.
- **Step 3** Select the target LakeFormation instance from the drop-down list on the left to access the instance page.
- **Step 4** Click **Clients** in the navigation pane.
- **Step 5** Click **Create**. In the displayed dialog box, set the following parameters and click **OK**.

If no suitable VPC or subnet is available, click **create one** to access the VPC console to create one.

Parameter	Description
Client	LakeFormation client name.
VPC	VPC where the running and commissioning environment is located.
Subnet	Subnet where the environment for running and commissioning is located.

Table 2-2 Parameters for creating a client

- Step 6 After the access client is created, return to the Clients page to view the information about the newly created access client. Wait until the client is created. It is created when its status changes to Running.
- **Step 7** Click **View Details** to view the access client information and record the access IP address.

----End

# 2.4 Modifying DNS Information

- **Step 1** Log in to the Huawei Cloud management console.
- **Step 2** Click the service list icon and choose **Networking > Domain Name Service**.
- **Step 3** In the navigation pane, choose **Private Zones**.
- Step 4 Search for lakeformation.lakecat.com in the search box, locate the row that contains the VPC domain name corresponding to the created client in the Associated VPC column, and click Manage Record Set in the Operation column.
- Step 5 Click Add Record Set, set the following parameters, and click OK.

Parameter	Description
Туре	Select <b>A – Map domains to IPv4 addresses</b> .
TTL (s)	Cache duration of the record set on a local DNS server. Set this parameter as required.
Value	Enter the client access IP address, which can be obtained by performing the operation described in <b>Step 7</b> .

Table 2-3 Parameters for adding a record set

#### NOTICE

When you add a record set, you might see a message that says it conflicts with an existing one. If this happens, look for any conflicting record sets in the existing one.

**Step 6** After a record set is added, you can find it in the record set list.

----End

# **3** Using LakeFormation Java SDK to Develop Programs

# 3.1 Preparing the Development Environment

#### **Preparing the Environment**

**Table 3-1** describes the preparations required before using LakeFormation Java SDK.

ltem	Description			
Java JDK environment	The Java environment is required and the Java version must be JDK 1.8 or later.			
IntelliJ IDEA	Tool used for developing applications. The version must be 2019.1 or other compatible versions. NOTE			
	<ul> <li>If you are using an IBM JDK, ensure that the JDK configured in IntelliJ IDEA is the IBM JDK.</li> </ul>			
	<ul> <li>If you are using an Oracle JDK, ensure that the JDK configured in IntelliJ IDEA is the Oracle JDK.</li> </ul>			
	<ul> <li>If you are using an open JDK, ensure that the JDK configured in IntelliJ IDEA is the Open JDK.</li> </ul>			
	<ul> <li>Do not use the same workspace and the sample project in the same path for different IntelliJ IDEA projects.</li> </ul>			
Maven installation	Basic configuration of the development environment. This tool is used for project management throughout the lifecycle of software development.			
7-Zip	This tool is used to decompress <b>.zip</b> and <b>.rar</b> packages.			
	The 7-Zip 16.04 version is supported.			

Table	3-1	Environment	requirements
Tuble	5	LINNORTH	requirements

#### **Collecting Dependency Information**

#### • Collecting LakeFormation Java SDK dependencies

View the JAR package of LakeFormation Java SDK of the latest version in the Maven repository at **Maven SDK address** and obtain the file content. The following is an example.

	· -····						
groupId> artifactIc version>	Com.huawe I>huaweiclo 3.1.45	ud-sdk-			actId>		
alideana Car							
		Marrian	444	Desiderity	Parathe Incom	Lineare Theast	Download
com huavaicloud sdk	huaweicloud-sdk-lakeformation	3.1.45		r updarny	Security insta	- Control Intell	pom, iar, iavadoc iar, sources iar
com hueveicloud sdk	hueveicloud-sdk-lakeformation	3.1.44		0	-		pom, jar, sources jar, javadoc jar
com hueveicloud sdk	huawaicloud-sdk-lakeformation	3.1.43					pom, iar, iavadoc iar, sources iar
com hueveicloud sdk		3.1.42					pom, jar, javadoc jar, sources jar
	hummicloud-sdk-lakeformation	3.1.41		0			pom, sources jar, javadoc jar
				0			pom, jar, sources jar, javadoc jar
							pom, jar, javadoc jar, sources jar
			-	0			pom, jar, javadoc jar, sources jar
							pom, jar, sources jar, javadoc jar
com hueweicloud sdk	huaweicloud-sdk-lakeformation	3.1.36	0	0			pom, sources jar, javadoc jar, jar
com hueveicloud sdk	huawaicloud-sdk-lakeformation	3.1.35	0				porn, sources Jar, lavadoc Jar, lar
com hueveicloud sdk	huawaicloud-sdk-lakeformation	3.1.34		0			pom, jar, javadoc jar, sources jar
com hueveicloud sdk	hummeicloud-sdk-lakeformation	3.1.33		0			pom, iar, javadoc jar, sources jar
com hummelcloud sdk	hummicloud-sck-lakeformation	3 1 32		0			pom, jar, javadoc jar, sources jar
com hummelckaud sidk	husueicloud-adk-lakeformation	3.1.31					pom, jar, sources jar, javadoc jar
Displaying Top 21 records - X (	Sear Results				head	land	
	ry Releases 👻						Mana Passadana
ⓐ 1.13 ⓐ 1.12 ⓐ 1.12 ⓐ 1.13 ⓐ 1.14 ⓐ 1.14 ⓐ 1.14 ⓐ 1.14 ⓑ	Sout officialetermation-2 1 45-paratic (ar Sout officialetermation-2 1 45-paratic (ar				Gro Arti Ver Ext	ap: com.huanveicloud.sdk hear: huanveicloud-sdk-lakeformation sen: 3.1.45 mston: jar 	- chinarati-
	pendency groupId> cartifactic version> ependence berendence manaecoda manaec	pendency> groupId>com.huawe cartifactId>huaweiclo version>3.1.45ependency> www.harcod & waterior huaecod & water	pendency> groupId>com.huaweicloud.sk cartifactId>huaweicloud-skk- version>3.1.45 ependency>	pendency> groupId>com.huaweicloud.sdkcartifactId>huaweicloud.sdk-lakeforma version>3.1.45 ependency> ************************************	pendency> groupId>com.huaweicloud.sdk cartifactId>huaweicloud.sdk-lakeformationversion>3.1.45 ependency>	pendency> groupId>com.huaweicloud.sdk cartifactId>huaweicloud-sdk-lakeformation cversion>3.1.45 ependency>	pendency> groupId>com.huaweicloud.sdk cartifactId>huaweicloud.sdk cartifactId>huaweicloud-sdk-lakeformation cversion>3.1.45 ependency>

#### Preparing maven-assembly-plugin dependencies

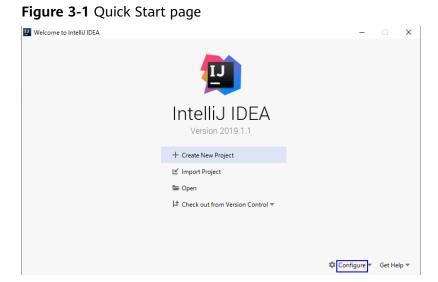
Prepare the following maven-assembly-plugin dependencies in advance:



#### Configuring IntelliJ IDEA and POM Files

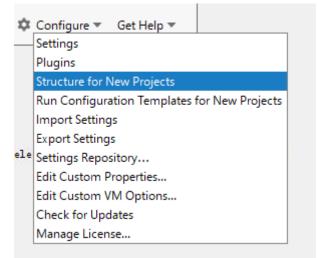
**Step 1** After the IntelliJ IDEA and JDK tools are installed, configure the JDK in IntelliJ IDEA.

1. Start IntelliJ IDEA and choose **Configure**.



2. Select Structure for New Projects from the drop-down list.

Figure 3-2 Clicking Configure



3. On the displayed **Project Structure for New Projects** page, select **SDKs**, click the plus sign (+), and click **JDK**.

5 ,	
Project Structure for N	ew Projects
$\leftarrow \rightarrow$	<u> </u>
Project Settings	Add New SDK
Project Libraries	<ul> <li>Mobile SDK</li> <li>Intellij Platform Plugin SDK</li> </ul>
Platform Settings /	+ Kotlin SDK
SDKs	👘 Android SDK
Global Libraries	X Flex/AIR SDK K Flexmojos SDK
Problems	
	Nothing to show

Figure 3-3 Project Structure for New Projects

4. On the **Select Home Directory for JDK** page that is displayed, select the JDK directory and click **OK**.

Figure 3-4 Select Home Directory for JDK

Select Home Directory for JDK         ×
🛧 😐 🛤 🛤 🛤 X 🕄 🧐 🖕 Hide path
C:\Java\jdk1.8.0_102
> Documents and Settings
> 🖿 elite
> 🖿 ePrint iRight Printer
> ESDK_UC
> FAWinInstallerLog
> FilePackage
> HdpLog
> ideploy
> idplite
> IdpTraceLog
> IrmTool
V Java
> bin
> jdk1.8.0_102
> jre1.8.0_102
> lib
Drag and drop a file into the space above to quickly locate it in the tree
OK         Cancel

5. After selecting the JDK, click **OK** to complete the configuration.

Project Structure fo			
$\rightarrow$	+ -	Name: 1.8	
oject Settings	1.0	JDK home path: C:\Java\jdk1.8.0_102	
Project			
Libraries		Classpath Sourcepath Annotations Documentation Paths	
atform Settings		C:\Java\jdk1.8.0_102\jre\lib\charsets.jar C:\Java\jdk1.8.0_102\jre\lib\deploy.jar	
SDKs		C:\Java\jdk1.8.0_102\jre\lib\deploy.jar C:\Java\jdk1.8.0_102\jre\lib\ext\access-bridge-64.jar	
Slobal Libraries		C:\Java\jdk1.8.0_102\jre\lib\ext\cldrdata.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\dnsns.jar	
Problems		C:\Java\jdk1.8.0_102\jre\lib\ext\jaccess.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\jfxrt.jar C:\Java\jdk1.8.0_102\jre\lib\ext\jdcaledata.jar	
		C:/Java/jdk1.8.0_102/jre/lib/ext/localedata.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\sunec.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\sunjce_provider.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\sunmscapi.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\sunpkcs11.jar	
		C:\Java\jdk1.8.0_102\jre\lib\ext\zipfs.jar C:\Java\jdk1.8.0_102\jre\lib\javaws.jar	
		C:\Java\jdk1.8.0_102\jre\lib\jce.jar	
		C:\Java\jdk1.8.0_102\jre\lib\jfr.jar	
		C:\Java\jdk1.8.0_102\jre\lib\jfxswt.jar	
		<ul> <li>C:\Java\jdk1.8.0_102\jre\lib\jsse.jar</li> <li>C:\Java\jdk1.8.0_102\jre\lib\management-agent.jar</li> </ul>	
		C:\Java\jdk1.8.0_102\jre\lib\plugin.jar	
		C:\Java\jdk1.8.0_102\jre\lib\resources.jar	
		C:\Java\jdk1.8.0_102\jre\lib\rt.jar	
			OK Cancel Apply

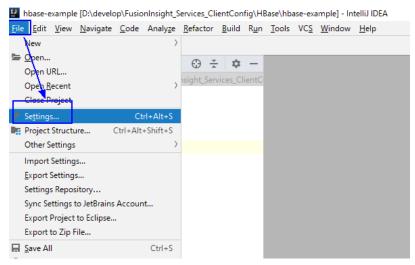
Figure 3-5 Completing the configuration

#### **NOTE**

The operation procedure may vary according to the IDEA version.

- **Step 2** Set the Maven version used by the project.
  - 1. Choose File > Settings... from the main menu of IntelliJ IDEA.

#### Figure 3-6 Settings



 Choose Build, Execution, Deployment > Maven and set Maven home directory to the Maven version installed on the local PC.

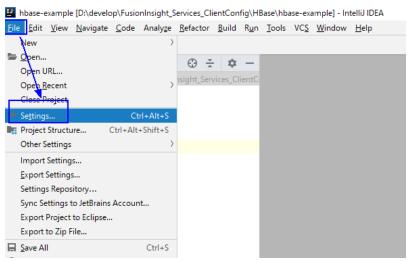
Set **User settings file** and **Local repository** as you need, and click **Apply** > **OK**.

P Settings				×	
Q	Build, Execution, Deploymer	nt → Build Tools → Mav	ven @ For current project		
Appearance & Behavior	Work <u>o</u> ffline				
Keymap	Use plugin <u>r</u> egistry				
▶ Editor	Execute goals recursively				
Plugins (2)	Print exception stack traces				
► Version Control	Always update snapshots				
Build, Execution, Deployment	✓ Update Incices on project open				
▼ Build Tools @	Output level:	·			
► Maven 🖷	Output level:	Info 💌			
Gradle 🖷	Checksum policy:	No Global Policy 🔹			
Gant 🖷	Multiproject build fail policy:	Default 👻			
► Compiler @					
▶ Debugger	Thread count		-T option		
Remote Jar Repositories 🛛 🖻	Maven <u>h</u> ome directory:	Bundled (Maven 3)		▼	
▶ Deployment 💿		Bundled (Maven 3)			
Arquillian Containers 🛛 🖻	User settings file:	D:/maven/apache-mave	n-3.5.2	verride	
Android					
Application Servers	Local <u>r</u> epository:	D:\maven-repo		🗁 🗌 Override	
Coverage					
▶ Docker					
Gradle-Android Compiler 🛛 🖻					
▶ Java Profiler					
Required Plugins 👘					
Languages & Frameworks					
?			ок	Cancel Apply	

Figure 3-7 Selecting the local Maven installation directory

- **Step 3** Set the IntelliJ IDEA text file coding format to prevent garbled characters.
  - 1. Choose File > Settings... from the main menu of IntelliJ IDEA.

#### Figure 3-8 Settings



2. In the navigation tree of the **Settings** page, choose **Editor** > **File Encodings**, and select **UTF-8** for both **Global Encoding** and **Project Encoding**.

#### Figure 3-9 File Encodings

U Settings	X				
Qr	Editor > File Encodings @ For current project Reset				
> Appearance & Behavior	Global Encoding: UTF-8 ~				
Keymap	Project Encoding: UTF-8 ~				
✓ Editor					
> General	Path A Encoding +				
Font	-				
> Color Scheme					
> Code Style @					
Inspections	Encodings are not configured				
File and Code Templates					
File Encodings @					
Live Templates					
File Types					
Android Layout Editor					
> Copyright @					
Android Data Binding					
> Emmet					
GUI Designer 👘	To change encoding Intellii IDEA uses for a file, a directory, or the entire project, add its path if necessary and then select encoding from the encoding its: Built-Inflee encoding (e.g. 35, PHL or XMU) override encoding you specify here. If not specified, files and directories inherit encoding settings from the parent directory or from the Project Encoding.				
Images					
Intentions	Properties Files (*, properties)				
> Language Injections (1)					
Spelling	Default encoding for properties files:				
TODO	BOM for new UTF-8 files				
Plugins					
> Version Control	Create UTF-8 files: with NO BOM				
<ul> <li>Version control</li> <li>Build, Execution, Deployment</li> </ul>	IDEA will NOT add UTF-8 BOM to every created file in UTF-8 encoding				
<ul> <li>Build, Execution, Deployment</li> </ul>					
0	OK Cancel Apply				

- 3. Click **Apply** and **OK** to complete the configuration.
- Step 4 Add the LakeFormation Java SDK dependencies collected in Collecting Dependency Information and the maven-assembly-plugin dependencies to the end of the pom.xml file of Maven.

----End

### 3.2 Reference Example

After the development and running environments are prepared, you can develop samples as required. For example, the reference code is as follows:

package com.huawei.cloud.dalf.lakecat.examples;

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ClientRequestException;
import com.huaweicloud.sdk.core.exception.ServerResponseException;
import com.huaweicloud.sdk.core.http.HttpConfig;
import com.huaweicloud.sdk.lakeformation.v1.LakeFormationClient;
import com.huaweicloud.sdk.lakeformation.v1.model.ListCatalogsRequest;
import com.huaweicloud.sdk.lakeformation.v1.model.ListCatalogsResponse;
import java.util.ArrayList;
import java.util.List;
public class LakeFormationExample {
  public static void main(String[] args) {
     // The getAk() and getSk() methods need to be implemented by yourself. You can obtain the AK/SK
from the configuration item or other locations.
     // Do not hard-code AK and SK in codes.
     String ak = getAk();
     String sk = getSk();
     // projectId: project ID
     String projectId = "{******your project id*****}";
     // 1. Initialize the SDK.
     HttpConfig config = HttpConfig.getDefaultHttpConfig();
     config.withIgnoreSSLVerification(true);
     List<String> endpoints = new ArrayList<>();
     endpoints.add("lakeformation.lakecat.com");
```

```
BasicCredentials basicCredentials = new
BasicCredentials().withAk(ak).withSk(sk).withProjectId(projectId);
     // 2. Create a LakeFormationClient instance.
     LakeFormationClient client = LakeFormationClient.newBuilder()
        .withHttpConfig(config)
        .withCredential(basicCredentials)
        .withEndpoints(endpoints)
        .build();
     // 3. Create a request and add parameters.
     ListCatalogsRequest listCatalogsRequest =
        new ListCatalogsRequest().withInstanceId("{******your instance id*****}");
     // 4. Query the catalog list.
     try {
        ListCatalogsResponse response = client.listCatalogs(listCatalogsRequest);
        System.out.println(response.getHttpStatusCode());
        System.out.println(response);
     } catch (ClientRequestException | ServerResponseException e) {
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getMessage());
     }
  }
```

# **3.3 Commissioning Applications**

After configuring the sample code, export the JAR package and upload it to the node where the LakeFormation client is deployed.

**Step 1** Export a JAR file.

}

Choose Maven, locate the target project name, and double-click clean under Lifecycle to run the clean command of Maven.

Choose Maven, locate the target project name, and double-click install under Lifecycle to run the install command of Maven.

-	
🕼 👻 😫 📄 🛛 Git: 🖌 🗸 🕓 🍤 📑 🗖 💽	Q
Maven 🌣 —	m
:46 <mark></mark> 영 🔩 土   +   <b>&gt; m -// 중</b> 표 >>	Maven
r () . ( Profiles	2
▼ 🚮 hbase-example	<b>C2</b>
🔻 🏣 Lifecycle	S
🗘 clean	🧖 My Review Tasks
e jαl 🗘 validate	Nie
💭 📫 compile	N Ta
th.ld test	Isks
you i 🌼 package	
r, tl 🔅 verify	(()))
🖈 install	())) Database
🔯 site	bas
_L0G. 🌼 deploy	æ
e , 🛛 k / 🕞 🕞 Plugins	
Dependencies	Bea
·=	n <
	😵 Bean Validat

Figure 3-10 Maven clean and install

**Step 2** Run the **maven** command to package the project and run the following command to upload the JAR package to the node where the LakeFormation client is located:

java -cp lakeformation-lakecat-opensource-1.0.0-jar-with-dependencies.jar com.huawei.cloud.dalf.lakecat.examples.LakeFormationExample

The command output is as follows.



----End