

LakeFormation

开发指南

文档版本 01
发布日期 2024-10-22



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

目录

1 服务介绍	1
1.1 应用开发简介.....	1
1.2 开发流程.....	1
2 环境准备	2
2.1 简介.....	2
2.2 准备 LakeFormation 实例.....	2
2.3 创建客户端.....	2
2.4 修改 DNS 信息.....	3
3 使用 LakeFormation Java SDK 开发程序	5
3.1 准备开发程序环境.....	5
3.2 参考样例.....	11
3.3 调测程序.....	12

1 服务介绍

1.1 应用开发简介

湖仓构建（DataArts Lake Formation，简称LakeFormation）是企业级数据湖一站式构建服务。

在存算分离架构基础上提供数据湖元数据统一管理的可视化界面及API，兼容Hive元数据模型以及Ranger权限模型，支持对接多种计算引擎及大数据云服务

本文档提供LakeFormation服务应用开发流程、环境准备及程序编写介绍，供需要进行业务应用开发的开发人员使用。

1.2 开发流程

开发流程如下：

表 1-1 开发流程

阶段	说明	参考文档
环境准备	准备LakeFormation开发环境	环境准备
根据场景编写程序	使用LakeFormation SDK编写程序	使用LakeFormation Java SDK开发程序

2 环境准备

2.1 简介

在进行湖仓构建应用开发时，要准备的环境如[表1](#)所示。

同时需要准备运行调测的环境，用于验证应用程序运行正常。

表 2-1 准备项


准备项	说明
准备LakeFormation实例	在LakeFormation控制台创建实例，或者使用已有LakeFormation实例。
创建客户端	在LakeFormation控制台接入管理界面创建客户端。
修改DNS信息	在云解析服务中，修改linux环境所在子网的内网域名信息。

2.2 准备 LakeFormation 实例

- 如果您是第一次使用LakeFormation，请参考[准备工作](#)章节，完成账号注册及授权等操作。
- 创建实例相关步骤请参考[创建LakeFormation实例](#)章节。
- 已创建待操作华为云账号的访问密钥AK/SK信息。具体操作请参考[获取AK/SK](#)章节。

2.3 创建客户端

步骤1 登录华为云管理控制台。

步骤2 在左上角单击“”，选择“大数据 > 湖仓构建 LakeFormation”进入LakeFormation控制台。

步骤3 在左侧下拉框中选择待操作的LakeFormation实例，进入实例界面。

步骤4 在左侧导航栏选择“接入管理”，进入接入管理界面。

步骤5 单击“创建客户端”，在弹出的窗口中填写以下参数后，单击“确定”。

如果没有合适的虚拟私有云或子网，可以单击“前往VPC创建”。

表 2-2 创建客户端

参数	参数说明
客户端名称	LakeFormation客户端名称。
虚拟私有云	准备运行调测的环境所在虚拟私有云。
所属子网	准备运行调测的环境所在子网。

步骤6 创建成功后返回接入管理界面，即可查看到新创建的接入客户端信息。等待客户端创建成功，状态变为“运行中”表示客户端已创建成功。

步骤7 单击“查看详情”查看接入客户端信息，记录“接入IP”。

---结束

2.4 修改 DNS 信息

步骤1 登录华为云管理控制台。

步骤2 在“服务列表”中，选择“网络 > 云解析服务DNS”进入云解析控制台。

步骤3 在左侧选择“内网域名”，进入内网域名界面。

步骤4 在搜索框中搜索“lakeformation.lakecat.com”，查找“已关联的VPC”列为创建的客户端对应的虚拟私有云的域名所在的行，单击对应“操作”列的“管理解析”。

步骤5 单击“添加记录集”，填写以下参数后，单击“确定”。

表 2-3 添加记录集

参数	说明
类型	选择“A - 将域名指向IPv4地址”。
TTL (秒)	解析记录在本地DNS服务器的缓存时间，根据需要填写。
值	填写客户端接入IP，可以通过 步骤7 获取。

须知

如果添加记录集时出现“与已有解析记录冲突”，需要检查已有记录集中是否存在冲突的解析记录。

步骤6 添加成功后，即可在记录集列表中找到解析记录。

----结束

3 使用 LakeFormation Java SDK 开发程序

3.1 准备开发程序环境

准备开发环境

在进行LakeFormation Java SDK开发程序前，要准备的开发环境如表3-1所示。

表 3-1 开发环境

准备项	说明
安装Java和JDK环境	已安装Java环境，Java版本要求使用JDK1.8或更高版本。
安装和配置IntelliJ IDEA	用于开发应用程序的工具，版本要求：2019.1或其他兼容版本。 说明 <ul style="list-style-type: none">若使用IBM JDK，请确保IntelliJ IDEA中的JDK配置为IBM JDK。若使用Oracle JDK，请确保IntelliJ IDEA中的JDK配置为Oracle JDK。若使用Open JDK，请确保IntelliJ IDEA中的JDK配置为Open JDK。不同的IntelliJ IDEA不要使用相同的workspace和相同路径下的示例工程。
安装Maven	开发环境的基本配置。用于项目管理，贯穿软件开发生命周期。
7-zip	用于解压“*.zip”和“*.rar”文件。 支持7-Zip 16.04版本。

准备依赖信息

- 准备LakeFormation Java SDK依赖
参考[Maven SDK地址](#)，在Maven仓库中查看LakeFormation Java SDK最新版本Jar包，并获取文件内容，例如：


```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-lakeformation</artifactId>
  <version>3.1.45</version>
</dependency>
```

- **准备maven-assembly-plugin插件相关依赖**
需提前准备如下maven-assembly-plugin插件相关相关依赖：

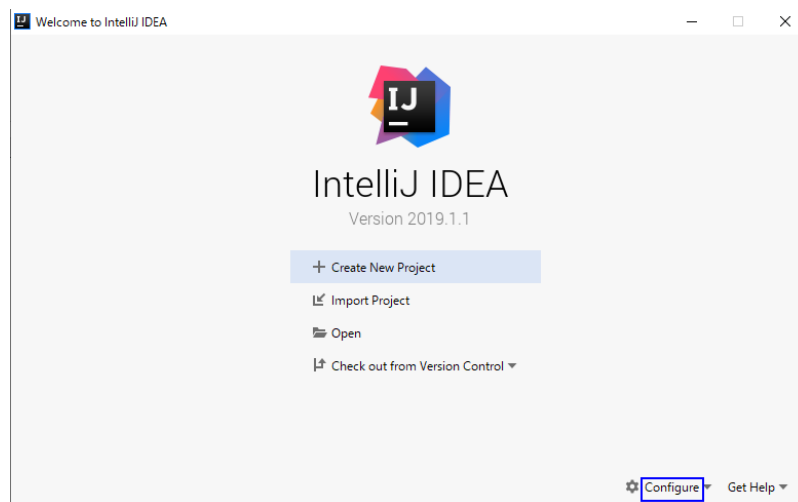
```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-assembly-plugin</artifactId>
      <version>3.3.0</version>
      <configuration>
        <descriptorRefs>
          <descriptorRef>jar-with-dependencies</descriptorRef>
        </descriptorRefs>
        <archive>
          <manifest>
            <mainClass>com.huawei.cloud.dalf.lakecat.examples.CatalogExample</mainClass>
          </manifest>
        </archive>
      </configuration>
      <executions>
        <execution>
          <id>make-assembly</id>
          <phase>package</phase>
          <goals>
            <goal>single</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

配置 IDEA 及 pom 文件

步骤1 安装IntelliJ IDEA和JDK工具后，需要在IntelliJ IDEA配置JDK。

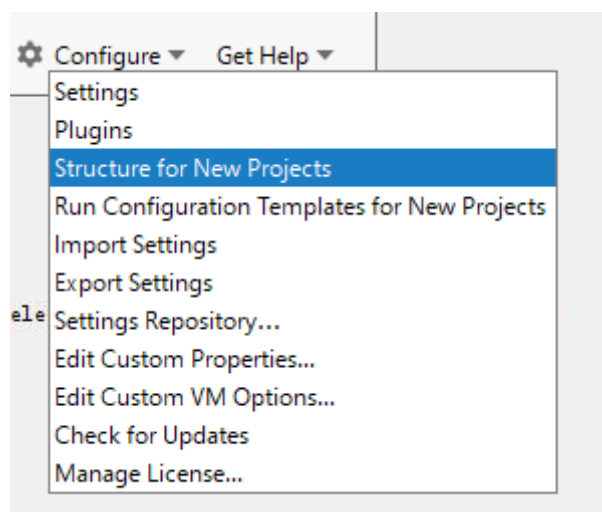
1. 打开IntelliJ IDEA，选择“Configure”。

图 3-1 Quick Start



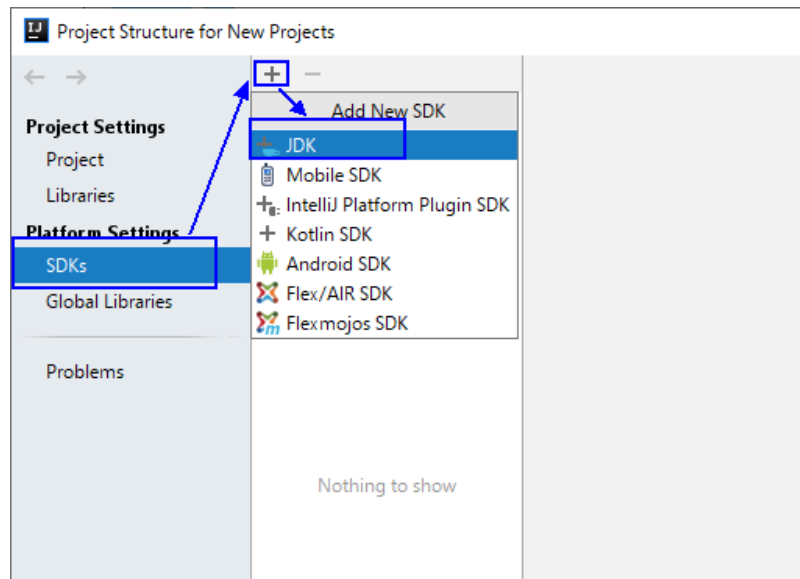
2. 在下拉框中选择“Structure for New Projects”。

图 3-2 Configure



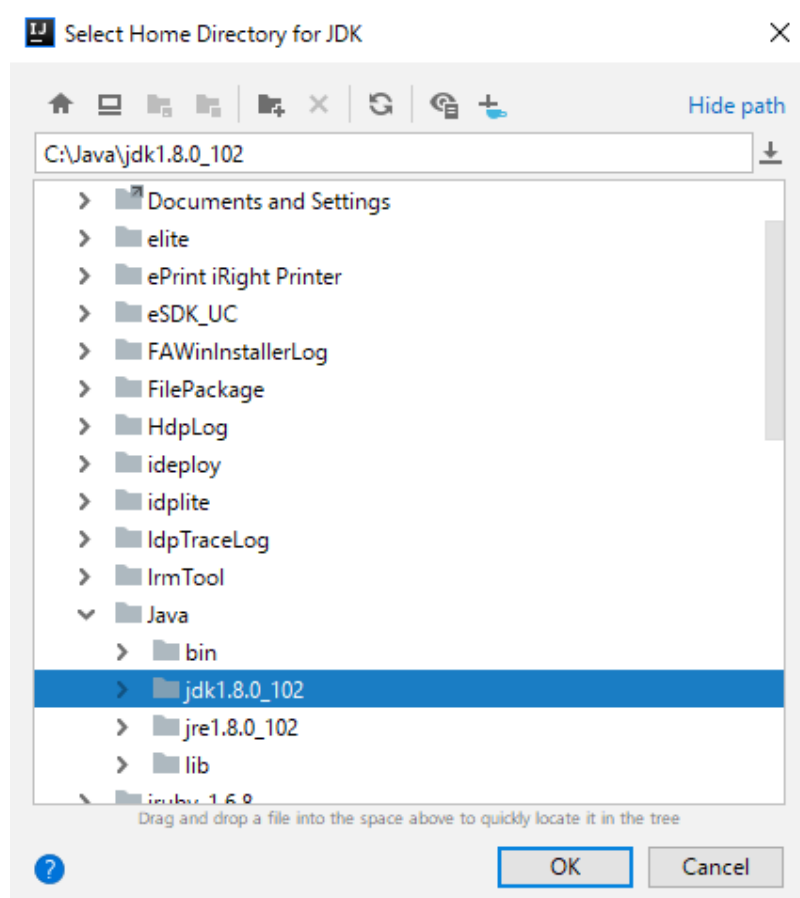
3. 在弹出的“Project Structure for New Projects”页面中，选择“SDKs”，单击加号添加JDK。

图 3-3 Project Structure for New Projects



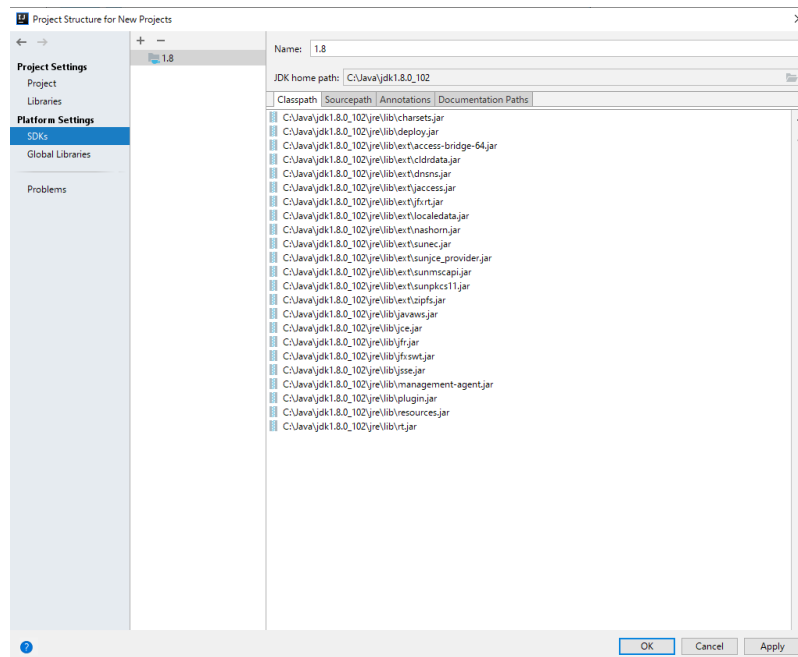
4. 在弹出的“Select Home Directory for JDK”窗口，选择对应的JDK目录，然后单击“OK”。

图 3-4 Select Home Directory for JDK



5. 完成JDK选择后，单击“OK”完成配置。

图 3-5 完成 JDK 配置



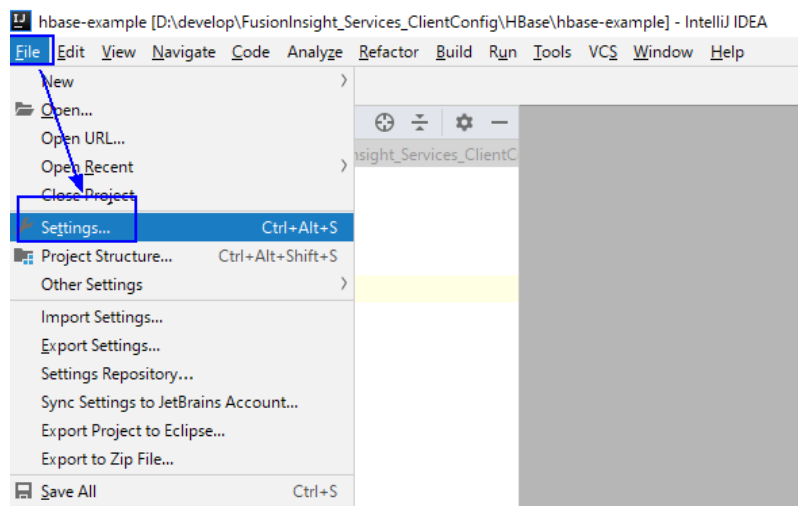
说明

不同的IDEA版本的操作步骤可能存在差异，以实际版本的界面操作为准。

步骤2 设置项目使用的Maven版本。

1. 在IntelliJ IDEA的菜单栏中，选择“File > Settings...”。

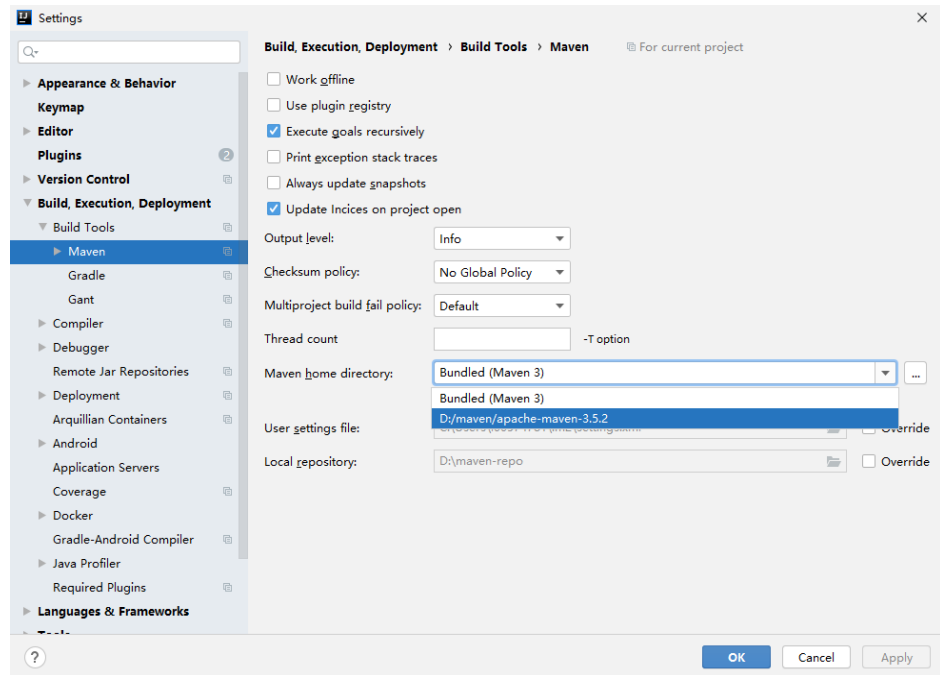
图 3-6 Settings



2. 选择“Build,Execution,Deployment > Maven”，选择“Maven home directory”为本地安装的Maven版本。

然后根据实际情况设置好“User settings file”和“Local repository”参数，依次单击“Apply > OK”。

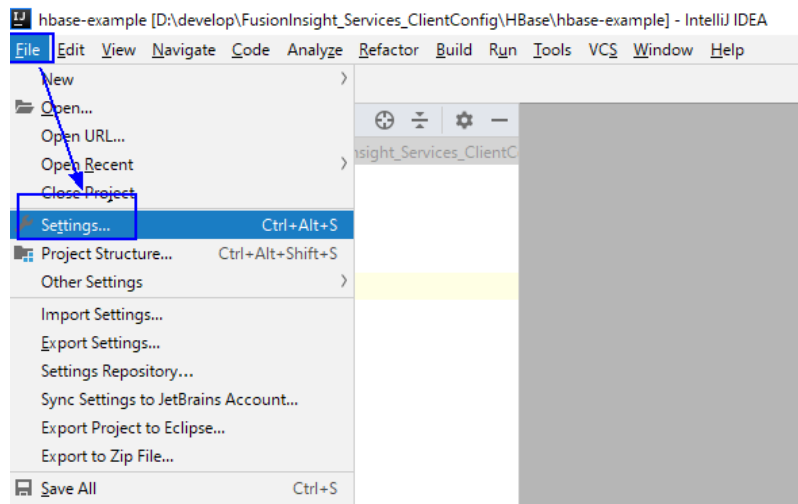
图 3-7 选择本地 Maven 安装目录



步骤3 设置IntelliJ IDEA的文本文件编码格式，解决乱码显示问题。

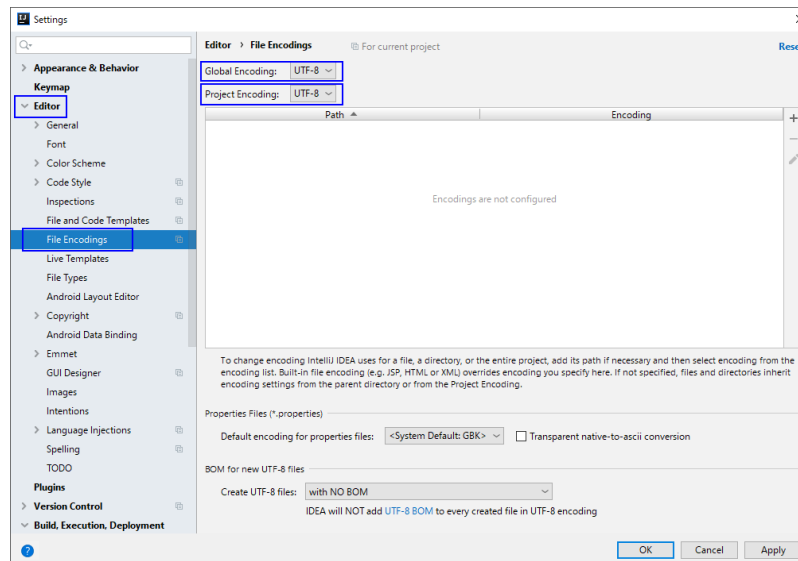
1. 在IntelliJ IDEA的菜单栏中，选择“File > Settings...”。

图 3-8 Settings



2. 在弹出的“Settings”页面左边导航上选择“Editor > File Encodings”，分别在右侧的“Global Encoding”和“Project Encoding”的下拉框中，选择“UTF-8”。

图 3-9 File Encodings



3. 然后单击“Apply”和“OK”，完成编码配置。

步骤4 在Maven的pom.xml文件末尾添加准备依赖信息中准备的LakeFormation Java SDK依赖和maven-assembly-plugin插件相关的依赖。

----结束

3.2 参考样例

在准备好开发和运行环境后，用户可根据需求开发样例。

例如使用LakeFormation Java SDK开发程序的参考代码如下：

（如下代码介绍了初始化SDK、创建LakeFormationClient实例、创建请求，添加参数、查询Catalogs列表的相关代码。）

```
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) {
        // getAk().getSk()方法需自行实现，可以从配置项或其他位置获取AK/SK
        // 请勿在代码中硬编码AK/SK
        String ak = getAk();
        String sk = getSk();

        // projectId: 项目ID
        String projectId = "{*****your project id*****}";

        // 1.初始化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.创建LakeFormationClient实例
LakeFormationClient client = LakeFormationClient.newBuilder()
.withHttpConfig(config)
.withCredential(basicCredentials)
.withEndpoints(endpoints)
.build();

// 3.创建请求, 添加参数
ListCatalogsRequest listCatalogsRequest =
new ListCatalogsRequest().withInstanceId("{*****your instance id*****}");

// 4.查询Catalogs列表
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 调测程序

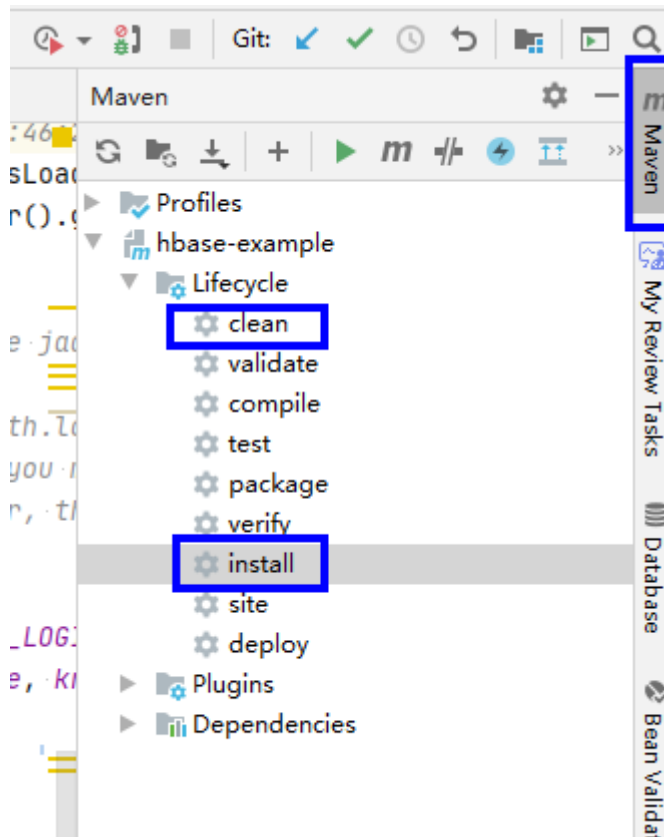
配置完样例代码后，参考该章节操作导出Jar包并将jar包上传到接入LakeFormation客户端所在节点。

步骤1 导出Jar包。

选择“Maven > *样例工程名称* > Lifecycle > clean”，双击“clean”运行maven的clean命令。

选择“Maven > *样例工程名称* > Lifecycle > install”，双击“install”运行maven的install命令。

图 3-10 maven 工具 clean 和 install



步骤2 使用maven命令打包后，执行以下命令将jar包上传到接入LakeFormation客户端所在节点。

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

例如执行后结果如下：

```
root@performance-test ~# java -cp lakeformation-lakecat-opensource-1.0.0-jar-with-dependencies.jar com.huawei.cloud.dalf.lakecat.examples.LakeFormationExample
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/manual.html#staticLoggerBinder for further details.
{
  "body": {
    "class ListCatalogsResponse {
      body: {class Catalog {
        catalogName: hive
        description: Default catalog, for Hive sa
        location: obs://lakeformation-
        databaseLocationList: null
      }, class Catalog {
        catalogName: test
        description:
        location: obs://
        databaseLocationList: null
      }, class Catalog {
        catalogName: x
        description:
        location: obs://
        databaseLocationList: null
      }
    }
  }
}
```

----结束