

工业数字模型驱动引擎(iDME)

SDK 参考

文档版本 01
发布日期 2024-03-31



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。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 服务端 SDK	1
1.1 服务端 SDK 概述.....	1
2 客户端 SDK	2
2.1 客户端 SDK 概述.....	2
2.2 Java.....	2
2.2.1 准备环境.....	2
2.2.2 获取与安装.....	3
2.2.3 本地应用注入 iDME 注解示例.....	6
2.2.4 创建数据实例示例.....	7
2.2.5 查询某实体的实例数据示例.....	8
2.3 技术支持渠道.....	9
2.4 修订记录.....	9

1 服务端 SDK

1.1 服务端 SDK 概述

本文介绍了工业数字模型驱动引擎（Industrial Digital Model Engine，简称iDME）提供的SDK语言版本，列举了最新版本SDK的获取地址。

在线生成 SDK 代码

[API Explorer](#)能根据需要动态生成SDK代码功能，降低您使用SDK的难度，推荐使用。

SDK 列表

提供了iDME支持的SDK列表，您可以在GitHub仓库查看SDK更新历史、获取安装包以及查看指导文档。

表 1-1 SDK 列表

编程语言	Github地址	参考文档
Java	huaweicloud-sdk-java-v3	Java SDK使用指导
Python	huaweicloud-sdk-python-v3	Python SDK使用指导
Go	huaweicloud-sdk-go-v3	Go SDK使用指导

2 客户端 SDK

2.1 客户端 SDK 概述

工业数字模型驱动引擎软件开发工具包（iDME SDK, Industrial Digital Model Engine Software Development Kit）是对数据建模引擎（xDM Foundation, 简称 xDM-F）提供的REST API进行封装，以简化用户的开发工作。用户直接调用iDME SDK提供的接口函数即可实现使用iDME服务业务能力的目的。

相关开发包请从[应用设计态](#)获取。

SDK 列表

下表提供了iDME支持的主流语言SDK。其中，开发指南详细介绍了SDK的安装、开发操作、参数、示例和常见问题处理。

表 2-1 SDK 列表

语言	开发指南
Java	Java SDK开发指南

2.2 Java

2.2.1 准备环境

iDME 环境

步骤1 注册云服务账号。

使用工业数字模型驱动引擎（Industrial Digital Model Engine, 简称iDME）之前必须要有一个云服务账号，具体操作请参见[注册华为账号并开通华为云](#)。

步骤2 实名认证。

具体操作请参见[实名认证](#)。

步骤3 开通iDME服务。

使用iDME之前必须已[开通iDME](#)，才能正常使用iDME。

----结束

开发环境

须知

开发过程中，您有任何问题可以在[华为云工业数字模型驱动引擎论坛](#)中发帖求助。

- 从[Oracle官网](#)下载并安装推荐使用的JDK版本。
推荐使用的JDK版本：JDK 8 以上版本。
- 根据开发需要，下载并安装开发工具。
 - 从[Eclipse官网](#)下载并安装Eclipse IDE for Java Developers最新版本。
 - 从[IntelliJ IDEA官网](#)下载并安装IntelliJ IDEA开发工具。

2.2.2 获取与安装

SDK 源码和 API 文档

- iDME Java SDK最新版本源码：请参阅[应用发布历史](#)下载最新版本源码。
- iDME Java SDK历史版本下载地址：请参阅[应用发布历史](#)下载历史版本源码。
- iDME Java SDK API文档：请参阅[全量数据服务](#)了解设计态各API的基本信息、请求参数、响应参数等信息。

📖 说明

- 为了避免不必要的繁琐代码排查与程序定位问题，如果您初次使用iDME或者数据量不大，推荐使用更加友好、图形化界面的[数据建模引擎](#)。
- iDME Java SDK中提供如下JAR包：
 - core-sdk-api-1.0.0-SNAPSHOT：包含接口的入参、出参、工具类等。
 - rdm-common-1.0.0-SNAPSHOT：包含数据建模引擎（xDM Foundation，简称xDM-F）的公共类。
 - rdm-delegate-1.0.0-SNAPSHOT：包含代理请求iDME应用运行态的封装类。
 - *应用名称*.api-1.0.0-optimization-SNAPSHOT：包含用户模型生成的代码（客户端），例如entity、DTO、service、delegator等相关代码。
 - *应用名称*.impl-1.0.0-optimization-SNAPSHOT：包含用户模型实现的功能代码（服务端）。如果您已在华为云上部署应用，生成iDME应用运行态，安装SDK时需忽略此JAR包。
 - *应用名称*.controller.api-1.0.0-*{应用环境标识}*-SNAPSHOT：包含用户模型生成的controller API代码（客户端）。
 - *应用名称*.controller.impl-1.0.0-*{应用环境标识}*-SNAPSHOT：包含用户模型生成的controller实现类代码（客户端）。如果您已在华为云上部署应用，生成iDME应用运行态，安装SDK时需忽略此JAR包。

安装 SDK

须知

开发过程中，您有任何问题可以在[华为云工业数字模型驱动引擎论坛](#)中发帖求助。

步骤1 解压缩下载的Java SDK。

步骤2 将解压缩后的JAR包引入到已有的Maven工程中。

以IntelliJ IDEA开发工具为例。

1. 在“resources”目录下创建一个“lib”目录，并将JAR包存放至该目录下。
2. 单击“File > Project Structure”。
3. 在弹出的窗口中，选择“Modules > Dependencies”，单击“+ > JARs or Directories...”。
4. 选中引入的JAR包，单击“OK”。
5. 单击“Apply”。

步骤3 在pom.xml文件中添加依赖。

在“<dependencies>”节点中加入如下配置：

```
<!--dme core-sdk-api jar包依赖-->
<dependency>
  <groupId>com.huawei.isc.xdm-f</groupId>
  <artifactId>core-sdk-api</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/src/main/resources/lib/core-sdk-api-1.0.0-SNAPSHOT.jar</systemPath>
</dependency>
<!--dme rdm-common jar包依赖-->
<dependency>
  <groupId>com.huawei.isc.xdm-f</groupId>
  <artifactId>rdm-common</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/src/main/resources/lib/rdm-common-1.0.0-SNAPSHOT.jar</systemPath>
</dependency>
<!--dme rdm-delegate jar包依赖-->
<dependency>
  <groupId>com.huawei.isc.xdm-f</groupId>
  <artifactId>rdm-delegate</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/src/main/resources/lib/rdm-delegate-1.0.0-SNAPSHOT.jar</systemPath>
</dependency>
<!--dme 应用名称.api jar包依赖-->
<dependency>
  <groupId>com.huawei.isc.xdm-f</groupId>
  <artifactId>publishtest.api</artifactId>
  <version>1.0.0-optimization-SNAPSHOT</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/src/main/resources/lib/publishtest.api-1.0.0-optimization-SNAPSHOT.jar</systemPath>
</dependency>
<!--dme 应用名称.impl jar包依赖-->
<!--如果您已在本部署DME，可使用下方代码引用应用名称.impl jar包-->
<!--<dependency-->
<!-- <groupId>com.huawei.isc.xdm-f</groupId-->
<!-- <artifactId>publishtest.impl</artifactId-->
<!-- <version>1.0.0-optimization-SNAPSHOT</version-->
```

```
<!-- <scope>system</scope>-->
<!-- <systemPath>${project.basedir}/src/main/resources/lib/publish-test.impl-1.0.0-optimization-
SNAPSHOT.jar</systemPath>-->
<!--</dependency>-->
<!--jsr validation-api包依赖-->
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>2.0.1.Final</version>
</dependency>
<!--apache httpClient包依赖-->
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpClient</artifactId>
  <version>4.5.13</version>
</dependency>
<!--mikesamuel json-sanitizer包依赖-->
<dependency>
  <groupId>com.mikesamuel</groupId>
  <artifactId>json-sanitizer</artifactId>
  <version>1.2.3</version>
</dependency>
<!--hibernate hibernate-core包依赖-->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.6.9.Final</version>
</dependency>
```

步骤4 在application.yml文件中添加delegate信息。

```
delegate:
  subAppId: rdm_e22c66fb1d05453fa*****e3cc9c0_app
  domain: https://dme.cn-north-4.huaweicloud.com
  userName: DME_validation
  password: *****
  endpoint: https://iam.cn-north-4.myhuaweicloud.com
  domainName: hwstaff_IDT_SRE
  regionName: cn-north-4
  serviceType: services/dynamic
  ssl: true
  trustStore: classpath:uat.jks
  trustStorePassword: *****
```

- subAppId: rdm_应用ID_app。
- domain: iDME服务所在的域名或IP。
- userName: IAM用户名。
- password: IAM用户密码。
- endpoint: IAM用户所在的域名或IP。
- domainName: 租户名/原华为账号。
- regionName: 当前iDME服务所在的地域。
- serviceType: URI格式的配置，默认为“publicservices”。如果您购买的数据建模引擎为基础版，则需要将“publicservices”修改为“services/dynamic”。
- ssl: 是否开启SSL认证。默认为false，不开启SSL认证。如果开启SSL认证，则需要配置trustStore和trustStorePassword参数。
- trustStore: SSL证书的安装路径。
- trustStorePassword: SSL证书的私钥密码。

----结束

2.2.3 本地应用注入 iDME 注解示例

示例场景

对于拥有本地应用的用户，在本地安装SDK后，只需要在启动类中注入 `@ComponentScan(basePackages = {"com.huawei.innovation"})` 注解，运行启动类即可。

操作步骤

在启动类上方添加 `@ComponentScan(basePackages = {"com.huawei.innovation"})` 注解，引入 iDME 的 JAR 包。

```
package com.singzai;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan(basePackages = {"com.huawei.innovation"})
public class MainApplication{
    public static void main(String[] args) { SpringApplication.run(MainApplication.class, args); }
}
```

异常处理

在启动类注入 iDME 注解并运行启动类时，本地调试可能会提示如下错误：

```
java.security.cert.CertificateException: No subject alternative DNS name matching xxx found
```

您可以参考如下操作进行修复。

1. 在项目中创建一个类，并添加 `RestTemplateBuilder` 和 `RestTemplate` 方法。
2. 分别在 `RestTemplateBuilder` 和 `RestTemplate` 方法上注入 `bean`。

例如，您在“src/main/java/com/xdm/configuration/”路径下创建“RestTemplateConfiguration.java”，其示例代码如下：

```
package com.xdm.configuration;

import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.springframework.boot.web.client.RestTemplateBuilder;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.client.HttpComponentsClientHttpRequestFactory;
import org.springframework.web.client.RestTemplate;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.cert.X509Certificate;

@Configuration
public class RestTemplateConfiguration {

    @Bean
    public RestTemplateBuilder restTemplateBuilder() {
        return new RestTemplateBuilder();
    }
}
```

```
@Bean
public RestTemplate restTemplate(RestTemplateBuilder builder) throws
NoSuchAlgorithmException, KeyManagementException {

    TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                return new X509Certificate[0];
            }
            public void checkClientTrusted(
                java.security.cert.X509Certificate[] certs, String authType) {
            }
            public void checkServerTrusted(
                java.security.cert.X509Certificate[] certs, String authType) {
            }
        }
    };
    SSLContext sslContext = SSLContext.getInstance("SSL");
    sslContext.init(null, trustAllCerts, new java.security.SecureRandom());
    CloseableHttpClient httpClient = HttpClients.custom()
        .setSSLContext(sslContext)
        .setSSLHostnameVerifier(NoopHostnameVerifier.INSTANCE)
        .build();
    HttpComponentsClientHttpRequestFactory customRequestFactory = new
HttpComponentsClientHttpRequestFactory();
    customRequestFactory.setHttpClient(httpClient);
    return builder.requestFactory() -> customRequestFactory).build();
}
}
```

3. 在操作步骤添加的@ComponentScan注解中，添加1创建的类的扫描路径。
package com.singzai;

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan(basePackages = {"com.huawei.innovation", "com.xdm"})
public class MainApplication{
    public static void main(String[] args) { SpringApplication.run(MainApplication.class, args); }
}
```

2.2.4 创建数据实例示例

示例场景

本文对使用SDK方式创建数据实例进行示例说明。

使用SDK时，需要在代码中注入Delegator和RestTemplate。下面以创建XZTEST数据实体的数据实例为例。

操作步骤

在代码中注入Delegator和RestTemplate，并调用create方法创建数据实例。

```
import com.huawei.innovation.rdm.publishstest.delegator.XZTESTDelegator;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.web.client.RestTemplate;

@SpringBootTest
public class DMETest {

    @Autowired
```

```
XZTESTDelegator xztestDelegator;  
@Autowired  
RestTemplate restTemplate;  
  
@Test  
public void createTest() {  
    TenantUtil.setTenantId(444082253783236608L); // 设置当前操作的租户ID  
    XZTESTCreateDTO createDTO = new XZTESTCreateDTO(); // 创建XZTEST数据实体的CreateDTO  
    createDTO.setUid("123456");  
    createDTO.setUvalue("abc");  
    createDTO.setXzdate(new Timestamp(1672888888000L));  
    System.out.println(createDTO);  
    XZTESTViewDTO viewDTO = xztestDelegator.create(createDTO); //采用实体代理类  
    (xztestDelegator), 调用create方法创建实例  
    TenantUtil.clearTenantId(); // 清除租户信息  
    System.out.println(viewDTO); //打印实例的所有信息  
}  
}
```

2.2.5 查询某实体的实例数据示例

示例场景

本文对使用SDK方式查询某实体所有实例数据进行示例说明。

使用SDK时，需要在代码中注入Delegator和RestTemplate。下面以查询XZTEST数据实体的指定实例数据为例。

操作步骤

在代码中注入Delegator和RestTemplate，并调用find方法查询UID含有“123”的数据实例。

```
import com.huawei.innovation.rdm.publishtest.delegator.XZTESTDelegator;  
import org.junit.jupiter.api.Test;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.web.client.RestTemplate;  
  
import java.util.List;  
  
@SpringBootTest  
public class DMETest {  
  
    @Autowired  
    XZTESTDelegator xztestDelegator;  
    @Autowired  
    RestTemplate restTemplate;  
  
    @Test  
    public void findTest() {  
        QueryRequestVo queryRequestVo = new QueryRequestVo();  
        QueryCondition queryCondition = new QueryCondition();  
        queryRequestVo.setFilter(queryCondition);  
        List<String> values = new ArrayList<>();  
        values.add("123");  
        queryCondition.setConditionValues(values); // 设置查询条件: where uid like 123  
        queryCondition.setConditionName("uid");  
        queryCondition.setOperator("like");  
        RDMPageVO rdmPageVO = new RDMPageVO(); // 分页查询  
        rdmPageVO.setCurPage(1);  
        rdmPageVO.setPageSize(20);  
        List<XZTESTViewDTO> xztestViewDTOS = xztestDelegator.find(queryRequestVo, rdmPageVO);  
        System.out.println(xztestViewDTOS); //打印查询结果  
    }  
}
```

2.3 技术支持渠道

您可以通过关键字在左侧搜索框中搜索本产品的文档，帮助您快速找到需要的文档。

开发者社区提供的技术支持渠道如下：

开发过程中，您有任何问题可以在[华为云工业数字模型驱动引擎论坛](#)中发帖求助。

2.4 修订记录

发布日期	修改说明
2024-02-23	第四次正式发布。 修改 获取与安装
2023-07-22	第三次正式发布。 修改 准备环境
2023-06-17	第二次正式发布。 修改 准备环境
2023-05-20	第一次正式发布。