工业数字模型驱动引擎(iDME)

开发指南

文档版本 04

发布日期 2023-10-29

华为云计算技术有限公司

商标声明


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 简介

概述

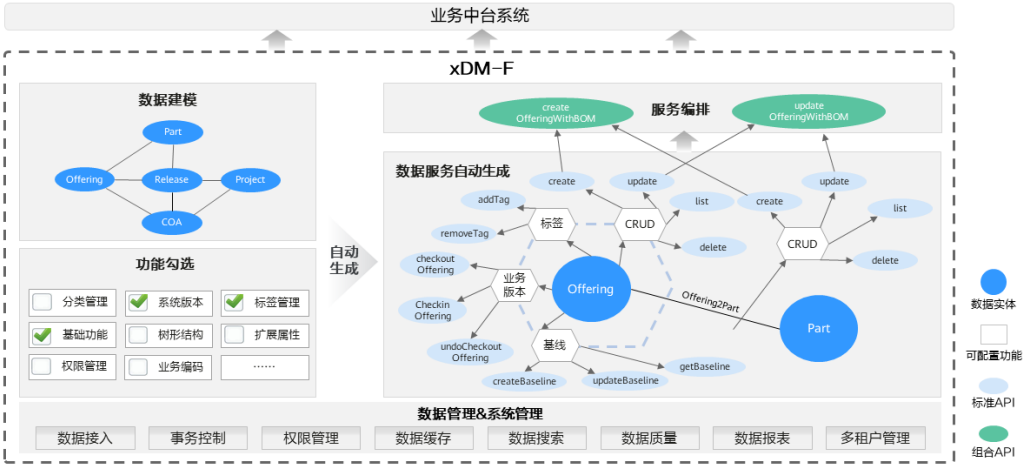
随着企业数字化转型的浪潮不断来袭，众多非数字原生企业的多年信息化建设，除了给企业带来了成长的红利，也给企业带来了数据管理和消费的历史包袱。当前，国内企业业务持续的风险不断升高，现有系统面临业务数据获取周期长、业务数据转换效率低、升级替换难等挑战，亟须破局。工业数字模型驱动引擎（Industrial Digital Model Engine，简称iDME），是基于数字化变革和数据管理优秀实践开发的创新型工业软件开发与运营平台，打造“基于模型+数据驱动”的公共底座，为协同打造新一代全栈自主可控工业软件体系提供根技术服务，助力快速构建和运营云化SaaS化工业软件，汇聚企业全场景数据，构建企业级数据图谱，提供万数互联的数据索引、追溯、交互服务。

开发指南从iDME功能实现的角度，介绍iDME的功能、使用方法、样例工程及代码介绍，提供给需要进行业务应用开发的并具备Java开发经验的开发人员使用。

xDM-F 架构

您可以华为云上管理和使用工业数字模型驱动引擎-数据建模引擎（xDM Foundation，简称xDM-F）服务，也可以将xDM-F集成到您自己的应用中。

图 1-1 xDM-F 架构



功能使用

表 1-1 xDM-F 功能使用

功能	描述	使用方式
基础数据服务	针对数据实体、关系实体、接口模型等自动生成数据服务，例如生成增、删、改、查、分页查询等原子接口，实现基础服务的调用。	• 控制台 • API
文件服务	可以在数据实例中上传、下载多种格式文件，可满足文件管理的诉求。	• 控制台 • API
安全受控	安全受控功能可以高安全、高可靠的管控数据访问权限及业务关键信息。	• 控制台 • API
主版本服务	可以对数据对象进行检出、修订、检入操作，为M-V模型数据对象中master提供父类。	• 控制台 • API
版本服务	可以为多版本对象提供M-V模型原子服务，用于业务对象追溯管理，提高追溯效率。	• 控制台 • API
系统版本	可以通过版本号标识记录修改历史记录。	• 控制台 • API
树形结构	可以指定全路径字段，为树形节点全路径绑定属性。	• 控制台 • API
权限管理	可以根据自身业务需求，对数据或功能进行授权鉴权的开发。	• 控制台 • API
业务编码生成器	业务编码生成器功能可以自定义规则，自动生成业务编码，为系统维护者提供便利。	• 控制台 • API
文件夹管理	可创建多层级的文件夹结构，在创建数据实例时可指定所属文件夹，方便对数据进行分门别类。	• 控制台 • API
分类管理	可以根据分类节点及分类属性快速检索对象，节约时间成本、提高查询效率和业务效率。	• 控制台 • API
扩展属性	可以为数据对象添加扩展属性，用于租户的定制化扩展业务。	• 控制台 • API
扩展类型	可以灵活定义对象的扩展类型，用于标识数据对象是否可被扩展。	• 控制台 • API

功能	描述	使用方式
生命周期管理	可以通过生命周期模板定义及管理，实现数据对象的全过程管理，以及对不同类型业务对象的过程追溯。	● 控制台 ● API
失效管理	可以灵活管理用户权限、业务属性（如零部件的损坏），便于用户识别。	● 控制台 ● API
所有者管理	可以通过所有者属性为数据对象授权。	● 控制台 ● API
标签管理	可以对已有数据对象从不同维度进行规划和分类管理。	● 控制台 ● API
多维视图&多维分支	可以通过不同视角（如设计、工艺、制造、服务等）对同一个数据对象（如BOM编码）进行差异化管理。	API **说明** xDM-F只能对M-V模型数据实体设置多维视图&多维分支能力，只支持通过API方式创建多维视图&多维分支数据实例。

2 准备工作

资源环境要求

- 已**开通iDME设计服务**。
- 已**购买数据建模引擎**。

权限要求

使用iDME时，需要对用户进行授权。建议您通过IAM用户使用iDME服务。IAM用户可以帮助您安全的控制华为云资源的访问。

具体操作请参见**创建iDME操作用户**和**用户管理**。

技能要求

- 了解iDME的基本概念、使用场景、使用方式等。
- 熟悉Java语言，能够编写Java语言代码。
- 熟悉Maven构建方式和使用方法。

开发环境要求

- 已安装JDK 8+。
- 已安装Java语言的开发工具。

3 构建 iDME 应用

应用即一个App，是一个实现了某种业务管理的可运行应用程序。创建应用是在iDME中开发项目的第一步，也是端到端构建软件应用的入口。在开发项目工程前，请先创建一个应用，再在应用设计态中创建数据模型，在应用运行态使用数据服务。

操作步骤

步骤1 创建应用，用于对软件应用的设计和构建。

iDME提供控制台和API两种创建方式，具体操作请参见**创建应用（控制台）**和**创建应用（API）**。

步骤2 在应用设计态构建并发布应用。

- 构建应用：对业务规则进行设计和开发，具体操作请参见**数据建模引擎用户指南**。
- 发布应用：生成相应代码包，具体操作请参见**发布应用**。

步骤3 部署应用，生成应用运行态，用于多租户应用集成测试及数据服务调用。

iDME提供控制台和API两种部署方式，具体操作请参见**部署应用（控制台）**和**部署应用（API）**。

----结束

下一步操作

当您完成应用构建后，即可在应用运行态对设计态的数据模型进行全生命周期管理和控制，或根据用户的定制化需求进行模型扩展、搜索服务定义、服务编排等操作。常用场景包括：

- **使用全量数据服务API**
- **使用生命周期管理**
- **使用服务编排管理**
- **使用搜索服务管理**

4 使用全量数据服务 API

4.1 全量数据服务概述

为提升应用开发人员的开发效率，工业数字模型驱动引擎-数据建模引擎（xDM Foundation，简称xDM-F）提供对应用全量数据服务的管理功能，支持查看所有可调用API的信息，支持调用应用部署后的所有API。全量数据服务API包括系统管理API、数据模型生成API、定制API的基本信息、参数信息和示例。

应用开发人员可以在应用运行态的全量数据服务快速获取应用下所有类型API的清单、**查看API详情**、**导出API文档**，提升模型的消费速度和API调用速度，快速使能工业应用。

4.2 创建指定实体的实例

功能介绍

在应用设计态完成模型的构建、发布（模型发布和应用发布），以及控制台部署应用后，会在应用运行态自动生成相应的CRUD接口。其中，create接口用于创建模型实例。

更多API信息请参见**全量数据服务**。

URI

- URI格式：

  POST http://**{Endpoint}**/rdm_**{appID}**_app/services/dynamic/api/**{entityName}**/create

- 参数说明：

表 4-1 URI 参数说明

参数	是否必填	参数类型	描述
Endpoint	是	String	承载REST服务端点的服务器域名或IP地址。
appID	是	String	应用ID。
entityName	是	String	实体的英文名称。

请求参数

表 4-2 请求 body 参数

参数	是否必填	参数类型	描述
id	是	Int	唯一编码。
rdmVersion	是	Int	系统版本。
rdmExtensionType	否	String	实体类型。
createTime	否	Date	创建时间。
creator	否	String	创建者。
lastUpdateTime	否	Date	最后更新时间。
modifier	否	String	更新者。
tenant	否	Object	租户。

根据实体类型、功能的不同，请求body参数不同，您可以在**全量数据服务**查看API具体参数。

响应参数

返回创建实例的详细信息。

请求示例

示例背景

假设您已在设计态构建一个名称为“Design_document”的数据实体，并完成了实体发布、应用发布和应用部署。根据运行态的“数据服务管理 > 全量数据服务”查看到的“Design_document”数据实体Create接口信息，传入相应的参数。

示例代码

```
{
    "params": {
```

```
        "clsAttrs": null,
        "department": null,
        "income": null,
        "securityLevel": "internal",
        "kiaguid": null,
        "name": "产品设计",
        "description": null,
        "tenant": {
            "id": "-1",
            "clazz": "Tenant"
        },
        "rdmExtensionType": null,
        "id": null
    }
}
```

响应示例

```
{
    "result": "SUCCESS",
    "data": [
        {
            "id": "502903940749205504",
            "creator": "TestAccount2 5547b9adee54423cb**********",
            "modifier": "TestAccount2 5547b9adee54423cb**********",
            "createTime": "2023-05-24T10:00:42.978+0000",
            "lastUpdateTime": "2023-05-24T10:00:42.978+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "Design_document",
            "tenant": {
                "id": "-1",
                "creator": "xdmAdmin",
                "modifier": "xdmAdmin",
                "createTime": "2022-12-01T11:24:39.000+0000",
                "lastUpdateTime": "2022-12-01T11:24:39.000+0000",
                "rdmVersion": 1,
                "rdmDeleteFlag": 0,
                "rdmExtensionType": "Tenant",
                "tenant": null,
                "className": "Tenant",
                "name": "basicTenant",
                "description": "默认租户",
                "kiaguid": null,
                "securityLevel": "internal",
                "code": "basicTenant",
                "disableFlag": false,
                "dataSource": "DefaultDataSource"
            },
            "className": "Design_document",
            "name": "产品设计",
            "description": null,
            "kiaguid": null,
            "securityLevel": "internal",
            "income": null,
            "department": null
        }
    ],
    "errors": []
}
```

4.3 查询满足条件的实例数据

功能介绍

在应用设计态完成模型的构建、发布（模型发布和应用发布），以及控制台部署应用后，会在应用运行态自动生成相应的CRUD接口。其中，find接口用于分页查询实例信息。

更多API信息请参见**全量数据服务**。

URI

- URI格式：

  POST http://**{Endpoint}**/rdm_**{appID}**_app/services/dynamic/api/**{entityName}**/find/**{pageSize}**/**{curPage}**

- 参数说明：

  表 4-3 URI 参数说明

参数	是否必填	参数类型	描述
Endpoint	是	String	承载REST服务端点的服务器域名或IP地址。
appID	是	String	应用ID。
entityName	是	String	实体的英文名称。
pageSize	否	String	分页查询时，每页最多展示的记录数。
curPage	否	String	分页查询的页数。

请求参数

根据实体类型、功能的不同，请求body参数不同，您可以在**全量数据服务**查看API具体参数。为篇幅起见，这里只展示部分内容。

表 4-4 请求 body 参数

参数	是否必填	参数类型	描述
sorts	否	List	排序。
sort	否	String	排序方式（asc/desc）。

参数	是否必填	参数类型	描述
orderBy	否	String	排序字段，可填写模型自身属性、参考对象的属性、扩展属性及分类属性。
filter	否	Object	查询条件。
isNeedTotal	否	Boolean	是否需要查询总记录数（false/true）。

响应参数

返回模型所有属性、直接关联的参考对象、扩展属性、分类属性、级联的数据等。

请求示例

示例背景

- 已在设计态构建一个名称为“employee”的数据实体，该实体自定义了“age”和“sex”两个属性，并完成了实体发布、应用发布和应用部署。
- 已在运行态通过控制台或API的方式创建多个数据实例。

需要查询“sex”为“男性”的数据实例，且结果需要先按名称倒序排序，再按年龄倒序排序。

示例代码

```
{
   "params": {
      "sorts": [
         {
            "sort": "DESC",
            "orderBy": "name"
         },
         {
            "sort": "DESC",
            "orderBy": "age"
         }
      ],
      "filter": {
         "joiner": "and",
         "conditions": [
            {
               "conditionName": "sex",
               "operator": "=",
               "conditionValues": [
                  "男"
               ]
            }
         ]
      },
      "isNeedTotal": true
   }
}
```

响应示例


```
{
   "result": "SUCCESS",
   "data": [
      {
         "id": "455304645330341888",
         "creator": "test1 3c03e719256a427eb9277b64fcXXXXXX",
         "createTime": "2023-01-13T01:38:07.000+00:00",
         "modifier": "test1 3c03e719256a427eb9277b64fcXXXXXX",
         "lastUpdateTime": "2023-01-13T01:38:07.000+00:00",
         "rdmVersion": 1,
         "rdmExtensionType": "People",
         "rdmDeleteFlag": 0,
         "tenant": {
            "id": "-1",
            "clazz": "Tenant"
         },
         "className": "People",
         "name": "李四",
         "description": null,
         "kiaguid": null,
         "securityLevel": "internal",
         "sex": "男",
         "age": 20
      },
      {
         "id": "455304534248394752",
         "creator": "test1 3c03e719256a427eb9277b64fcXXXXXX",
         "createTime": "2023-01-13T01:37:40.000+00:00",
         "modifier": "test1 3c03e719256a427eb9277b64fcXXXXXX",
         "lastUpdateTime": "2023-01-13T01:37:40.000+00:00",
         "rdmVersion": 1,
         "rdmExtensionType": "People",
         "rdmDeleteFlag": 0,
         "tenant": {
            "id": "-1",
            "clazz": "Tenant"
         },
         "className": "People",
         "name": "张三",
         "description": null,
         "kiaguid": null,
         "securityLevel": "internal",
         "sex": "男",
         "age": 18
      }
   ],
   "errors": [],
   "pageInfo": {
      "curPage": 1,
      "pageSize": 20,
      "totalRows": 2,
      "totalPages": 1
   }
}
```


4.4 分页查询指定用户的实例数据

操作场景

本文介绍如何通过鉴权接口和分页查询接口实现数据对象的权限过滤。

假设，在cn-north-4区域创建一个应用（ TestApp ），应用ID为01a2b2c4764d4e00f123g345fd9baa9f，且已部署至数据建模引擎。应用中存在一个

数据实体（DataEntity1），现在需要分页查询该数据实体的所有实例数据，并过滤指定用户权限的实例数据。

URI

- Find接口

  URL格式：POST http://**{Endpoint}**/rdm_**{appID}**_app/publicservices/api/**{entityName}**/find/**{pageSize}**/**{curPage}**

- batchHasAccess接口

  URL格式：POST http://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/AccessService/batchHasAccess

- 参数说明：

  表 4-5 URI 参数说明

参数	是否必填	参数类型	描述
Endpoint	是	String	承载REST服务端点的服务器域名或IP地址。
appID	是	String	应用ID。
entityName	是	String	实体的英文名称。
pageSize	否	String	分页查询时，每页最多展示的记录数。
curPage	否	String	分页查询的页数。

步骤 1：分页查询 DataEntity1 实体的所有实例数据

步骤1 在Request Header中增加“X-Auth-Token”，值为用户Token。

步骤2 在Request Header中增加“Content-Type”，值为“application/json”。

步骤3 在Request Body中传入参数如下：

```
{
   "params":{
      "sort":"DESC",
      "orderBy":"lastUpdateTime",
      "filter": {
         "joiner":"and",
              "conditions":[]
      },
      "isNeedTotal":false
   }
}
```

- sort：排序类型，也可为空。
- orderBy：填写需要按哪个字段进行排序，可填写模型自身属性、参考对象的属性、扩展属性及分类属性，也可为空。

- filter：填写过滤条件，可为空。

步骤4 发送POST“https://dme.cn-north-4.huaweicloud.com/rdm_01a2b2c4764d4e00f123g345fd9baa9f_app/publicservices/api/DataEntity1/find/20/1”。

若请求失败，会返回错误码及对应的错误信息说明。

----结束

步骤 2：基于分页查询的实例数据，指定用户过滤实例数据

步骤1 在Request Header中增加“X-Auth-Token”，值为用户Token。

步骤2 在Request Header中增加“Content-Type”，值为“application/json”。

步骤3 在Request Body中传入参数如下：

```
{
   "params":{
      "userId":"490183140267008000",
      "rdmExtensionType":" DataEntity1",
      "operations":["000000005"],
      "ids":["490930153786974208", "23213", "20230414300", "9898989898989898"]
   }
}
```

- userId：被鉴权用户在XDMUser实体实例中的ID，不可为空。
- rdmExtensionType：被鉴权实例的实体类型，不可为空。
- operations：操作列表，不可为空。
- ids：被鉴权的实例ID列表，不可为空。

步骤4 发送POST“https://dme.cn-north-4.huaweicloud.com/rdm_01a2b2c4764d4e00f123g345fd9baa9f_app/services/rdm/basic/api/AccessService/batchHasAccess”。

若请求失败，会返回错误码及对应的错误信息说明。

----结束

4.5 比较 M-V 模型的版本对象

操作场景

在应用设计态完成模型的构建、发布（模型发布和应用发布），以及控制台部署应用后，会在应用运行态自动生成相应的接口。本文介绍如何通过版本对象比较接口（compareBusinessVersion）实现比较不同版本对象间的基本属性、扩展属性和关联关系。

URI

- URL格式：

  POST http://**{Endpoint}**/rdm_**{appID}**_app/publicservices/api/**{entityName}**/compareBusinessVersion

- 参数说明：

表 4-6 URI 参数说明

参数	是否必填	参数类型	描述
Endpoint	是	String	承载REST服务端点的服务器域名或IP地址。
appID	是	String	应用ID。
entityName	是	String	实体的英文名称。

请求参数

表 4-7 请求 body 参数

参数	是否必填	参数类型	描述
id	是	Int	主对象ID，用于标识版本对象。
basicVersion	是	String	基础版本对象的版本号。
correlationVersion	是	String	待比较的版本对象的版本号。

为篇幅起见，这里只展示部分内容。更多参数信息，您可以在**全量数据服务**进行查看。

响应参数

返回两个版本对象的对比结果。

请求示例

示例背景

假设，您已在cn-north-4区域的TestApp应用中，构建一个名称为“TestMV”的M-V模型数据实体，并完成了实体发布、应用发布和应用部署。希望对主对象ID为“492060584863342592”的A.1版本和A.2版本进行比较。

示例代码

```
{
   "params":{
      "id":492060584863342592,
      "basicVersion":"A.1",
      "correlationVersion":"A.2"
   }
}
```

响应示例

为篇幅起见，这里只展示部分内容。

```
{
   "result": "SUCCESS",
   "data": [
      {
        ......    //返回基础版本对象版本号A.1的所有内容
      },
      {
         ......    //返回待比较版本对象版本号A.2的基本属性的区别
         "relations": [
      ......    //返回待比较版本对象版本号A.2的关联关系的区别（所有区别都是A.2与A.1对比后A.2的区别）。如需对比A.1的区别，传参时调换A.1和A.2的顺序，将A.2作为基础版本对象版本号，A.1作为待比较的版本对象的版本号即可。
         ],
         "extAttrs": {
            ......    //返回待比较版本对象版本号A.2的扩展属性的区别
         },
         "latest": true,
         "lastUpdateTime": "2023-04-27 14:35:56"
      }
   ],
   "errors": []
}
```

4.6 另存版本对象的实例数据

功能介绍

版本对象的另存为接口（saveAs）用于创建一条与原版本对象实例数据相同的数据实例。该实例数据会完全复制原实例现有的数据，包括与其关联的主对象和分支对象，且新实例数据的版本号从初始值开始计算。

URI

- URL格式：

  POST http://**{Endpoint}**/rdm_**{appID}**_app/publicservices/api/**{entityName}**/saveAs

- 参数说明：

  表 4-8 URI 参数说明

参数	是否必填	参数类型	描述
Endpoint	是	String	承载REST服务端点的服务器域名或IP地址。
appID	是	String	应用ID。
entityName	是	String	实体的英文名称。

请求参数

表 4-9 请求 body 参数

参数	是否必填	参数类型	描述
sourceInstanceId	是	Int	version.唯一编码。
sourceMasterId	是	String	master.唯一编码。

传入参数时，必填参数需至少传入一个，即至少传入sourceInstanceId参数或sourceMasterId参数。为篇幅起见，这里只展示部分内容。更多参数信息，您可以在**全量数据服务**进行查看。

响应参数

返回创建的数据实例信息。

请求示例

示例背景

假设，您已在cn-north-4区域的TestApp应用中，构建一个名称为“TestMV”的M-V模型数据实体，并完成了实体发布、应用发布和应用部署。希望可以根据指定的版本对象唯一编码（481785585908850688）或主对象唯一编码（481785532712488960），创建一个新的数据实例。

示例代码

为篇幅起见，这里只展示必填参数。

```
{
   "params":{
            ......
      "sourceInstanceId":"481785585908850688",
      "sourceMasterId":"481785532712488960"
   }
}
```

响应示例

为篇幅起见，这里只展示部分内容。

```
{
   "result": "SUCCESS",
   "data": [
      {
         "id": "515167665271083008",
         "creator": "TestAccount2 5547b9adee54423cb**********",
         "modifier": "TestAccount2 5547b9adee54423cb**********",
         "createTime": "2023-06-27T05:57:56.000+0000",
         "lastUpdateTime": "2023-06-27T06:12:22.752+0000",
         "rdmVersion": 1,
         "rdmDeleteFlag": 0,
         "rdmExtensionType": "TestMV",
         "tenant": {
            ......
         },
```

```
        "className": "TestMV",
        "master": {
            "id": "515167665275277312",
            "tenant": {
                ......
            },
            "className": "TestMVMaster"
        },
        "branch": {
            "id": "515167665313026048",
            "creator": "TestAccount2 5547b9adee54423cb**********",
            "modifier": "TestAccount2 5547b9adee54423cb**********",
            "createTime": "2023-06-27T06:12:22.774+0000",
            "lastUpdateTime": "2023-06-27T06:12:22.774+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "TestMVBranch",
            "tenant": {
                ......
            },
            "className": "TestMVBranch",
            "version": "A"
        },
        "latest": true,
        "versionCode": 1,
        "iteration": 1,
        "version": "A",
        "workingState": {
            "code": "CHECKED_IN",
            "cnName": "已检入",
            "enName": "checked in",
            "alias": "CHECKED_IN"
        },
        "checkOutUserName": null,
        "checkOutTime": null,
        "preVersionId": null
    }
  ],
  "errors": []
}
```


4.7 创建多维版本的数据实例

功能介绍

版本对象的创建视图接口（createView）和批量创建视图接口（batchCreateView）可基于已有M-V模型的数据实例创建多维版本数据实例。

本章节以createView为例，如需调用batchCreateView，请前往**全量数据服务**查看。

URI

- URL格式：

  POST http://**{Endpoint}**/rdm_**{appID}**_app/publicservices/api/**{entityName}**/createView

- 参数说明：

表 4-10 URI 参数说明

参数	是否必填	参数类型	描述
Endpoint	是	String	承载REST服务端点的服务器域名或IP地址。
appID	是	String	应用ID。
entityName	是	String	实体的英文名称。

前提条件

- 已**获取用户Token**。
- 已获取Endpoint值（数据建模引擎所在域名或IP地址）、应用ID、M-V模型的实体名称，以及涉及接口的请求参数（如versionId、workCopyType、customLinkSet、needSetNull和versionId），具体信息可前往**全量数据服务**查看。
- 基于已获取的M-V模型实体名称创建一个数据实例。

请求参数

表 4-11 请求 body 参数

参数	是否必填	参数类型	描述
versionId	是	Integer	原视图的versionId，即已创建数据实例的version.唯一编码。
workCopyType	否	Object	关系的复制类型。 • BOTH：复制作为源端与目标端的关系。 • SOURCE：复制作为源端的关系。 • TARGET：复制作为目标端的关系。 • NONE：不复制关系。 • CUSTOM：自定义复制关系。
customLinkSet	否	List	关系实体名称的集合。 当“workCopyType”设置为“CUSTOM”时，需要设置此参数。
needSetNull	否	List	指定不复制的属性。被指定不复制的属性，其返回值将被设置为“null”。
modifier	是	String	更新者。

说明

如果用户在应用设计态创建多维视图&多维分支功能的数据实体时，将“多维版本”功能配置新增的属性设置为必填，该数据实体的API请求参数中需设置相应参数。

响应参数

返回创建的多维版本数据实例。

请求示例

示例背景

假设，您已在cn-north-4区域的TestApp应用中，构建一个名称为“TestMV”的M-V模型数据实体，并完成了实体发布、应用发布和应用部署。希望可以根据已创建的M-V模型数据实例，创建一个多维版本数据实例。

示例代码

```
{
   "params": {
      "versionId": "521722330943061234",
      "modifier": "DME_Developer",
      "view2": {
         "id": "11",
         "clazz": "ViewAttr"
      }
   }
}
```

响应示例

为篇幅起见，这里只展示部分内容。

```
{
   "result": "SUCCESS",
   "data": [
      {
         "id": "523616294595981234",
         ......
         "rdmVersion": 1,
         "rdmDeleteFlag": 0,
         "rdmExtensionType": "View2",
         "tenant": {
            ......
         },
         "className": "View2",
         "name": "AS",
         "description": null,
         "kiaguid": null,
         "securityLevel": "internal",
         "master": {
            ......
            "rdmExtensionType": "View2Master",
            "tenant": {
               ......
            },
            "className": "View2Master"
         },
         "branch": {
            ......
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "View2Branch",
```

```
            "tenant": {
                ......
            },
            "className": "View2Branch",
            "version": "A"
        },
        "latest": true,
        "latestIteration": true,
        "versionCode": 1,
        "iteration": 1,
        "version": "A",
        "latestVersion": true,
        "workingCopy": false,
        "workingState": {
            "code": "CHECKED_IN",
            "cnName": "已检入",
            "enName": "checked in",
            "alias": "CHECKED_IN"
        },
        "checkOutUserName": null,
        "checkOutTime": null,
        "preVersionId": "52281****087179264",
        "viewAttr1": null,
        "viewAttr3": null,
        "viewAttr2": null
    }
  ],
  "errors": []
}
```

5 使用生命周期管理

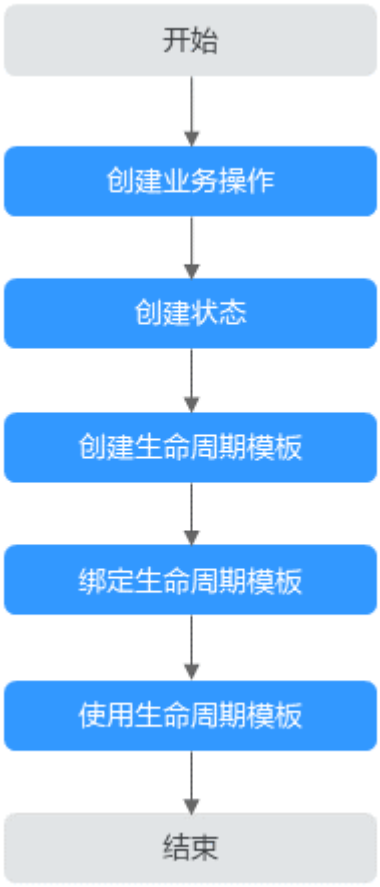
5.1 生命周期管理概述

概述

为了方便应用开发人员对不同的业务对象定义不同的生命周期模板及访问规则、方便业务设计人员跟踪/流转业务对象状态，iDME提供生命周期管理功能。用户可以通过生命周期模板定义及管理实现对象的全过程管理，实现对不同类型业务对象的过程追溯，提升系统扩展性、灵活性，降低开发的人力及时间成本；也可以根据生命周期状态定义对象的动态访问权限，实现权限的灵活定义，提升应用开发人员的体验、降低应用运维成本。

使用流程

图 5-1 生命周期管理流程



使用方法

- 使用应用运行态
  a. 根据业务对象的流程标志，依次创建业务操作，具体操作请参见**创建业务操作**。
     例如，Charter开发流程的概念决策评审点（Concept Decision Check Point，CDCP）、计划决策评审点（Plan Decision Check Point，PDCP）等。
  b. 根据业务对象的流程阶段，依次创建状态，具体操作请参见**创建状态**。
     例如，IPD（Integrated Product Development，集成产品开发）流程的概念阶段、计划阶段、开发阶段、验证阶段、发布阶段和维护阶段。
  c. 根据业务对象的生命周期操作流程，创建生命周期模板，关联相关的业务操作和状态，具体操作请参见**创建生命周期模板**。
  d. 将具有生命周期管理功能的数据实体和刚创建的生命周期模板进行绑定，具体操作请参见**修改数据实体**。
     如果应用运行态没有生命周期管理功能的数据实体，请前往应用设计态创建一个具有生命周期管理功能的数据实体，并完成“发布该数据实体 > 发布应

用 > 部署应用”的操作后，再返回应用运行态通过修改数据实体绑定生命周期模板。

e. 根据实例数据的流转状态，灵活使用生命周期模板的业务操作和状态。

- 如果您的数据实例是单实体实例，可通过编辑的方式更新实例的生命周期阶段，具体操作请参见**修改数据实例**。
- 如果您的数据实例是M-V模型实体实例，可通过检入/检出的方式更新实例的生命周期阶段，具体操作请参见**检入数据实例（M-V模型）**和**检出数据实例（M-V模型）**。

- 使用全量数据服务API

  您可以通过**全量数据服务**提供的API使用生命周期管理，具体操作请参见**5.2 生命周期管理开发指导**。

5.2 生命周期管理开发指导

本文指导您通过全量数据服务API使用生命周期管理功能。

涉及的接口

表 5-1 涉及的接口

接口名称	说明
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/LifecycleBusinessOperation/create	创建业务操作。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/LifecycleState/create	创建状态。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/LifecycleTemplate/create	创建生命周期模板。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/LifecycleTemplate/updateLifecycleInfo	为实体绑定生命周期模板。
https://**{Endpoint}**/rdm_**{appID}**_app/services/dynamic/api/**{entityName}**/update	修改数据实例，更新数据实例当前的生命周期阶段。
https://**{Endpoint}**/rdm_**{appID}**_app/services/dynamic/api/**{entityName}**/checkout	检出M-V模型的数据实例，更新数据实例当前的生命周期阶段。
https://**{Endpoint}**/rdm_**{appID}**_app/services/dynamic/api/**{entityName}**/checkin	检入M-V模型的数据实例，更新数据实例当前的生命周期阶段。

步骤 1：创建业务操作

步骤1 准备工作。

在应用运行态中获取创建业务操作的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务对象的生命周期管理流程，调用API创建业务操作。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/basic/api/LifecycleBusinessOperation/create

{
   "params": {
      "name": "发布",
      "nameEn": "Publish",
      "description": "",
      "descriptionEn": ""
   }
}
```

其中，{Endpoint}、{appID}表示数据建模引擎所在域名/IP地址和应用ID，“name”、“nameEn”、“description”、“descriptionEn”表示业务操作的中英文名称和中英文描述。

- 响应示例


```
{
   "result": "SUCCESS",
   "data": [
      {
         "id": "504753788985286656",
         "creator": "XDM_Developer 93172bbfd0f644**********345386",
         "modifier": "XDM_Developer 93172bbfd0f644**********345386",
         "createTime": "2023-05-29T12:31:21.166+0000",
         "lastUpdateTime": "2023-05-29T12:31:21.166+0000",
         "rdmVersion": 1,
         "rdmDeleteFlag": 0,
         "rdmExtensionType": "LifecycleBusinessOperation",
         "tenant": {
            "id": "-1",
            "creator": "xdmAdmin",
            "modifier": "xdmAdmin",
            "createTime": "2023-05-29T11:27:42.000+0000",
            "lastUpdateTime": "2023-05-29T11:27:42.000+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "Tenant",
            "tenant": null,
            "className": "Tenant",
            "name": "basicTenant",
            "description": "默认租户",
            "kiaguid": null,
            "securityLevel": "internal",
            "code": "basicTenant",
            "disableFlag": false,
            "dataSource": "DefaultDataSource"
         },
         "className": "LifecycleBusinessOperation",
         "businessCode": "LOP00000001",
         "description": "",
         "descriptionEn": "",
         "nameEn": "Publish",
         "name": "发布",
         "disableFlag": false
      }
   ],
```



```
    "errors": []
}
```

----结束

步骤 2：创建状态

步骤1 准备工作。

在应用运行态中获取创建状态的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务对象的生命周期管理流程，调用API创建状态。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/basic/api/LifecycleState/create

{
    "params": {
        "name": "已发布",
        "nameEn": "Published",
        "description": "",
        "descriptionEn": "",
        "internalName": "Published"
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“name”、“nameEn”、“description”、“descriptionEn”表示生命周期状态的中英文名称和中英文描述。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [
        {
            "id": "504966116662059008",
            "creator": "XDM_Developer 93172bbfd0f644**********345386",
            "modifier": "XDM_Developer 93172bbfd0f644**********345386",
            "createTime": "2023-05-30T02:35:04.029+0000",
            "lastUpdateTime": "2023-05-30T02:35:04.029+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "LifecycleState",
            "tenant": {
                "id": "-1",
                "creator": "xdmAdmin",
                "modifier": "xdmAdmin",
                "createTime": "2023-05-29T11:27:42.000+0000",
                "lastUpdateTime": "2023-05-29T11:27:42.000+0000",
                "rdmVersion": 1,
                "rdmDeleteFlag": 0,
                "rdmExtensionType": "Tenant",
                "tenant": null,
                "className": "Tenant",
                "name": "basicTenant",
                "description": "默认租户",
                "kiaguid": null,
                "securityLevel": "internal",
                "code": "basicTenant",
                "disableFlag": false,
                "dataSource": "DefaultDataSource"
            },
            "className": "LifecycleState",
            "descriptionEn": "",
            "businessCode": "LCS00000003",
            "name": "已发布",
            "description": "",
            "nameEn": "Published",
```

```
            "internalName": "Published",
            "disableFlag": false,
            "extAttrs": [],
            "extAttrMap": {}
        }
    ],
    "errors": []
}
```

----结束

步骤 3：创建生命周期模板

步骤1 准备工作。

在应用运行态中获取创建生命周期模板的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务对象的生命周期管理流程，调用API创建生命周期模板。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/basic/api/LifecycleTemplate/create

{
    "params": {
        "name": "产品发布流程",
        "nameEn": "Product Release Process",
        "description": "",
        "descriptionEn": "",
        "master": {
            "id": null,
            "category": "",
            "name": "产品发布流程",
            "nameEn": "Product Release Process"
        },
        "branch": {
            "id": null
        }
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“master”、“branch”表示生命周期模板的主对象和分支对象。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [
        {
            "id": "504971863990996992",
            "className": "LifecycleTemplate",
            "name": "产品发布流程",
            "description": "",
            "master": {
                "id": "504971863999385600",
                "className": "LifecycleTemplateMaster",
                "businessCode": "LCT00000002",
                "nameEn": "Product Release Process",
                "name": "产品发布流程",
                "disableFlag": false
            },
            "branch": {
                "id": "504971863999385601",
                "className": "LifecycleTemplateBranch",
                "version": "A"
            },
            "latest": true,
            "latestIteration": true,
            "versionCode": 1,
```

```
            "iteration": 1,
            "version": "A",
            "latestVersion": true,
            "workingCopy": false,
            "workingState": {
                "code": "CHECKED_IN",
                "cnName": "已检入",
                "enName": "checked in",
                "alias": "CHECKED_IN"
            },
            "checkOutUserName": null,
            "checkOutTime": null,
            "preVersionId": null,
            "descriptionEn": "",
            "nameEn": "Product Release Process",
            "reserved": null,
            "lifecyclePhaseList": []
        }
    ],
    "errors": []
}
```


----结束

步骤 4：绑定生命周期模板

步骤1 准备工作。

在应用运行态中获取绑定生命周期模板的API信息，具体操作请参见。

步骤2 根据实际业务对象的生命周期管理，调用API绑定生命周期模板。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/basic/api/LifecycleTemplate/updateLifecycleInfo

{
    "params": {
        "applicationId": "1bab18b61c6f4f5bbd031d2d8e0e3fda",
        "id": "478135618698743808",
        "curLifecycleTemplateAttached": [
            {
                "id": "63c7e2aad21a4715bd8ac08d73da4805",
                "templateAttrName": "生命周期模板",
                "templateAttrNameEn": "LifecycleTemplate",
                "templateAttrDescription": "生命周期模板",
                "templateAttrDescriptionEn": "LifecycleTemplate",
                "stateAttrName": "生命周期状态",
                "stateAttrNameEn": "LifecycleState",
                "stateAttrDescription": "生命周期状态",
                "stateAttrDescriptionEn": "LifecycleState",
                "attachTemplateId": "504971863990996992",
                "attachTemplateName": "Product Release Process",
                "attachTemplateNameEn": "产品发布流程"
            }
        ]
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“attachTemplateId”、“attachTemplateName”、“attachTemplateNameEn”表示需要绑定的生命周期模板ID、名称和英文名称。

- 响应示例


```
{
    "result": "SUCCESS",
    "data": [
        {
```



```
        "id": "478135618698743808",
        "rdmExtensionType": "TypeDefinition",
        "className": "TypeDefinition",
        ..... // 实体详细信息
        "lifecycleTemplateAttached": [
          {
            "attachTemplateId": "477901037156446208",
            "stateAttrDescriptionEn": "LifecycleState",
            "readOnly": true,
            "stateAttrName": "生命周期状态",
            "attachTemplateNameEn": "Test0316",
            "templateAttrDescriptionEn": "LifecycleTemplate",
            "templateAttrDescription": "生命周期模板",
            "stateAttrDescription": "生命周期状态",
            "id": "63c7e2aad21a4715bd8ac08d73da4805",
            "attachTemplateName": "Test0316",
            "templateAttrNameEn": "LifecycleTemplate",
            "templateAttrName": "生命周期模板",
            "stateAttrNameEn": "LifecycleState"
          }
        ],
        "curLifecycleTemplateAttached": [
          {
            "id": "63c7e2aad21a4715bd8ac08d73da4805",
            "templateAttrName": "生命周期模板",
            "templateAttrNameEn": "LifecycleTemplate",
            "templateAttrDescription": "生命周期模板",
            "templateAttrDescriptionEn": "LifecycleTemplate",
            "stateAttrName": "生命周期状态",
            "stateAttrNameEn": "LifecycleState",
            "stateAttrDescription": "生命周期状态",
            "stateAttrDescriptionEn": "LifecycleState",
            "attachTemplateId": "504971863990996992",
            "attachTemplateName": "Product Release Process",
            "attachTemplateNameEn": "产品发布流程"
          }
        ]
      }
    ],
    "errors": []
}
```


----结束

步骤 5：使用生命周期模板

步骤1 准备工作。

在应用运行态中获取更新生命周期阶段的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务对象当前的生命周期阶段，调用API更新生命周期。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/dynamic/api/{entityName}/update

{
    "params": {
        ..... // 其他实体属性
        "lifecycleTemplate": {
            "id": "455805277254459392",
            "clazz": "LifecycleTemplate"
        },
        "lifecycleState": {
            "id": "455663911463555072",
            "clazz": "LifecycleState"
        },
        "tenant": {
            "id": "-1",
```



```
            "clazz": "Tenant"
        },
        "creator": "XDM_Developer 93172bbfd0f644**********345386",
        "modifier": "XDM_Developer",
        "id": "505046194565681152"
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，{entityName}表示实体的英文名称，“lifecycleTemplate”、“lifecycleState”表示实例需要更新的生命周期模板和状态。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [
        {
            "id": "505046194565681152",
            ..... // 其他实例属性
            "disableFlag": false,
            "lifecycleTemplate": {
                "id": "455805277254459392",
                "className": "lifecycleTemplate"
                ..... // 其他生命周期模板属性
            },
            "lifecycleState": {
                "id": "455801165657935872",
                "className": "LifecycleState"
                ..... // 其他状态属性
            }
        }
    ],
    "errors": []
}
```

----结束

6 使用服务编排管理

6.1 服务编排管理概述

简介

工业数字模型驱动引擎-数据建模引擎（xDM Foundation，简称xDM-F）不仅为用户提供了丰富的标准API（如数据实体API、关系实体API、系统管理API），还提供了高代码的服务编排能力，适配Java和JavaScript类型服务编排。当iDME提供的标准API无法满足用户的业务需求时，用户可使用服务编排能力编排原子API形成一个跨实体（表）的组合API，提高应用开发的速度与质量。

使用方法

- 使用应用运行态

  您可以通过应用运行态使用服务编排，具体操作请参见**服务编排管理**和**使用高代码服务编排自定义API**。

- 使用全量数据服务API

  您可以通过**全量数据服务**提供的API使用服务编排管理，具体操作请参见**6.2 服务编排开发指导**。

使用说明

服务编排管理的使用说明请参见**服务编排管理**。

6.2 服务编排开发指导

涉及的接口

表 6-1 涉及的接口

接口名称	说明
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/customservice/create	创建服务编排。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/customservice/code/save	保存服务编排。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/customservice/publish	发布服务编排。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/customservice/**{APIName}**/execute	调用服务编排脚本。

步骤 1：创建服务编排

步骤1 准备工作。

在应用运行态中获取创建服务编排的API信息，具体操作请参见**查询API**。

步骤2 根据实际的业务需求，调用API创建服务编排。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/customservice/create
{
    "params": {
        "name": "按票价和余额查询",
        "nameEn": "QueryByPriceRemain",
        "description": "按票价和余额查询",
        "descriptionEn": "Query by price and remain",
        "owner": "",
        "needPublishToApiCenter": false,
        "tags": [],
        "supportPagination": true,
        "master": {
            "supportPagination": true,
            "apiType": "javascript",
            "tenantNameEn": "TestDME",
            "applicationId": "{applicationId}", // 应用ID
            "tenantAppId": "0",
            "tagIds": []
        },
        "gray": false  // 此参数已废弃
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“name”、“nameEn”、“description”、“descriptionEn”表示服务编排的中英文名称和中英文描述，“master”表示服务编排所属的应用信息。

- 响应示例

```
{
   "result": "SUCCESS",
   "data": [
      {
         "id": "505105143826485248",
         "creator": "XDM_Developer 93172bbfd0f644**********345386",
         "modifier": "XDM_Developer 93172bbfd0f644**********345386",
         "createTime": "2023-05-30T11:47:30.688+0000",
         "lastUpdateTime": "2023-05-30T11:47:30.688+0000",
         "rdmVersion": 1,
         "rdmDeleteFlag": 0,
         "rdmExtensionType": "CustomService",
         "tenant": {
            "id": "-1",
            "rdmExtensionType": "Tenant",
            ......
         },
         "className": "CustomService",
         "name": "按票价和余额查询",
         "state": "INWORK",
         "nameEn": "QueryByPriceRemain",
         "descriptionEn": "Query by price and remain",
         "description": "按票价和余额查询",
         "owner": "",
         "disabled": null,
         "needPublishToApiCenter": false,
         "publishedToApiCenter": false,
         "publishedToApiCenterTime": null,
         "needAccessControlled": false,
         "latest": true,
         "workingCopy": true,
         "version": "A",
         "iteration": 1,
         "master": {
            "id": "505105143805513728",
            "creator": "XDM_Developer 93172bbfd0f644**********345386",
            "modifier": "XDM_Developer 93172bbfd0f644**********345386",
            "createTime": "2023-05-30T11:47:30.683+0000",
            "lastUpdateTime": "2023-05-30T11:47:30.683+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "CustomServiceMaster",
            "tenant": {
               "id": "-1",
               "rdmExtensionType": "Tenant"
               ......
            },
            "className": "CustomServiceMaster",
            "apiType": "javascript",
            "apiNumber": "CS00000010",
            "tenantNameEn": "TestDME",
            "tenantAppId": "0",
            "tagList": []
         },
         "apiCenterInfo": {
            ......
         },
         "gray": false,  // 此参数已废弃
         "supportPagination": true
      }
   ],
```

```
    "errors": []
}
```


----结束

步骤 2：设置服务编排脚本

步骤1 准备工作。

在应用运行态中设置服务编排脚本的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务需求，调用API设置服务编排脚本。

- 请求示例


```
POST http://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/customservice/code/save

{
    "params": {
        "id": "505105143826485248",
        "apiNameEn": "QueryByPriceRemain",
        "requestClass": "",
        "responseClass": "",
        "serviceClass": "var request = \"\"; \nvar sqlCount  = \"select count(*) as count from TestDME_Play_REL p \";  \nvar sql = \"select c.name as cinema, f.name as film, p.price, p.remain from TestDME_Play_REL p \"; \nsql += \"left join TestDME_Cinema c on c.id = p._sourceid \"; \nsql += \"left join TestDME_Film f on f.id = p._targetid \"; \n \nif (request.price != null && request.remain!=null) \n{ \n    sql += \" where price <= {#price} and remain>={#remain} \"; \n    sqlCount += \" where price <= {#price} and remain>= {#remain} \"; \n}\n",
        "apiType": "javascript"
    }
}
```


其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“apiNameEn”、“apiType”、“serviceClass”表示服务编排的名称、类型和脚本。

- 响应示例


```
{
    "result": "SUCCESS",
    "data": [
        {
            "requestClass": null,
            "responseClass": null,
            "serviceClass": "var request = \"\"; \nvar sqlCount  = \"select count(*) as count from TestDME_Play_REL p \";  \nvar sql = \"select c.name as cinema, f.name as film, p.price, p.remain from TestDME_Play_REL p \"; \nsql += \"left join TestDME_Cinema c on c.id = p._sourceid \"; \nsql += \"left join TestDME_Film f on f.id = p._targetid \"; \n \nif (request.price != null && request.remain!=null) \n{ \n    sql += \" where price <= {#price} and remain>={#remain} \"; \n    sqlCount += \" where price <= {#price} and remain>= {#remain} \"; \n}\n",
            "requestExample": "{\"params\":{\"price\":\"String\",\"remain\":\"String\",\"returnTotalCountFlag\":\"String\"}}"
        }
    ],
    "errors": []
}
```


----结束

步骤 3：发布服务编排

步骤1 准备工作。

在应用运行态中获取发布服务编排的API信息，具体操作请参见**查询API**。

步骤2 完成服务编排脚本编写后，调用API发布服务编排。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/customservice/publish?versionId=505105143826485248
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“versionId”表示待发布服务编排的版本ID。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [],
    "errors": []
}
```

----结束

步骤 4：调用服务编排生成的 API

步骤1 准备工作。

在应用运行态中获取创建业务操作的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务对象的生命周期管理流程，调用API创建业务操作。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/publicservices/rdm/basic/api/customservice/QueryByPriceRemain/execute/{pageSize}/{curPage}

{
    "params": {
        "price": "19.9",
        "remain": "9",
        "returnTotalCountFlag": "true"
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“price”、“remain”表示对应服务编排的自定义入参。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [
        {
            "cinema": "太平洋影城",
            "film": "萌鸡小队：萌闯新世界",
            "price": "9.90",
            "remain": 17
        },
        {
            "cinema": "保利国际影城",
            "film": "满江红",
            "price": "19.90",
            "remain": 9
        }
    ],
    "errors": [],
    "pageInfo": {
        "curPage": 1,
        "pageSize": 20,
        "totalRows": 2,
        "totalPages": 1
    }
}
```

----结束

7 使用搜索服务管理

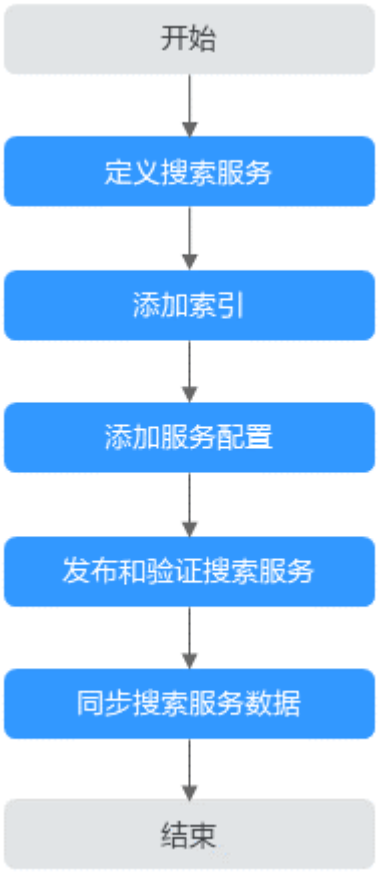
7.1 搜索服务管理概述

概述

在以往的工业应用开发场景中，常见的难题包括跨数据类型、跨数据库搜索难度大，业务数据种类多、搜索场景多、第三方搜索引擎在线应用无法在客户私有环境使用、图形搜索实现难度大等。为了满足应用开发人员研发效率的需求，工业数字模型驱动引擎-数据建模引擎（xDM Foundation，简称xDM-F）提供根据关键字查询特定目标并分页返回所有满足匹配要求的对象条目、按对象属性设置搜索条件并通过逻辑关系组合复杂搜索条件进行对象搜索的搜索服务管理功能。

使用流程

图 7-1 定义搜索服务流程



使用方法

- 使用应用运行态
  您可以通过应用运行态使用搜索服务，具体操作请参见**创建搜索服务**。
- 使用全量数据服务API
  您可以通过**全量数据服务**提供的API使用搜索服务管理，具体操作请参见**7.2 搜索服务管理开发指导**。

7.2 搜索服务管理开发指导

涉及的接口

表 7-1 涉及的接口

接口名称	说明
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/searchservicedefine/create	创建搜索服务。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/searchServiceIndexDefinition/saveList	添加索引。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/searchServColumnController/saveDataServIndexEntityList	添加服务配置。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/searchservicedefine/publishSingle	发布搜索服务。
https://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/searchServ/startEsSync/**{serNumber}**	同步搜索服务。

步骤 1：定义搜索服务

步骤1 准备工作。

在应用运行态中获取定义搜索服务的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务需求，调用API创建搜索服务。

- 请求示例

```
POST https://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/searchservicedefine/create

{
    "params": {
        "createDTO": {
            "name": "搜索服务示例",
            "description": "搜索服务示例",
            "descriptionEn": "Sample",
            "nameEn": "Sample",
            "owner": "",
            "tenantAliasName": "TestDME"
        },
        "tags": []
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“name”、“nameEn”、“description”、“descriptionEn”表示搜索服务的中英文名称和中英文描述。

- 响应示例


```
{
   "result": "SUCCESS",
   "data": [
      {
         "id": "505115507221864448",
         "className": "XDMSearchServDefine",
         "name": "搜索服务示例",
         "description": "搜索服务示例",
         "kiaguid": null,
         "securityLevel": "internal",
         "master": {
            "id": "505115507255418880",
            "creator": "TestAccount2 5547b9adee**********8aac057d2c",
            "modifier": "TestAccount2 5547b9adee**********8aac057d2c",
            "createTime": "2023-05-30T12:28:41.522+0000",
            "lastUpdateTime": "2023-05-30T12:28:41.522+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "XDMSearchServDefineMaster",
            "tenant": {
               "id": "-1",
               "creator": "xdmAdmin",
               "modifier": "xdmAdmin",
               "createTime": "2022-12-01T11:24:39.000+0000",
               "lastUpdateTime": "2022-12-01T11:24:39.000+0000",
               "rdmVersion": 1,
               "rdmDeleteFlag": 0,
               "rdmExtensionType": "Tenant",
               "tenant": null,
               "className": "Tenant",
               "name": "basicTenant",
               "description": "默认租户",
               "kiaguid": null,
               "securityLevel": "internal",
               "code": "basicTenant",
               "disableFlag": false,
               "dataSource": "DefaultDataSource"
            },
            "className": "XDMSearchServDefineMaster",
            "servNumber": "SS60466196"
         },
         "branch": {
            "id": "505115507255418881",
            "creator": "TestAccount2 5547b9adee**********8aac057d2c",
            "modifier": "TestAccount2 5547b9adee**********8aac057d2c",
            "createTime": "2023-05-30T12:28:41.569+0000",
            "lastUpdateTime": "2023-05-30T12:28:41.569+0000",
            "rdmVersion": 1,
            "rdmDeleteFlag": 0,
            "rdmExtensionType": "XDMSearchServDefineBranch",
            "tenant": {
               "id": "-1",
               "creator": "xdmAdmin",
               "modifier": "xdmAdmin",
               "createTime": "2022-12-01T11:24:39.000+0000",
               "lastUpdateTime": "2022-12-01T11:24:39.000+0000",
               "rdmVersion": 1,
               "rdmDeleteFlag": 0,
               "rdmExtensionType": "Tenant",
               "tenant": null,
               "className": "Tenant",
               "name": "basicTenant",
               "description": "默认租户",
               "kiaguid": null,
```



```
                "securityLevel": "internal",
                "code": "basicTenant",
                "disableFlag": false,
                "dataSource": "DefaultDataSource"
            },
            "className": "XDMSearchServDefineBranch",
            "version": "A"
        },
        "latest": true,
        "latestIteration": true,
        "versionCode": 1,
        "iteration": 1,
        "version": "A",
        "latestVersion": true,
        "workingCopy": true,
        "workingState": {
            "code": "INWORK",
            "cnName": "工作中",
            "enName": "inwork",
            "alias": "INWORK"
        },
        "checkOutUserName": "TestAccount2 5547b9adee54423cbc05978aac057d2c",
        "checkOutTime": "2023-05-30T12:28:41.514+0000",
        "preVersionId": null,
        "owner": "",
        "descriptionEn": "Sample",
        "esindexId": null,
        "dmentityVersion": null,
        "dmentityVersionName": null,
        "step": null,
        "nameEn": "Sample",
        "needRefresh": false,
        "graphId": null,
        "lifecycleTemplate": null,
        "lifecycleState": null,
        "disableFlag": false
    }
  ],
  "errors": []
}
```


----结束

步骤 2：添加索引

步骤1 准备工作。

在应用运行态中获取添加索引的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务需求，调用API添加索引。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/searchServiceIndexDefinition/saveList
```


```
{
  "params": {
    "searchServId": "505115507221864448",
    "searchServName": "搜索服务示例",
    "indexServColuVOList": [
      {
        "id": null,
        "indexName": "SampleIndex",
        "indexDesc": "",
        "indexType": "TEXT",
        "operator": null,
        "inputSeparator": null,
        "segMethod": "NOWORD",
```



```
            "segOption": "NOTINVOLVED",
            "searchUsage": true,
            "keywordUsage": true,
            "displayUsage": true,
            "matchType": "FUZZY",
            "key": "indexDefinition43"
        }
    ]
  }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“indexName”、“indexDesc”、“indexType”表示索引名称、描述和类型。

- 响应示例

```
{
  "result": "SUCCESS",
  "data": [
    {
      "id": "505115594543079424",
      "rdmExtensionType": "XDMSearchIndexEntity",
      "className": "XDMSearchIndexEntity",
      "searchServId": "505115507221864448",
      "indexName": "SampleIndex",
      "matchType": {
        "code": "fuzzy",
        "cnName": "模糊匹配",
        "enName": "fuzzy",
        "alias": "FUZZY"
      },
      "displayUsage": true,
      "keywordUsage": true,
      "operator": null,
      "segOption": {
        "code": "notInvolved",
        "cnName": "不涉及",
        "enName": "notInvolved",
        "alias": "NOTINVOLVED"
      },
      "indexMaxFieldSize": null,
      "indexType": {
        "code": "Text",
        "cnName": "文本",
        "enName": "Text",
        "alias": "TEXT"
      },
      "indexDesc": "",
      "searchUsage": true,
      "segMethod": {
        "code": "noWord",
        "cnName": "不分词",
        "enName": "noWord",
        "alias": "NOWORD"
      },
      "searchServName": "搜索服务示例",
      "inputSeparator": null,
      "relationIndexEntityList": null
    }
  ],
  "errors": []
}
```

----结束

步骤 3：添加服务配置

步骤1 准备工作。

在应用运行态中获取添加服务配置的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务需求，调用API添加服务配置。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/searchServColumnController/saveDataServIndexEntityList

{
    "params": {
        "searchServId": "505115507221864448",
        "searchServIndexEntityVOList": [
            {
                "entityName": "TestData",
                "entityNameEn": "TestData",
                "dataServRelationVOList": [
                    {
                        "attrName": "Code",
                        "attrDesc": null,
                        "searchIndexId": "505115594543079424",
                        "indexType": "TEXT",
                        "dataModelVersionNameEn": "Tenant",
                        "extendAttr": false,
                        "attrType": "STRING",
                        "sortNo": 1
                    }
                ]
            }
        ]
    }
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“searchServIndexEntityVOList”表示服务配置中选中的实体和属性。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [],
    "errors": []
}
```

----结束

步骤 4：发布和验证搜索服务

步骤1 准备工作。

在应用运行态中获取发布搜索服务的API信息，具体操作请参见**查询API**。

步骤2 完成搜索服务的配置后，调用API发布搜索服务。

- 请求示例

```
POST http://{Endpoint}/rdm_{appID}_app/services/rdm/basic/api/searchservicedefine/publishSingle

{
    "params": "505115507221864448"
}
```

其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“params”表示待发布的搜索服务ID。

- 响应示例

```
{
    "result": "SUCCESS",
    "data": [],
    "errors": []
}
```

----结束

步骤 5：同步搜索服务数据

步骤1 准备工作。

在应用运行态中获取同步搜索服务的API信息，具体操作请参见**查询API**。

步骤2 根据实际业务需求，调用API同步搜索服务数据。

- 请求示例

  PUT http://**{Endpoint}**/rdm_**{appID}**_app/services/rdm/basic/api/searchServ/startEsSync/**{serNumber}**

  其中，{Endpoint}表示数据建模引擎所在域名或IP地址，{appID}表示应用ID，“serNumber”待同步的搜索服务ID。

- 响应示例

```
{
   "result": "SUCCESS",
   "data": [],
   "errors": []
}
```

----结束

8 使用客户端 SDK

操作场景

当您已在应用设计态完成应用的构建和发布时，您可以通过客户端SDK在本地部署iDME环境，即可直接调用客户端SDK提供的接口函数，实现使用iDME业务能力。

操作步骤

步骤1 下载iDME Java SDK最新版本源码，安装SDK。

具体操作请参见**获取与安装SDK**。

步骤2 完成安装后，即可直接调用SDK的相关操作。

相关示例可参见：

- **本地应用注入iDME注解示例**
- **创建数据实例示例**
- **查询某实体的实例数据示例**

----结束

9 修订记录

发布日期	修订记录
2023-10-29	第四次正式发布。 修改**4.7 创建多维版本的数据实例**
2023-07-22	第三次正式发布。 新增 • **4.4 分页查询指定用户的实例数据** • **4.5 比较M-V模型的版本对象** • **4.6 另存版本对象的实例数据** • **4.7 创建多维版本的数据实例** 修改 • **2 准备工作** • **3 构建iDME应用**
2023-06-17	第二次正式发布。 修改 • **4.2 创建指定实体的实例** • **4.3 查询满足条件的实例数据** • **5.2 生命周期管理开发指导** • **6.2 服务编排开发指导** • **7.2 搜索服务管理开发指导**
2023-06-06	第一次正式发布。