|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **研发生产中心** | **文档编号** |  | **版本** | A1 | **密级** | 商密A |
| **项目名称** | **EI子系统REVIEW文档** | | | | |
| **项目来源** |  | | | | |

**EI子系统REVIEW文档**

**设计说明书**

(内部资料 请勿外传)

|  |  |  |  |
| --- | --- | --- | --- |
| **编 写：** | 黄 聊 | **日 期：** |  |
| **检 查：** |  | **日 期：** |  |
| **审 核：** |  | **日 期：** |  |
| **批 准：** |  | **日 期：** |  |

**深圳市计算科学院**

**版权所有 不得复制**

**文档变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1. 引言](#_Toc11144)

[1.1 背景](#_Toc18296)

[1.2 编写目的和范围](#_Toc4777)

[1.3 术语表](#_Toc7492)

[1.4 参考资料](#_Toc7512)

[2. 设计概述](#_Toc20510)

[2.1 任务和目标](#_Toc9914)

[2.1.1 需求概述](#_Toc32336)

[2.1.2 运行环境概述](#_Toc11452)

[2.1.3 条件与限制](#_Toc27548)

[2.1.4 详细设计方法和工具](#_Toc21375)

[3. 系统详细需求分析](#_Toc25978)

[3.1 详细需求分析](#_Toc27918)

[3.1.1 详细功能需求分析](#_Toc3701)

[3.1.2 详细性能需求分析](#_Toc28615)

[3.1.3 详细资源需求分析](#_Toc29034)

[3.1.4 详细接口需求分析](#_Toc22824)

[3.1.5 详细系统运行环境及限制条件分析](#_Toc940)

[4. 总体方案](#_Toc10201)

[4.1 特性总体流程](#_Toc13659)

[4.1.1 特性组成、逻辑结构及层次](#_Toc12071)

[5. 特性功能设计](#_Toc27000)

[5.1 xxx管理](#_Toc22359)

[6. 接口设计](#_Toc15218)

[6.1 接口说明](#_Toc14703)

[6.1.1 调用方式](#_Toc26175)

[7. 数据库设计](#_Toc14952)

[8. 系统安全保密设计](#_Toc334)

[8.1 说明](#_Toc10479)

[8.2 设计](#_Toc10690)

[8.2.1 数据传输部分](#_Toc14869)

[8.2.2 IP过滤分部](#_Toc8928)

[8.2.3 身份验证部分](#_Toc25766)

[9. 系统性能设计](#_Toc3734)

[10. 系统出错处理](#_Toc22516)

[11. 可靠性设计](#_Toc10109)

# 引言

## 背景

对EI子系统进行代码质量加固，查看EI中主要接口的参数、时序图和类图。

## 编写目的和范围

本特性说明书编写的目的是说明特性的设计考虑，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 术语表

定义系统或产品中涉及的重要术语，为读者在阅读文档时提供必要的参考信息。

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 |  |  |
| 2 |  |  |
|  |  |  |

## 参考资料

列出有关资料的名称、作者、文件编号或版本等。参考资料包括：

a．需求说明书、架构设计说明书等；

b．本项目的其他已发表的文件；

c．引用文件、资料、软件开发标准等。

|  |  |  |  |
| --- | --- | --- | --- |
| 资料名称 | 作者 | 文件编号、版本 | 资料存放地点 |
| *SEIOIS系统架构说明书* | *PoD研究室* |  |  |

# 设计概述

## 任务和目标

查看EI子系统中主要接口的逻辑，含时序图，类图。

### 需求概述

为加固提升EI子系统代码质量，对EI子系统的主要接口完成代码REVIEW。

### 运行环境概述

xxx

### 条件与限制

//性能/功能/可靠性等约束

### 详细设计方法和工具

# 系统详细需求分析

xxx

## 详细需求分析

### 详细功能需求分析

xxx

### 详细性能需求分析

xxx。

### 详细资源需求分析

xxx

### 详细接口需求分析

参考架构设计文档

### 详细系统运行环境及限制条件分析

暂无

# 总体方案

着重解决系统总体结构确认及界面划分问题。

## 特性总体流程

### 特性组成、逻辑结构及层次

XXX

# 特性功能设计

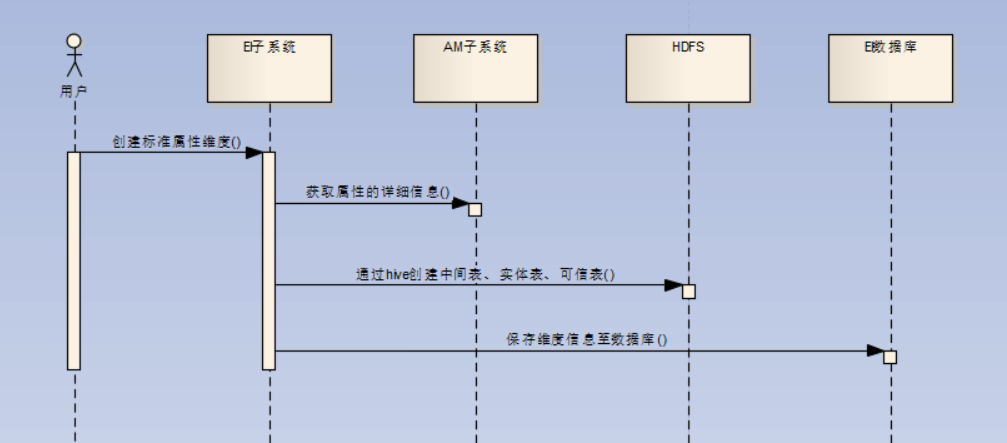
XXX

# 接口设计

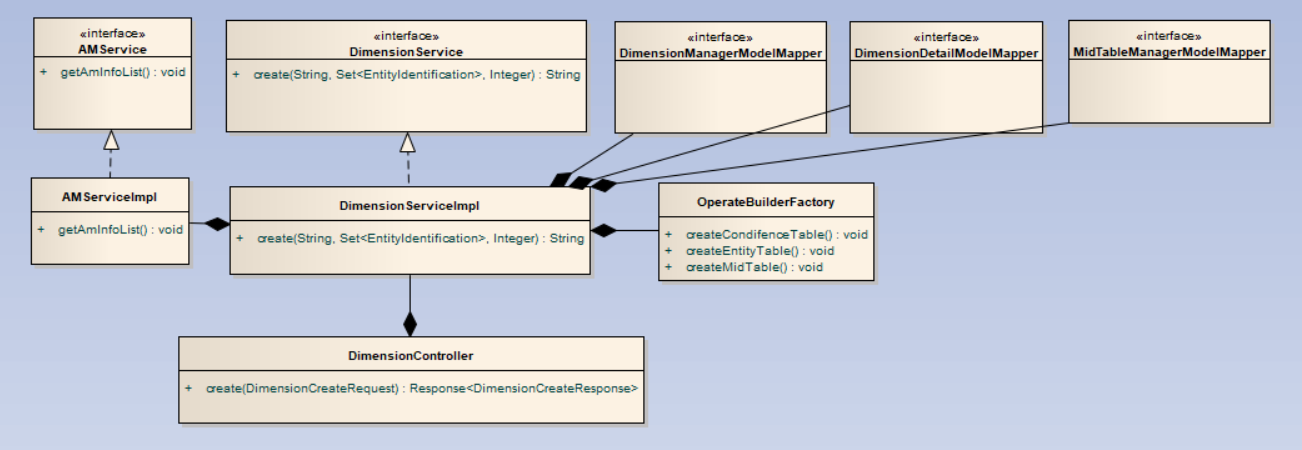
## 创建标准属性维度接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/dimension/create | |
| **入参** | **字段** | **描述** |
|  | attributeIds (Array[string]): | 属性列表 |
|  | dimensionName (string): | 维度名称 |
|  | identifications (Array[string], optional): | 识别标识属性 |
| **返回** | **字段** | **描述** |
|  | dimensionId (string): | 维度id |

### 接口时序图



### 接口类图



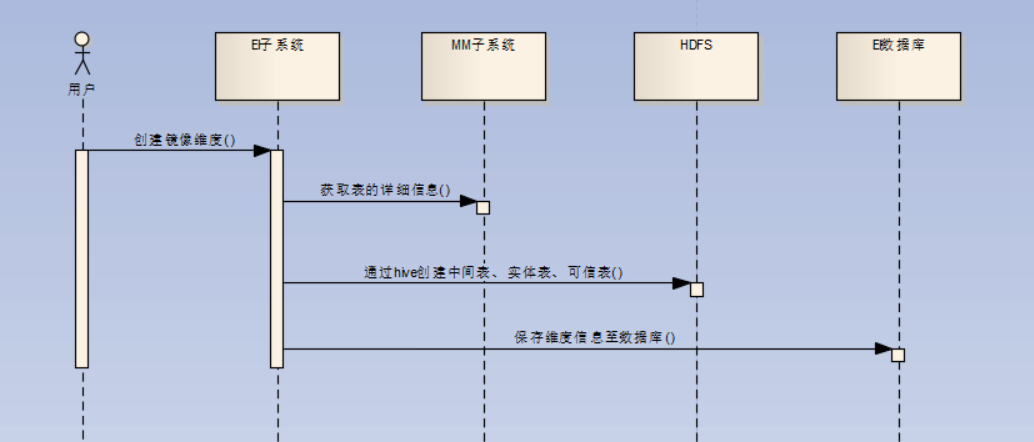
类描述：

1. 用户通过url调用控制层DimensionController的create方法；
2. DimensionServiceImpl类的create方法处理维度创建；
3. AMServiceImpl类的getAmInfoList方法请求AM子系统获取标准属性的详细信息；
4. OperateBuilderFactory类的用来处理中间表、实体表、可信表在hive上的创建；
5. MidTableManagerModelMapper类用来保存中间表信息；
6. DimensionManagerModelMapper类用来保存维度信息；
7. DimensionDetailModelMapper；类用来保存维度属性信息。

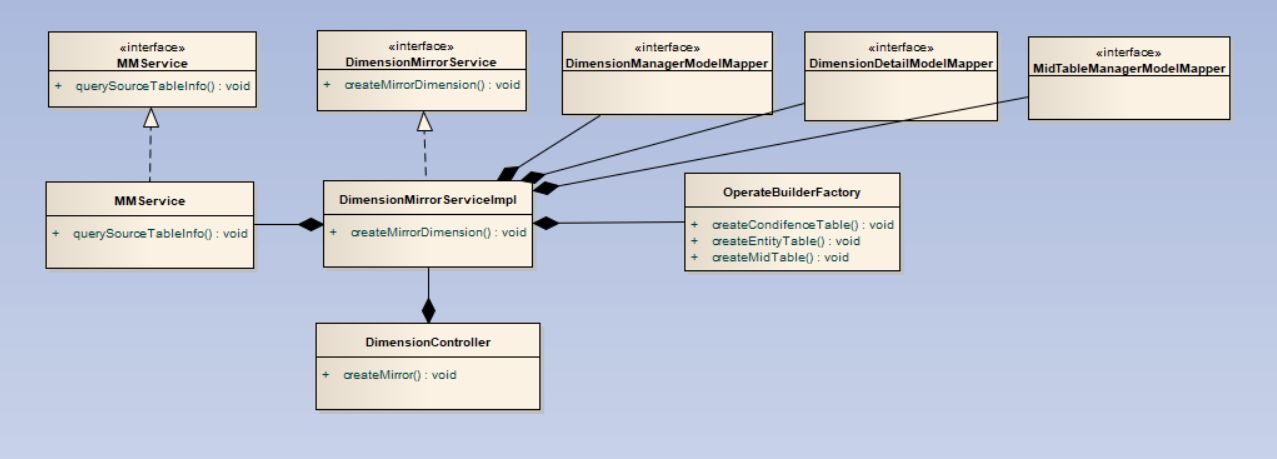
## 创建镜像维度接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/dimension/mirror/create | |
| **入参** | **字段** | **描述** |
|  | desc (string): | 描述 |
|  | dsId (string): | 数据源id |
|  | schemas (string): | schemas |
|  | tablename (string): | 表名 |
| **返回** | **字段** | **描述** |
|  | dimensionId (string): | 维度id |

### 接口时序图



### 接口类图



类描述：

1、 用户通过url调用控制层DimensionController的createMirror方法；

2、 DimensionMirrorServiceImpl类的createMirrorDimension方法处理镜像维度创建；

3、 MMServiceImpl类的querySourceTableInfo方法请求MM子系统获取表字段信息；

4、 OperateBuilderFactory类的用来处理中间表、实体表、可信表在hive上的创建；

5、 MidTableManagerModelMapper类用来保存中间表信息；

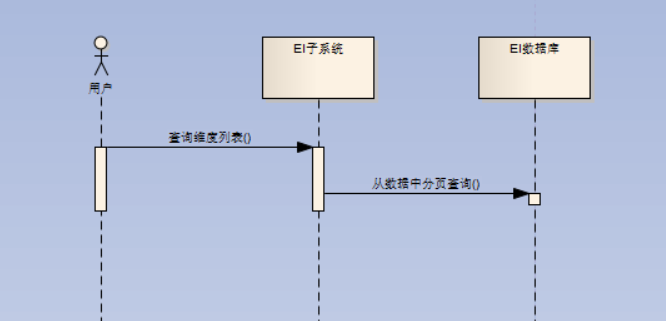
6、 DimensionManagerModelMapper类用来保存维度信息；

7、 DimensionDetailModelMapper；类用来保存维度属性信息。

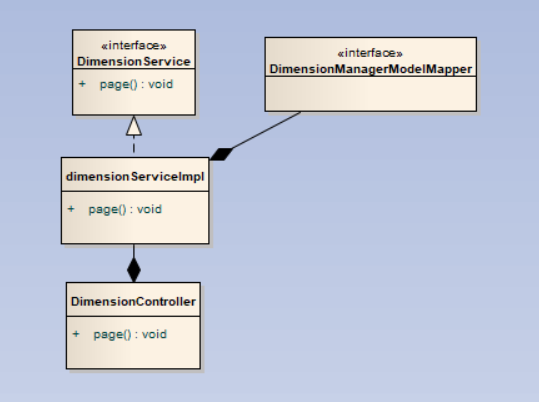
## 查询维度列表接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/dimension/page | |
| **入参** | **字段** | **描述** |
|  | dimensionName (string, optional): | 维度名称 |
|  | dimensionDesc (string, optional): | 维度描述 |
|  | pageSize (integer, optional): | 第几页，默认第1页 |
|  | pageNum (integer, optional): | 每页多上行，默认15行 |
| **返回** | **字段** | **描述** |
|  | id | 维度id |
|  | name | 维度名称 |
|  | remark | 维度描述 |
|  | type | 维度类型 |
|  | dsId | 数据源id |
|  | dsSchema | schema |
|  | dsTablename | 表名称 |
|  | createby | 创建者 |
|  | createtime | 创建时间 |

### 接口时序图



### 接口类图



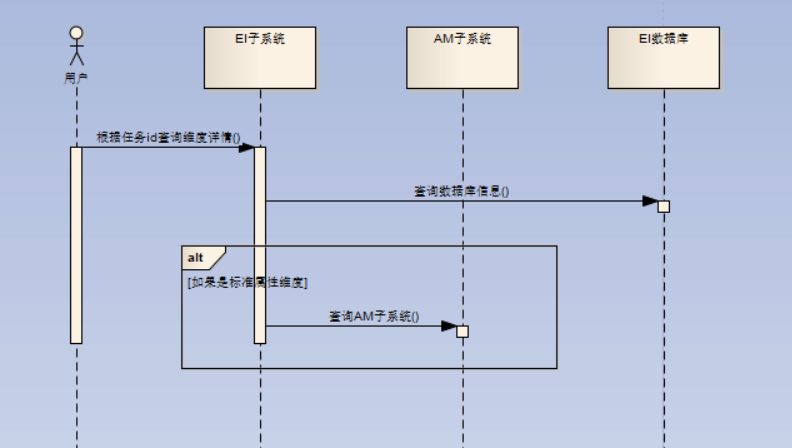
类描述：

1. 用户通过url访问DimensionController的list方法查询维度分页信息；
2. DimensionServiceImpl类的page方法用来分页查询维度信息；
3. DimensionManagerModelMapper类用来查询数据库。

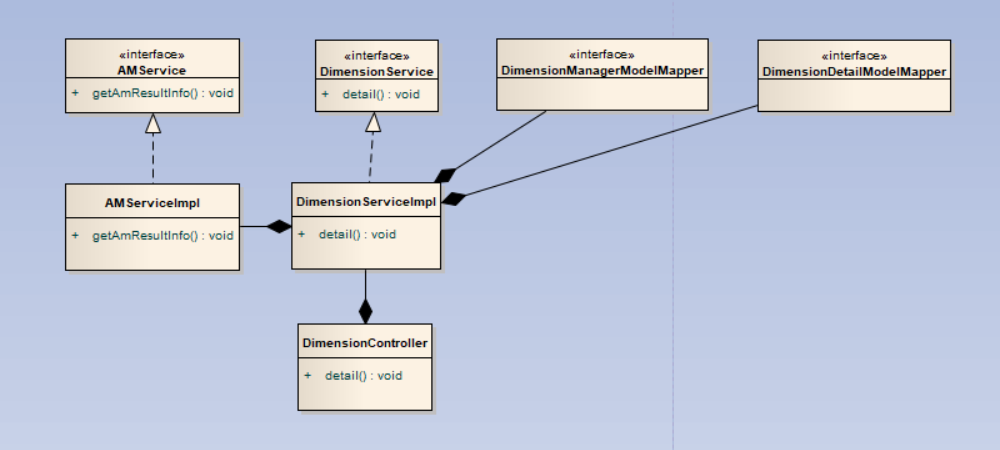
## 查询维度详情接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/dimension/detail | |
| **入参** | **字段** | **描述** |
|  | dimensionId (string, optional) | 维度id |
| **返回** | **字段** | **描述** |
|  | id | 维度id |
|  | name | 维度名称 |
|  | remark | 维度描述 |
|  | type | 维度类型 |
|  | dsId | 数据源id |
|  | dsSchema | schema |
|  | dsTablename | 表名称 |
|  | createby | 创建者 |
|  | createtime | 创建时间 |
|  | attributeDataList (Array[属性类]): | 属性列表 |
| 属性类 | attributeDesc (string) | 属性名称 |
|  | attributeDetailInfos (Array[EiAttributeDetailInfo]): | 属性的映射关系 |
|  | attributeId (string) | 属性id |
|  | attributeName (string) | 属性名称 |
|  | attributeType (string,) | 属性数据类型 |
|  | modelId (string) | 机器学习模型id |
| EiAttributeDetailInfo | attributionName (string) | 属性名称 |
|  | attributionType (string) | 属性类型 |
|  | catalog (string) | 映射结果的字段来自presto中配置的catalog |
|  | column (string) | 映射结果的字段名称 |
|  | comment (string): | 映射结果的字段描述 |
|  | database (string) | 映射结果的字段的数据库名称 |
|  | databaseSourceId (string) | 映射结果的字段来自数据源id |
|  | schema (string) | 映射结果的字段schema |
|  | table (string) | 映射结果的字段的表名称 |
|  | type (string) | 映射结果的字段类型 |

### 接口时序图



### 接口类图



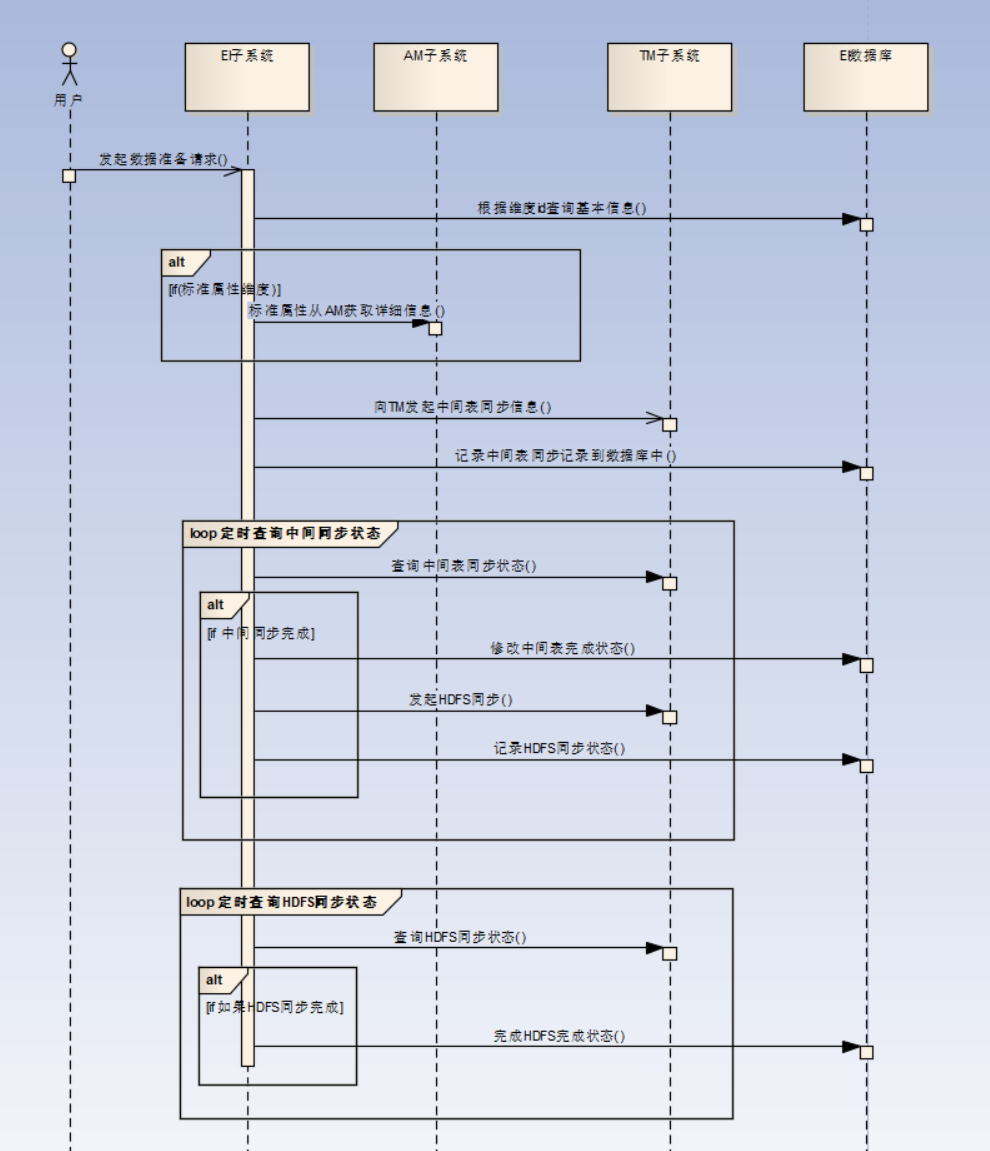
类描述：

1. 用户通过url访问DimensionController的detail方法查询维度详情信息；
2. DimensionServiceImpl类的detail方法用来处理维度详情信息；
3. 如果维度时标准属性维度，则通过AMServiceImpl类的getAmResultInfo方法获取维度的标准属性详情信息；
4. DimensionManagerModelMapper类用来查询数据库。
5. DimensionDetailModelMapper类用来查询维度列的详情信息。

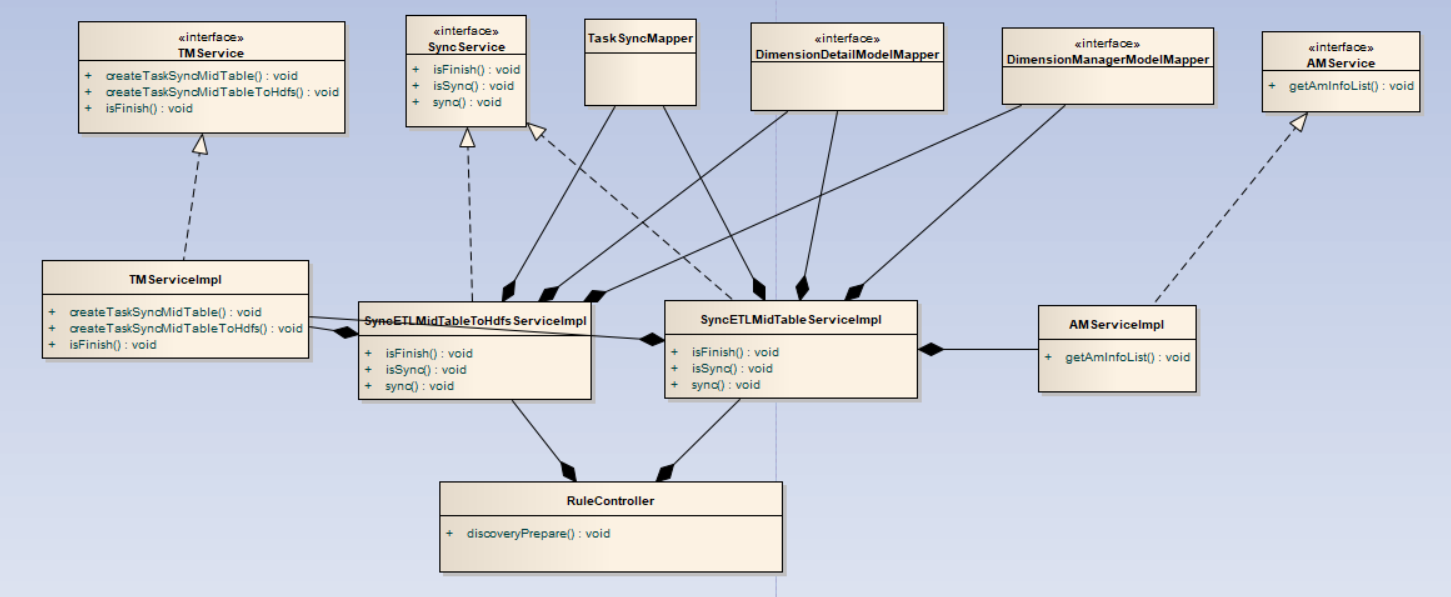
## 数据准备接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/discovery/prepare | |
| **入参** | **字段** | **描述** |
|  | dimensionIds (Array[string]) | 维度IDs |
|  | taskName (string) | 任务名称 |
| **返回** | **字段** | **描述** |
|  | mainTaskId | 主任务id |

### 接口时序图



### 接口类图



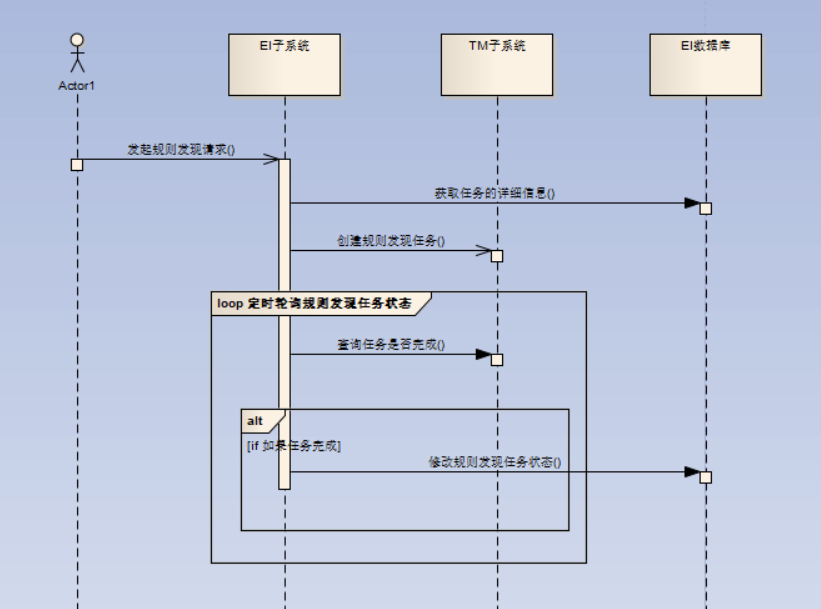
类描述：

1. 用户想RuleController的discoveryPrepare方法发起数据准备请求；
2. TMServiceImpl类用实现向TM子系统创建任务;
3. SyncService接口的实现类SyncETLMidTableServiceImpl和SyncETLMidTableToHdfsServiceImpl分别用来是实现中间表同步、HDFS同步、以及轮训查询TM子系统中任务的执行状态；
4. AMServiceImpl主要用来向AM子系统中查询标准属性维度的详细信息；
5. TaskSyncMapper用来存储中间表同步任务、HDFS同步任务；
6. DimensionManagerModelMapper类用来查询维度信息；
7. DimensionDetailModelMapper类用来查询维度的属性信息。

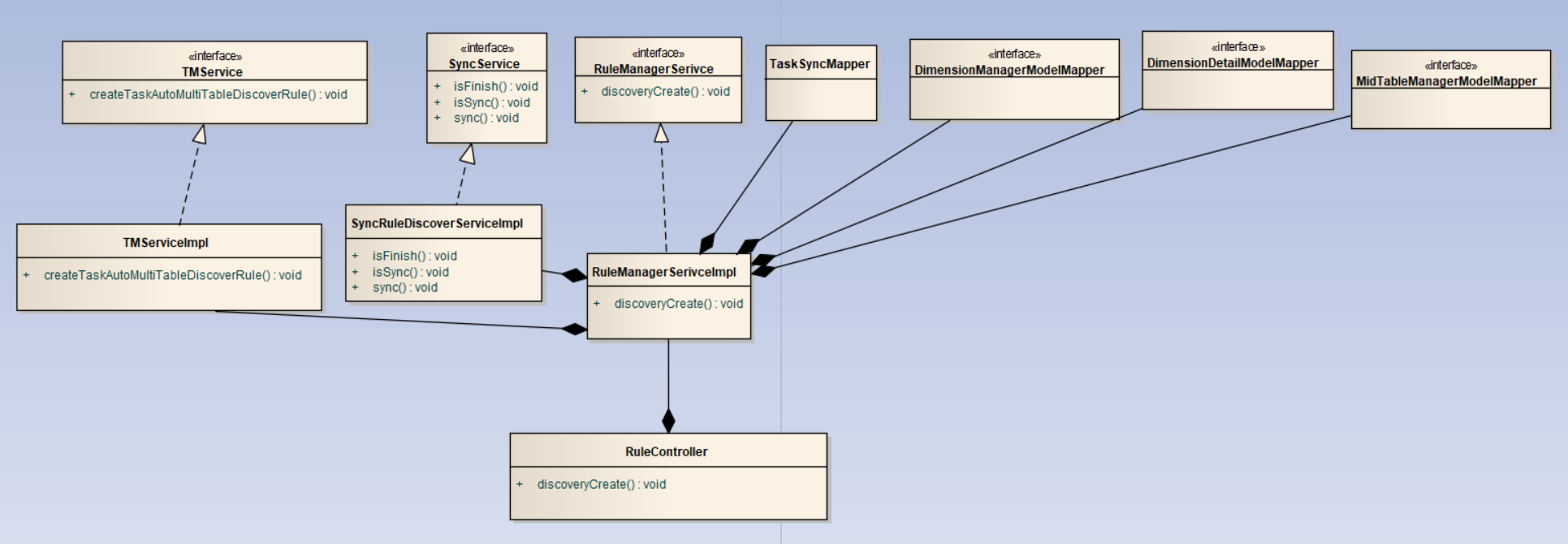
## 创建规则发现任务接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/discovery/create | |
| **入参** | **字段** | **描述** |
|  | cr (string): | 覆盖率 |
|  | ftr (string): | 容错率 |
|  | tableInfos (Array[TableInfoDto]): | 表信息列表 |
| TableInfoDto | dimensionId (string): | 表维度id |
|  | tableName (string): | 表名 |
|  | columnInfos (Array[ColumnInfoDto]): | 表对应的列信息 |
| ColumnInfoDto | columnName (string): | 标准属性名称 |
|  | columnType (string) | 标注属性类型 |
|  | modelId (string): | 模型id , |
|  | modelStorePath (string): | 机器学习模型存储在HDFS的路径 |
|  | skip (string): | 该列是否跳过(不参与)规则发现。ture:跳过 false:不跳过。默认为fasle:不跳过 , |
|  | target (string): | 该列是否是用户规则发现的目标列。true:是 false:否。 不传默认为true:是 |
|  | fkMsg (ForeignKeyDto): | 外键信息 , |
| ForeignKeyDto | fkColumnName (string): | 外键列名 |
|  | fkTableName (string): | 外键表名 |
| **返回** | **字段** | **描述** |
|  | mainTaskId | 主任务id |

### 接口时序图



### 接口类图



类描述：

1. 用户通过RuleController类的discoveryCreate方法发起规则发现请求；
2. RuleManagerServiceImpl类的discoverCreate方法主要用来实现规则发现请求；
3. TMServiceImpl类的createTaskAutoMultiTableDiscoverRule方法用来对TM子系统发起规则发现任务；
4. SyncRuleDiscoverServiceImpl主要同来向TM子系统创建规则发现任务，以及定时轮询任务状态；

5、 TaskSyncMapper用来存储中间表同步任务、HDFS同步任务；

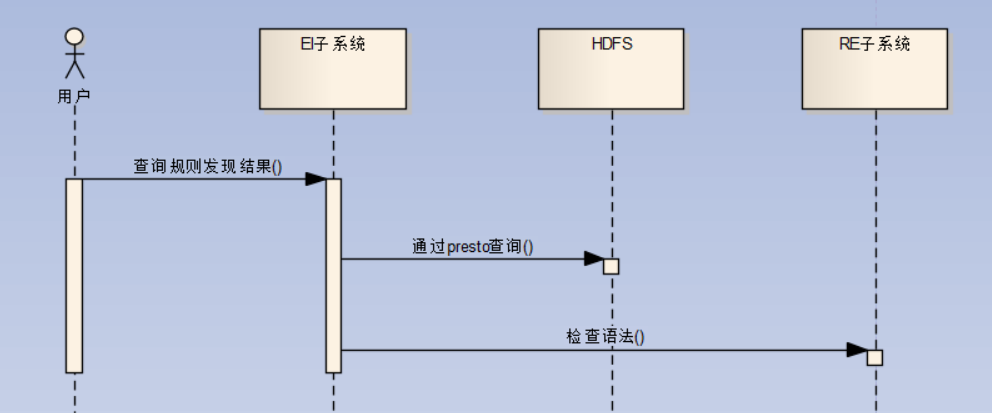
6、 DimensionManagerModelMapper类用来查询维度信息；

7、 DimensionDetailModelMapper类用来查询维度的属性信息。

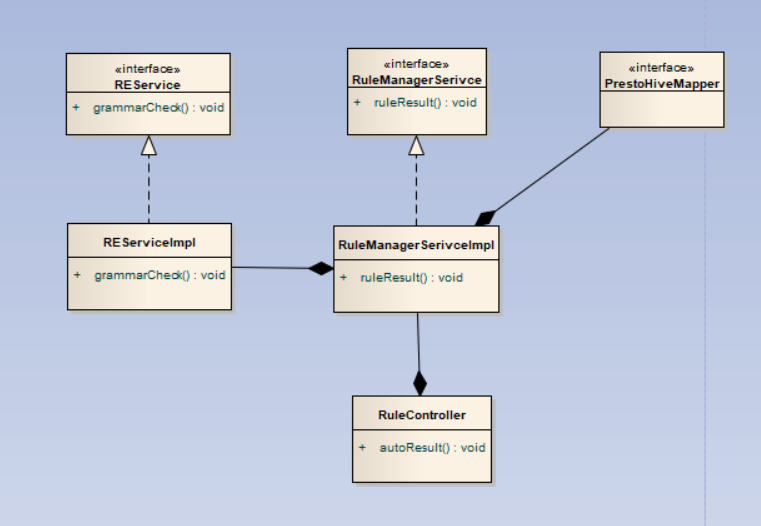
## 规则发现结果查询接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/discovery/result/list | |
| **入参** | **字段** | **描述** |
|  | id | 任务id |
| **返回** | **字段** | **描述** |
|  | id（Integer） | 规则发现的id |
|  | rule（String） | 规则发现的规则内容 |
|  | support（Integer） | 规则发现的支持度 |
|  | unsupport（Integer） | 规则发现的不支持度 |
|  | checkSql（String） | checkSql |
|  | rowsize（Integer） | Rowsize |
|  | ruleExecuteTypeList（List<Integer>） | 规则支持的类型 |

### 接口时序图



### 接口类图



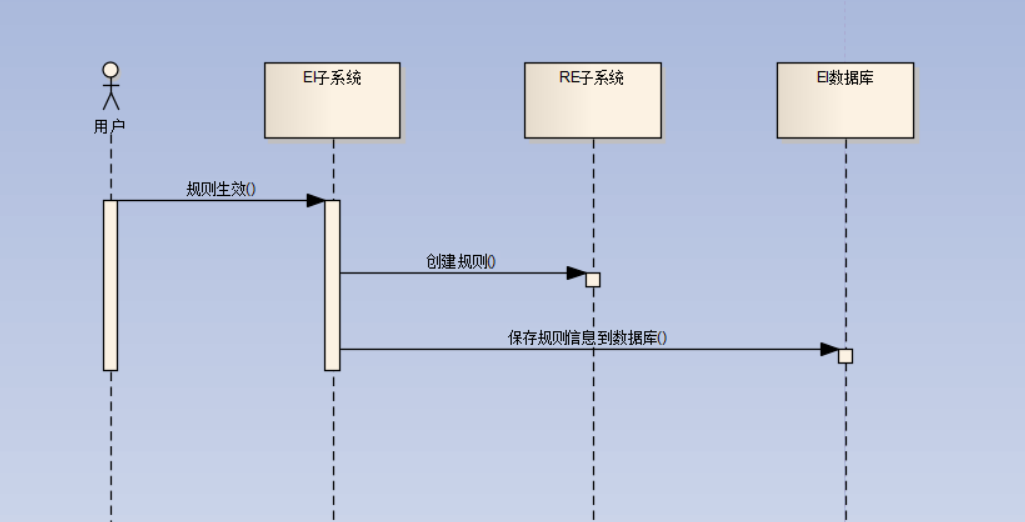
类描述：

1. 用户通过RuleController类的autoResult方法查询规则发现的结果；
2. RuleManagerServiceImpl类主要是实现规则查询的方法；
3. PrestoHiveMapper类主要是用来通过presto查询hdfs上的数据。
4. REServiceImpl类主要用来检查MLS生成的语法，以及语法支持的执行类型。

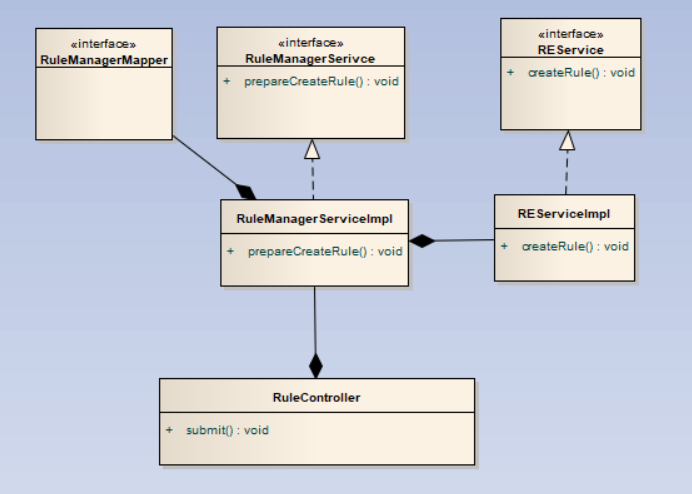
## 规则生效接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/discovery/submit | |
| **入参** | **字段** | **描述** |
|  | id (string): | 任务ID |
|  | ruleTypeDtoList (Array[规则类型], optional): | 规则id |
| 规则类型 | ruleContent (string, optional) | 规则内容 |
|  | ruleExecuteTypes (Array[integer], optional) | 规格支持的类型：10查错，30纠错 ,  ruleId (integer, optional): 规则id |
| **返回** | **字段** | **描述** |
|  | ruleIds(List<String>) | 规则id |

### 接口时序图



### 接口类图



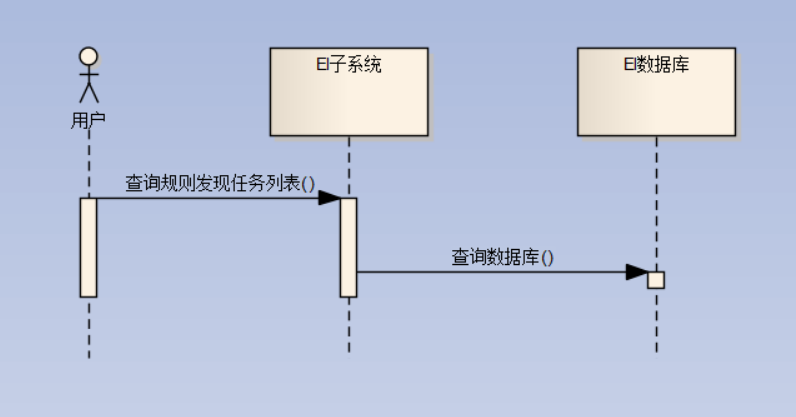
类描述：

1. 用户通过RuleController类的submit方法对规格进行生效；
2. RuleManagerServiceImpl类的prepareCreateRule方法实现规则生效操作；
3. REServiceImpl类用来向RE子系统创建规则；
4. RuleManagerMapper类用来在EI数据库中保存规则。

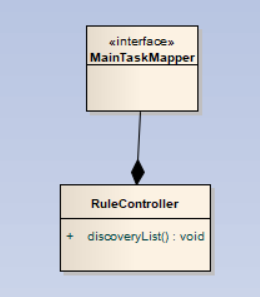
## 规则发现任务列表查询接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/discovery/list | |
| **入参** | **字段** | **描述** |
|  | mainTaskType (integer, optional) | 主任务类型 200-规则发现 300-规则运行 |
|  | pageNum (integer) | 第几页，默认第1页 |
|  | pageSize (integer) | 每页多上行，默认15行 |
| **返回** | **字段** | **描述** |
|  |  |  |

### 接口时序图



### 接口类图



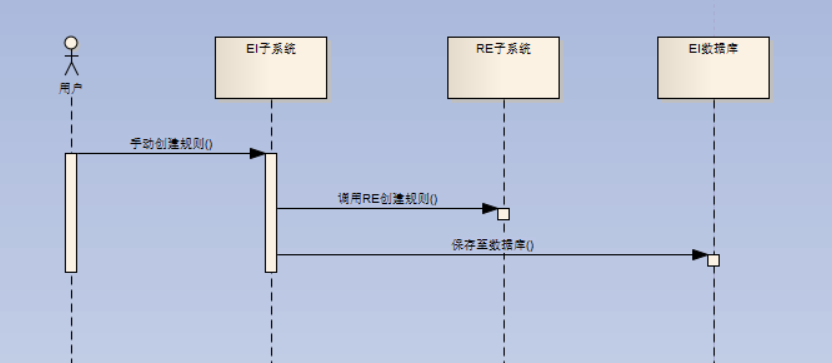
类说明：

1. 用户通过RuleController的discoveryList方法查询规则发现列表；
2. MainTaskMapper类用来查询规则发现任务的列表。

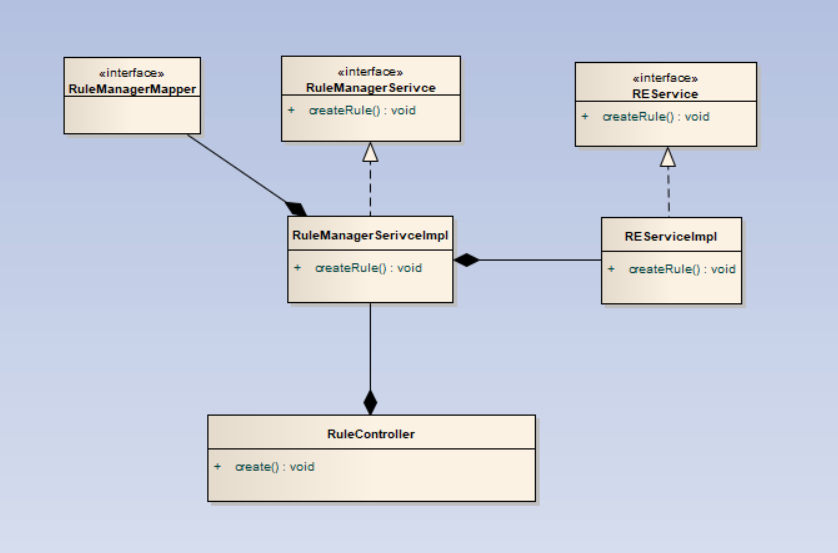
## 手动规则创建接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/create | |
| **入参** | **字段** | **描述** |
|  | ruleContent (string): | 规则内容 |
|  | ruleDesc (string) | 规则描述 |
|  | ruleExecuteTypes (Array[integer]) | 规格支持的类型：10查错，30纠错 |
| **返回** | **字段** | **描述** |
|  | ruleId | 规则id |

### 接口时序图



### 接口类图



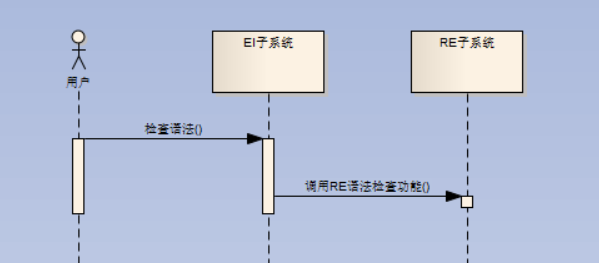
类描述：

1. 用户通过RuleController类的create方法进行手动创建规则；
2. RuleManagerServiceImpl类的createRule方法处理手动规则创建；
3. REServiceImpl类用来与RE子系统对接；
4. RuleManagerMapper类用来处理数据规则在EI数据库的存储。

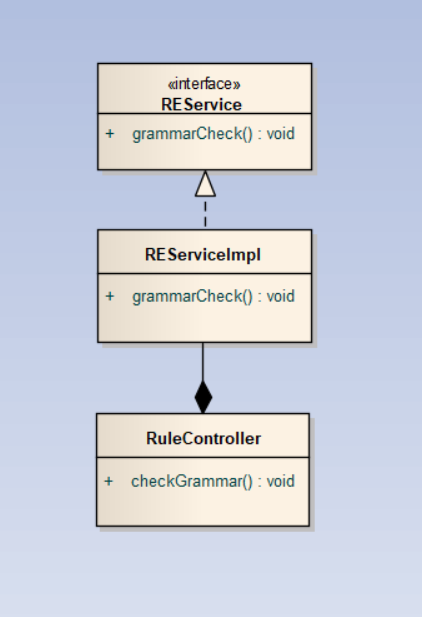
## 语法检查接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/rule/check/grammar | |
| **入参** | **字段** | **描述** |
|  | ruleContent (string): | 规则内容 |
| **返回** | **字段** | **描述** |
|  | tablename（String） | 目标表 |
|  | ruleExecuteTypes（Set<Integer>） | 规则支持的类型 |
|  | errorInfo（String） | 错误信息 |

### 接口时序图



### 接口类图



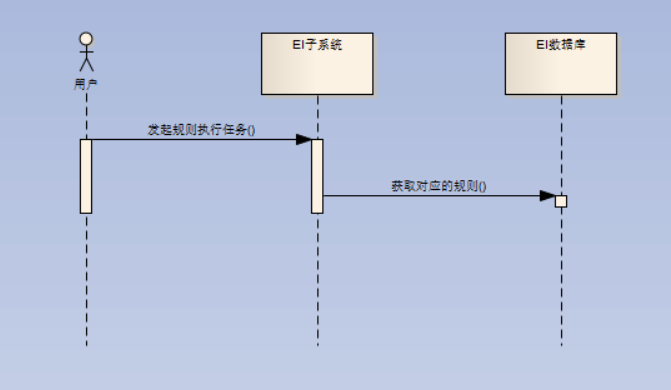
类描述：

1. 用户通过RuleController类的checkGrammar方法对语法进行检查；
2. REServiceImpl的grammarCheck方法向RE子系统发起语法检查功能；

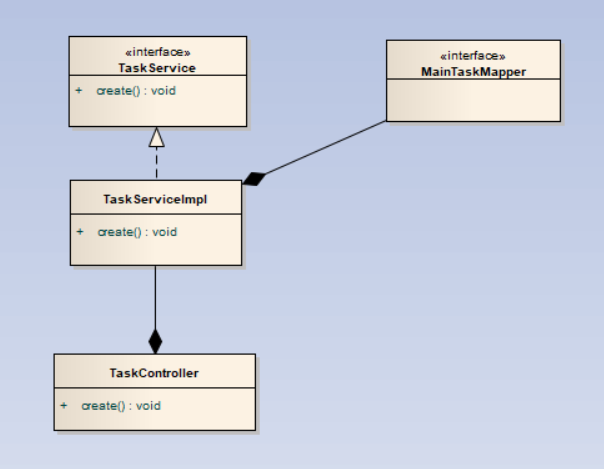
## 创建规则任务执行接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/task/create | |
| **入参** | **字段** | **描述** |
|  | taskName (string) | 任务名称 |
|  | executeType (integer, optional) | 10,查错规则任务; 30,纠错规则任务 |
|  | dimensionIds (Array[string]) | 维度id |
| **返回** | **字段** | **描述** |
|  | mainTaskId | 主任务id |

### 接口时序图



### 接口类图



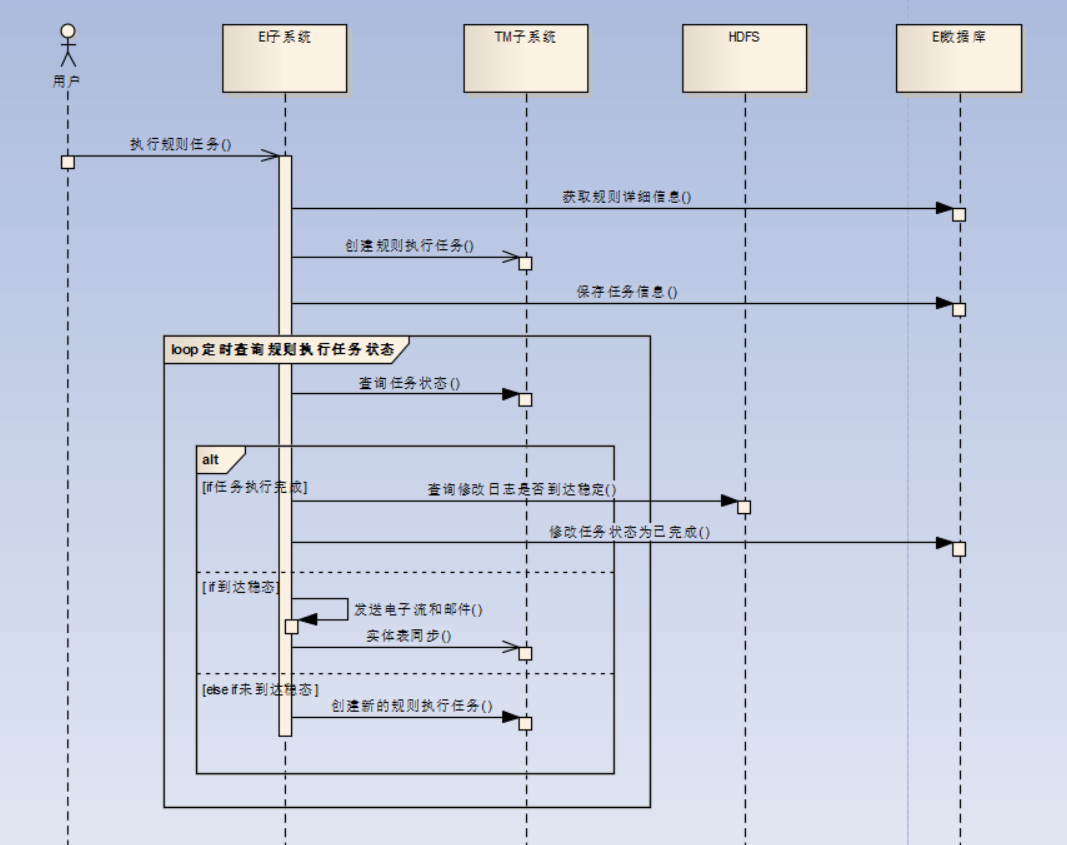
类描述：

1. 用户通过TaskController类的create方法创建规则任务；
2. TaskServiceImpl类的create方法处理规则任务创建；
3. MainTaskMapper类处理数据库。

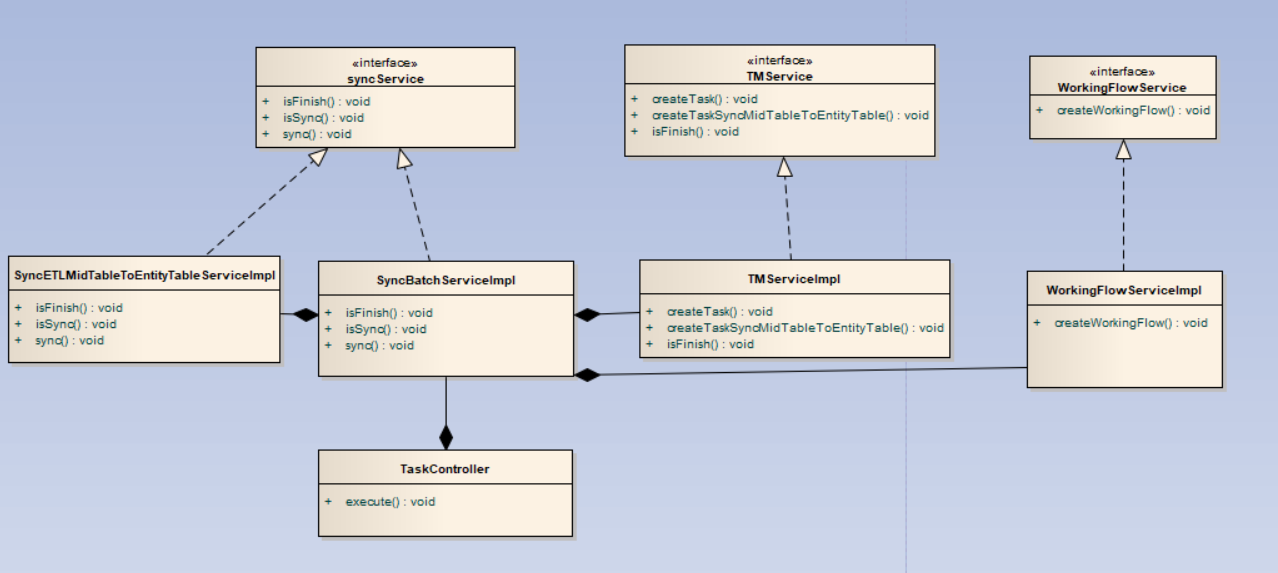
## 规则任务执行接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/task/execute | |
| **入参** | **字段** | **描述** |
|  | id (string, optional), | 规则任务id |
| **返回** | **字段** | **描述** |
|  | mainTaskId | 任务id |

### 接口时序图



### 接口类图



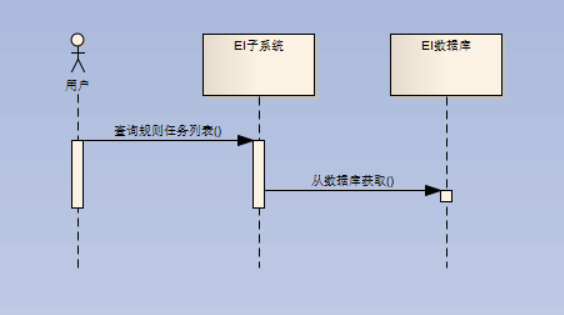
类描述：

1. 用户通过TaskController类的execute方法执行语法；
2. SyncBatchServiceImpl用来处理规则执行，稳态判断，已经定时轮询任务的完成状态；
3. SyncETLMidTableToEntityTableServiceImpl类用来处理规则实体表同步；
4. TMServiceImpl类用来向TM子系统提交任务执行请求；
5. WorkingFlowServiceImpl类用来处理规则执行达到稳态后发起电子流。

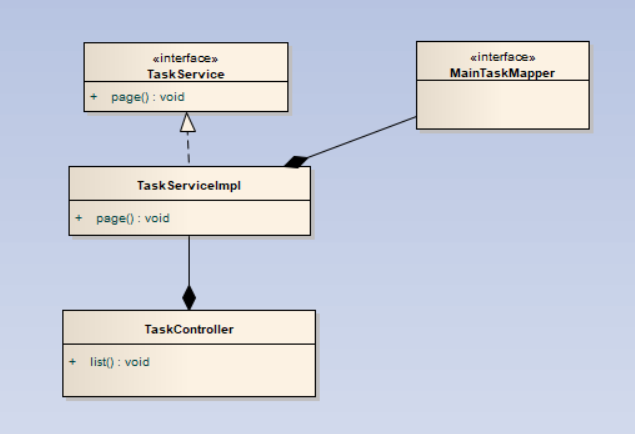
## 规则任务（查错、纠错）列表查询接口

|  |  |  |
| --- | --- | --- |
| **接口** | POST /api/v1/ei/task/list | |
| **入参** | **字段** | **描述** |
|  | pageNum (integer): | 第几页，默认第1页 |
|  | pageSize (integer): | 每页多上行，默认15行 |
|  | executeType (integer) | 10,查错规则任务; 30,纠错规则任务; |
| **返回** | **字段** | **描述** |
|  | Id | 任务id |
|  | taskName | 任务名称 |
|  | createTime | 创建时间 |
|  | taskType | 任务类型，10查错，30纠错 |
|  | status | 状态 |

### 接口时序图



### 接口类图

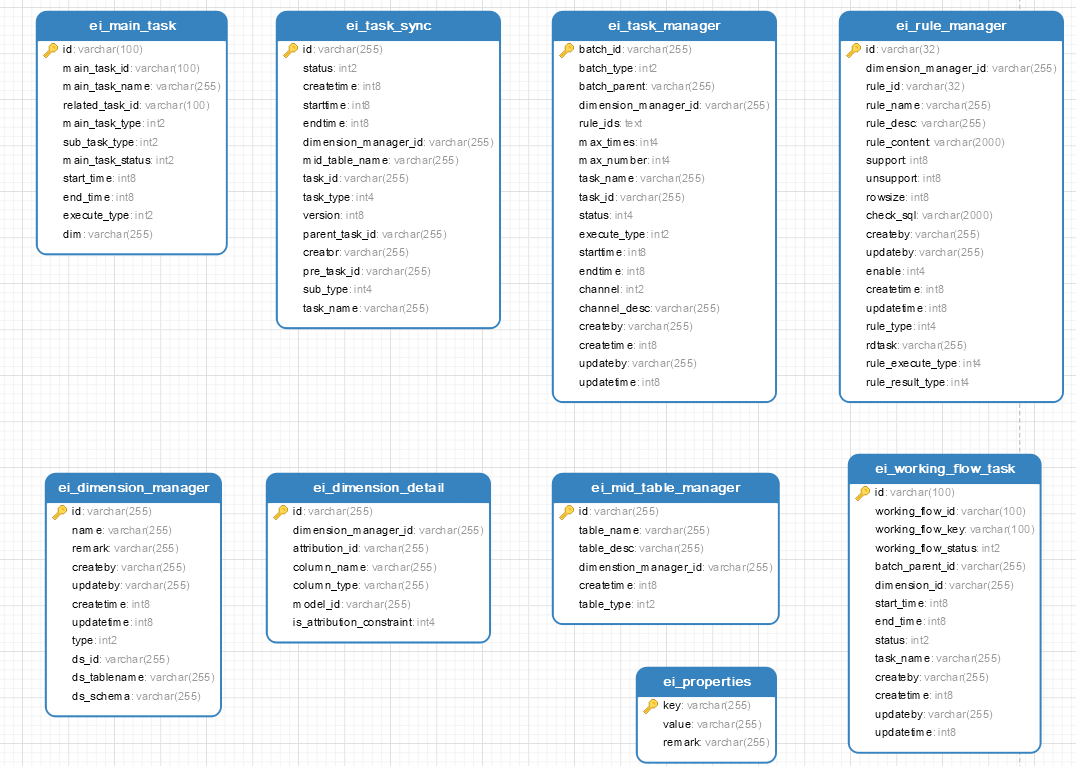


类描述：

1. 用户通过访问TaskController类的list方法获取查错，纠错任务的列表；
2. TaskServiceImpl类的page方法主要用来处理列表的分页查询功能；
3. MainTaskMapper类主要处理数据库的操作。

# 数据库设计

EI子系统数据库模型如下图所示：



数据库表说明：

1. ei\_dimension\_manager表主要用来存储维度：标准属性维度、镜像维度。主要通过type字段进行区分，type=1标准属性维度，type=2镜像维度。
2. ei\_dimension\_detail表主要用来存储维度的属性；
3. ei\_mid\_table\_manager表主要用来存储维度对应的中间表、实体表、可信表名称；
4. ei\_properties表主要用来存储ei-server子系统的key-value配置数据；
5. ei\_working\_flow\_task表主要用来存储电子流信息；
6. ei\_main\_task表主要用来存储主任务表信息；
7. ei\_task\_sync表用来存储中间表同步、hdfs同步、规则挖掘任务；
8. ei\_task\_manager表主要用来存储规则执行任务；
9. ei\_rule\_manager表主要用来存储规则信息。

## 说明

例如：由于存在与外部系统的接口,所以需要考虑访问安全的问题.

## 设计

例如：分为数据传输部分,IP过滤部分,身份验证部分.[章节可补充]

### 数据传输部分

例如：在部分数据传递的时候,考虑以https协议,需要在部署的时候作相关处理.

### IP过滤分部

例如：可在系统前端通过Filter实现,该Filter实现对该地址访问的IP过滤作用.可信任IP地址通过xml文件进行配置.

### 身份验证部分

例如：对信任的用户,颁发身份验证码,通过该标识进行身份识别.

# 系统性能设计

# 系统出错处理

例如：为了在系统出现异常情况下给用户以明确的提示，可采用两种方式予以提示：

1．使用Javascript的alert()函数直接提示，这主要在输入或修改的情况下使用；

2．使用统一的错误界面提示，该界面对应于errorpage.jsp页面。错误界面样式如下图所示：

错误提示界面

# 可靠性设计