**DNA数据库比对管理系统**

**开发说明文档**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作者 | 李志华 | 日期 | 2020.2.10 | 版本 | V 1.0 |
| 修订人 | 修订日期 | 版本 | 修订内容 | | |
| 李志华 | 2020.5.9 | v1.1 | 修改服务组件结构 | | |
|  |  |  |  | | |
|  |  |  |  | | |
|  |  |  |  | | |

# 文档介绍

* 1. 文档目的

明确业务范畴，确定技术方向，定义数据结构、开发结构和相关规则。

* 1. 文档范围

DNA数据库管理及比对系统

* 1. 读者对象
* 项目经理
* 架构师
* 前端开发人员
* 后台开发人员
* 测试人员
* 实施人员

# **产品介绍**

2.1 背景介绍

2.2 开发目的

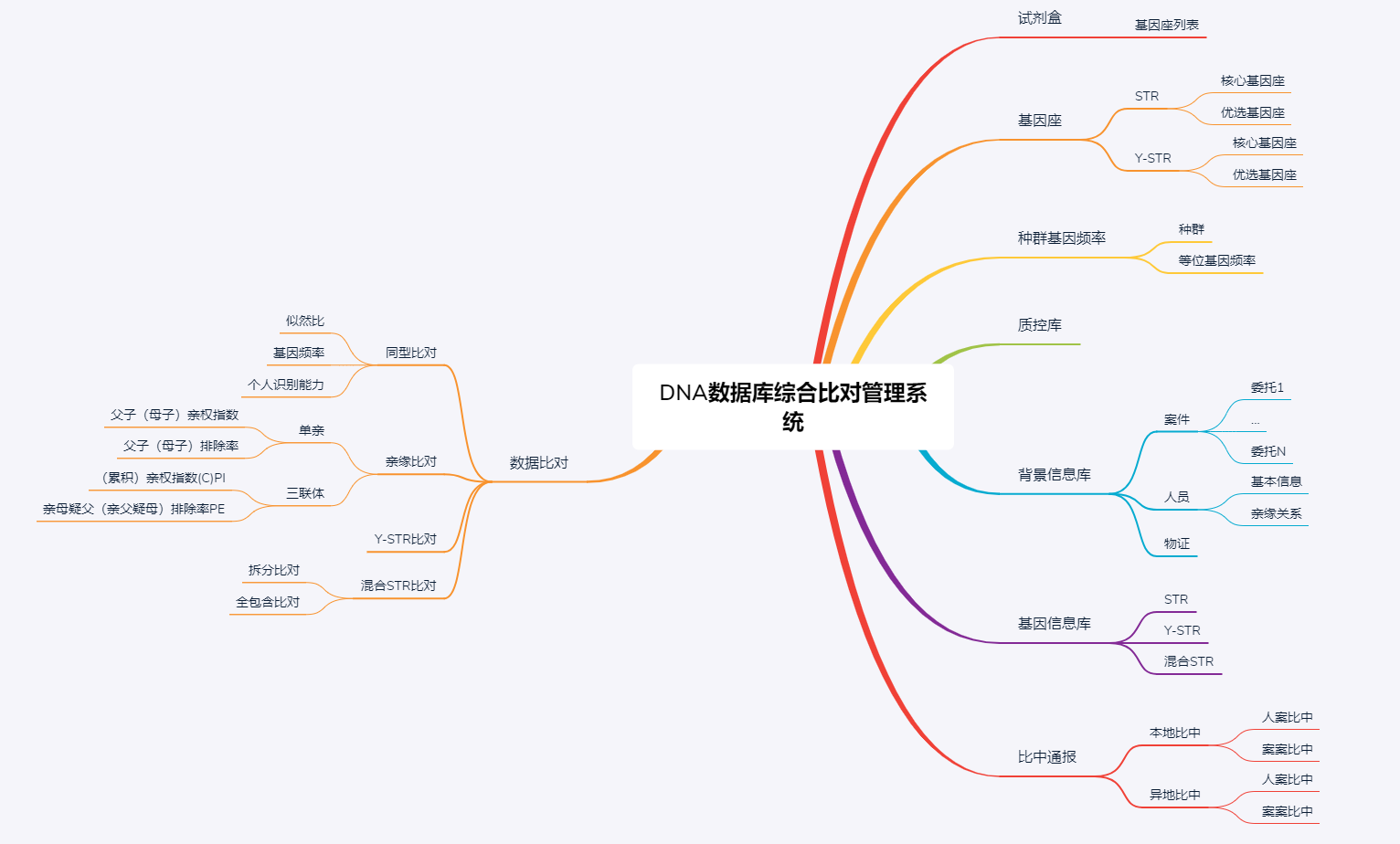
2.3 用户群体

北京市局刑侦总队八支队（法医中心）

分局DNA实验室

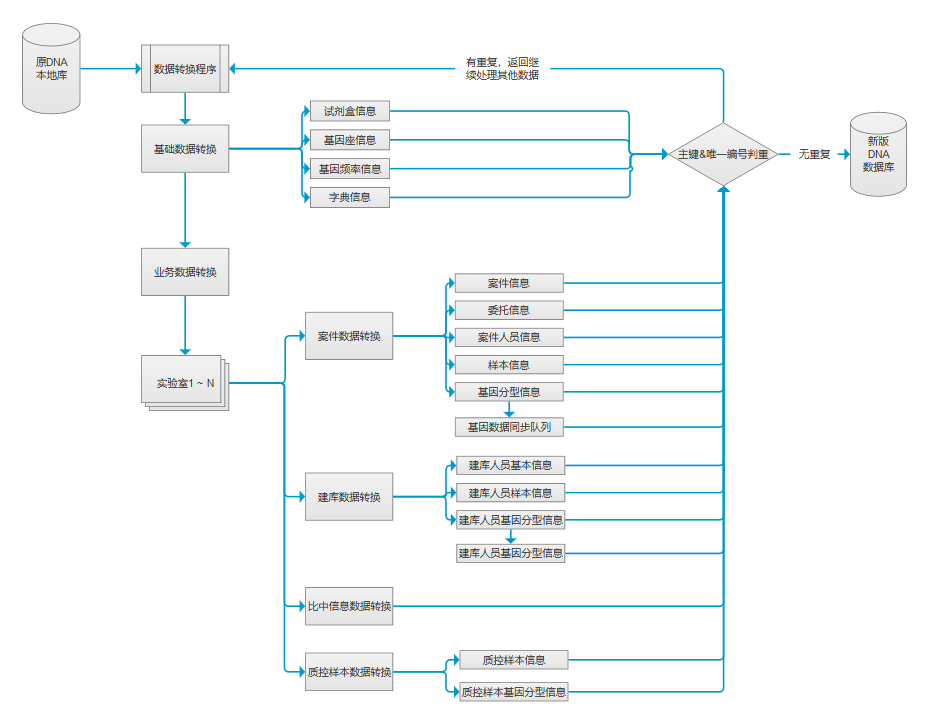
# 功能介绍

系统主要功能图示



# **流程说明**

4.1 数据转换处理流程



# **技术架构**

5.1 开发语言

开发语言选用Java， 使用版本为jdk-8u231.

5.2 数据库

* 关系型数据库

版本为MySQL 5.7.29 / MySQL 8.0，搭建”双主”生产环境。

*<TODO - 附：MySQL 双主 实施文档。>*

* NoSQL数据库

Redis，用于存储基因数据，并进行持久化存储。

*<TODO - 附：Redis安装实施文档。>*

5.3 中间件

Tomcat，版本为8.5.50.

*<TODO - 附：Tomcat安装实施文档， 含内存调优配置等。>*

5.4 接口规范

基于HTTP协议，统一使用RESTful设计风格进行接口开发。

5.5 开发框架

SpringCloud

SpringBoot

Mybatis plus 3.2.0

Alibaba druid-spring 1.1.21

Mysql-connector-java 8.0.19

FastDFS

Log4j2

JSON

lombok

# **开发规范**

6.1 数据库设计规范

6.2 编码规范

**1、常量定义**

6.3 接口规范

6.4 基因信息读写规范

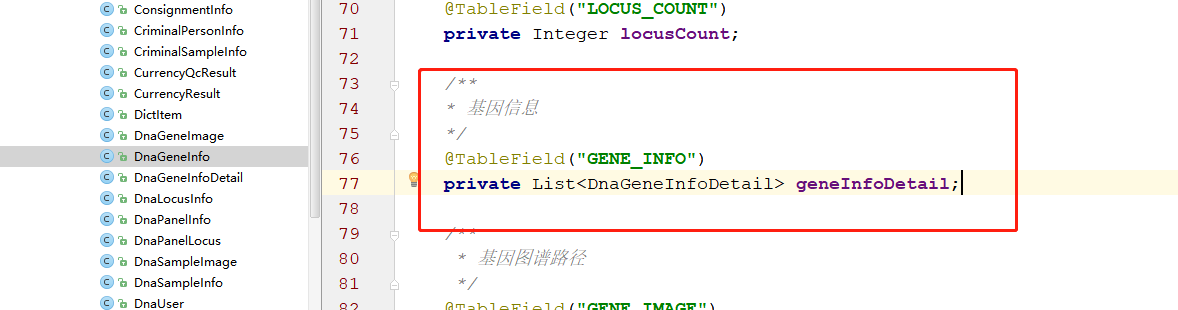
**1、存储格式**

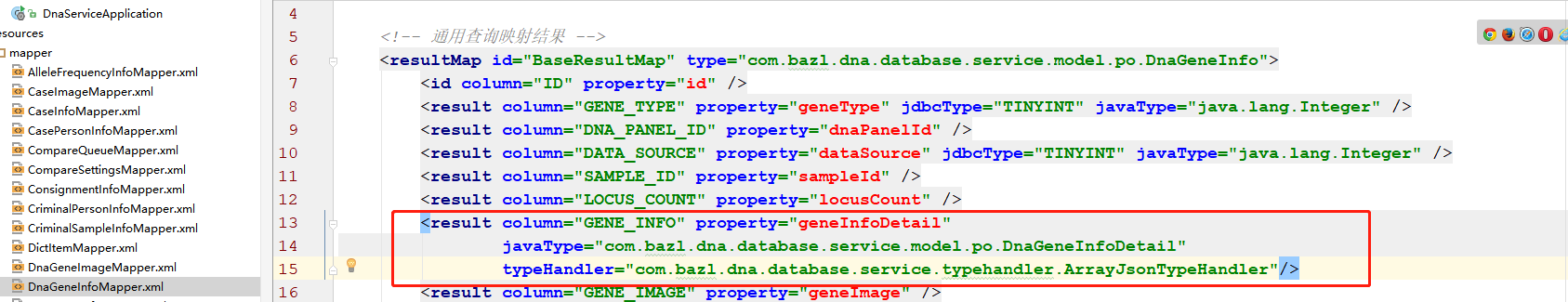
JSON，调用com.alibaba.fastjson转换为JSON字符串进行存储。示例：

[{"name":"D8S1179","value":"14/15"},{"name":"Amel","value":"X/Y"},{"name":"D2S11","value":"11/12"},{"name":"TH01","value":"7/9"},{"name":"vWA","value":"9/9"},...]

1. **String与JSON对象转换方法**

通过TypeHandler实现JSON与对象的转换。





6.5 Redis数据key-value规则

**1、keys规则**

定义该规则用于对写入Redis的数据进行分类，在检索比对时，可通过keys规则检索指定基因类型的数据、或指定入库数据类型的数据、或指定实验室服务器的数据进行匹配。

规则如下：

[基因类型]-[入库数据类型\_亲属关系1\_亲属关系2]-[实验室服务器编号]-[基因数据id]

*示例1、单一个体：现场物证*

*key: 1-01-110101-1234567890*

*1 - 表示“STR”*

*01\_0\_0 - 表示“现场物证”,00表示忽略亲属关系*

*110101 - 表示“东城分局DNA实验室”*

*1234567890 - 表示 “基因数据id”*

*示例2、亲属库：失踪人员亲属*

Key: 1-10\_1\_2-110101-98765432101\_98765432102

1. - 表示“STR”

10\_1\_2 - 表示“失踪人员亲属”，亲属关系分别为 1-父亲，

1. 母亲

110101 - 表示 “东城分局DNA实验室”

98765432101\_98765432102 - 分别表示两个亲属的基因数据id

*注：如果为单亲亲属，则第二个亲属的类型和基因id分别置为0.*

*如：10\_1\_0 - 表示“失踪人员父亲”，无另一亲属*

*98765432101\_0 - 表示失踪人员父亲的基因数据id*

**2、value规则**

* 单一个体数据：存储JSON字符串

*示例、现场物证 - 直接存储基因信息的JSON字符串*

*[{"name":"D8S1179","value":"14/15"},{"name":"Amel","value":"X/Y"},{"name":"D2S11","value":"11/12"},{"name":"TH01","value":"7/9"},{"name":"vWA","value":"9/9"},...]*

* 亲属库数据：存储亲属样本基因数据的JSON 数组

*示例、失踪人口亲属 - 按照key中的顺序分别存储对应的亲属基因信息JSON字符串*

*如：[[{"name":"D8S1179","value":"14/15"},{"name":"Amel","value":"X/Y"},{"name":"D2S11","value":"11/12"},{"name":"TH01","value":"7/9"},{"name":"vWA","value":"9/9"},...],[{"name":"D8S1179","value":"14/15"},{"name":"Amel","value":"X/Y"},{"name":"D2S11","value":"11/12"},{"name":"TH01","value":"7/9"},{"name":"vWA","value":"9/9"},...]]*

# **数据库表结构说明**

|  |  |  |
| --- | --- | --- |
| **序号** | **表名描述** | **表名** |
| 1 | 等位基因频率信息表 | allele\_frequency\_info |
| 2 | 案件照片信息表 | case\_image |
| 3 | 案件基本信息表 | case\_info |
| 4 | 案件人员信息表 | case\_person\_info |
| 5 | 比对队列表 | compare\_queue |
| 6 | 数据比对设置表 | compare\_settings |
| 7 | 委托信息表 | consignment\_info |
| 8 | 建库人员信息表（违法犯罪/前科人员） | criminal\_person\_info |
| 9 | 建库人员样本表 | criminal\_sample\_info |
| 10 | 字典类目信息表 | dict\_item |
| 11 | dna基因信息表 | dna\_gene\_info |
| 12 | DNA基因座信息表 | dna\_locus\_info |
| 13 | 试剂盒信息表 | dna\_panel\_info |
| 14 | 试剂盒基因座关系表 | dna\_panel\_locus |
| 15 | 样本照片信息表 | dna\_sample\_image |
| 16 | DNA样本信息表 （案件样本信息表） | dna\_sample\_info |
| 17 | 基因同步队列 | gene\_sync\_queue |
| 18 | 入库条件信息 | instore\_data\_condition |
| 19 | 入库数据类型表 | instore\_data\_type |
| 20 | 实验室服务器信息 | lab\_server\_info |
| 21 | dna混合比对结果信息表 | match\_result\_mix |
| 22 | 亲缘比对结果表 | match\_result\_relative |
| 23 | 同型比对结果表 | match\_result\_same |
| 24 | 同型比对结果分组信息表 | match\_result\_same\_group |
| 25 | ystr样本比中结果信息表 | match\_result\_ystr |
| 26 | Y-STR比中结果分组信息表 | match\_result\_ystr\_group |
| 27 | 机构信息表 | org\_info |
| 28 | 人员关系表 | person\_relative\_info |
| 29 | 种群基因频率信息表 | population\_frequency\_info |
| 30 | 质控样本信息表 | qc\_sample\_info |
| 31 | 质控污染记录表 | qc\_sample\_pollute\_record |
| 32 | 行政区划信息 | region\_info |
| 33 | 案件数据上报队列 | transfer\_case\_queue |
| 34 | 上报案件样本表 | transfer\_case\_sample |
| 35 | 上报建库人员队列 | transfer\_criminal\_queue |

# **服务组件说明**

**8.1 bazl-dna-database-service**

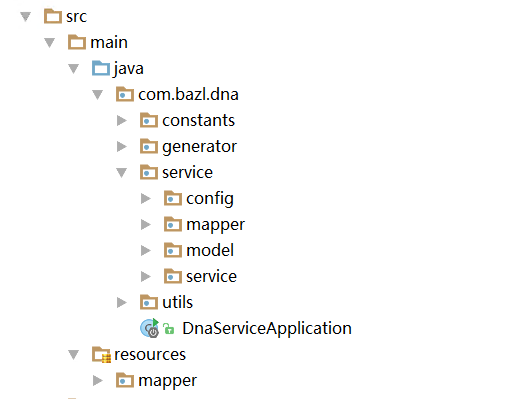
8.1.1 说明

本组件用于业务逻辑处理、数据库操作和事务控制，其中，model层用于定义数据库实体对象；mapper层用于数据库持久化操作，编写sql语句和封装查询结果；service层负责业务逻辑和事务控制，调用mapper（dao）层。

其他组件均通过调用该组件进行数据库交互操作。

8.1.2 Maven信息

<**groupId**>com.bazl.dna</**groupId**>  
<**artifactId**>bazl-dna-database-service</**artifactId**>  
<**version**>0.0.1-SNAPSHOT</**version**>

8.1.3 目录结构

**8.2 bazl-dna-database-core**

8.2.1 说明

作为控制层（Controller）定义RESTful格式的WEB接口，供各实验室DNALims系统调用，进行数据入库、快速比对、基因数据管理等处理。

8.2.2 Maven信息

<parent>

<artifactId>dna-cloud</artifactId>

<groupId>com.bazl.dna</groupId>

<version>1.0.0-SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>bazl-dna-database-core</artifactId>

引用 bazl-dna-database-service：

<dependency>

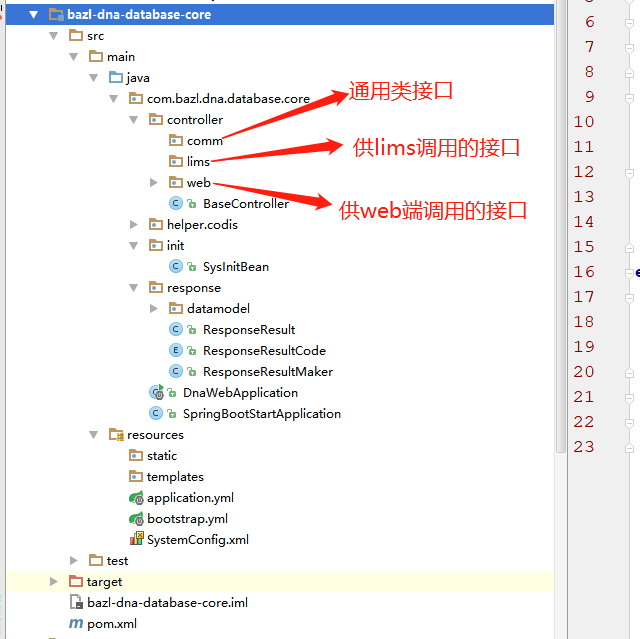
<groupId>com.bazl.dna</groupId>

<artifactId>bazl-dna-database-service</artifactId>

<version>${project.version}</version>

</dependency>

8.2.3 目录结构



8.2.4 接口列表

8.2.4.1 WEB应用接口

package: com.bazl.dna.database.core.controller.web;

8.2.4.1.1 首页 - 查询最新的10条物证比中人员列表

package: com.bazl.dna.database.core.controller.web.home;

Controller: HomeController

接口路径：/home/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.2 首页 - 查询最新的10条物证比中物证列表

package: com.bazl.dna.database.core.controller.web.home;

Controller: HomeController

接口路径：/home/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.3 首页 - 查询本地库数据汇总情况

package: com.bazl.dna.database.core.controller.web.home;

Controller: HomeController

接口路径：/home/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.4 数据综合管理 - 案件数据管理

package: com.bazl.dna.database.core.controller.web.caseinfo;

Controller: CaseInfoController

接口路径：/caseinfo/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.5 数据综合管理 - 建库数据管理

package: com.bazl.dna.database.core.controller.web.criminal;

Controller: CriminalPersonInfoController

接口路径：/criminal/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.6 数据综合管理 - 质控数据管理

package: com.bazl.dna.database.core.controller.web.qc;

Controller: QcSampleInfoController

接口路径：/qc/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.7 比中信息管理 - 同型比中

package: com.bazl.dna.database.core.controller.web.matched;

Controller: SameMatchedController

接口路径：/match/same/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.8 比中信息管理 - 亲缘比中

package: com.bazl.dna.database.core.controller.web.matched;

Controller: RelativeMatchedController

接口路径：/match/relative/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.9 比中信息管理 - Y-STR比中

package: com.bazl.dna.database.core.controller.web.matched;

Controller: YstrMatchedController

接口路径：/match/ystr/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.10 快速比对管理 - 同型比对

package: com.bazl.dna.database.core.controller.web.rapid;

Controller: RapidStrComparisonController

接口路径：/rapid/str/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.11 快速比对管理 - 亲缘比对

package: com.bazl.dna.database.core.controller.web.rapid;

Controller: RapidRelativeComparisonController

接口路径：/rapid/relative/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.12 快速比对管理 - 混合STR比对

package: com.bazl.dna.database.core.controller.web.rapid;

Controller: RapidMixComparisonController

接口路径：/rapid/mix/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.13 快速比对管理 - Y-STR比对

package: com.bazl.dna.database.core.controller.web.rapid;

Controller: RapidYstrComparisonController

接口路径：/rapid/ystr/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.14 基础数据管理 - STR基因座管理

package: com.bazl.dna.database.core.controller.web.panel;

Controller: DnaLocusInfoController

接口路径：/locus/str/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.15 基础数据管理 - Y-STR基因座管理

package: com.bazl.dna.database.core.controller.web.panel;

Controller: DnaLocusInfoController

接口路径：/locus/ystr/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.16 基础数据管理 - 试剂盒管理

package: com.bazl.dna.database.core.controller.web.panel;

Controller: DnaPanelInfoController

接口路径：/panel/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.17 基础数据管理 - 种群基因频率管理

package: com.bazl.dna.database.core.controller.web.frequency;

Controller: PopulationFrequencyInfoController

接口路径：/frequency/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.18 数据监控管理 - 案件上报监控

package: com.bazl.dna.database.core.controller.web.monitor.transfer;

Controller: TransferCaseController

接口路径：/transfer/case/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.19 数据监控管理 - 建库上报监控

package: com.bazl.dna.database.core.controller.web.monitor.transfer;

Controller: TransferCriminalController

接口路径：/transfer/criminal/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.20 数据监控管理 - 数据比对监控

package: com.bazl.dna.database.core.controller.web.monitor.compare;

Controller: CompareMonitorController

接口路径：/monitor/compare/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.21 系统管理 - 入库基因位点数设置

package: com.bazl.dna.database.core.controller.web.system;

Controller: InstoreDataController

接口路径：/system/instore/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.22 系统管理 - STR比对设置

package: com.bazl.dna.database.core.controller.web.system;

Controller: AutoCompareSettingsController

接口路径：/system/compare/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.23 系统管理 - Y-STR比对设置

package: com.bazl.dna.database.core.controller.web.system;

Controller: AutoCompareSettingsController

接口路径：/system/compare/？

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<?>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

8.2.4.1.24 通用类查询接口

8.2.4.1.24.1 行政区划接口

8.2.4.1.24.2 公安机构信息接口

8.2.4.1.24.3 实验室信息查询接口

8.2.4.2 供Lims系统调用接口

package: com.bazl.dna.database.core.controller.lims

8.2.4.3 通用接口

package: com.bazl.dna.database.core.controller.comm

8.2.4.3.1 字典接口

package: com.bazl.dna.database.core.controller.comm.dict;

Controller: DictItemController

接口路径：/comm/dict/listByTypeCode?dictTypeCode=xxx

请求方法：GET or POST

请求参数：String dictTypeCode

返回数据：ResponseResult<List<DictItem>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “success”

“data”: []

}

字典类型列表：

|  |  |  |
| --- | --- | --- |
| 序号 | 字典类型名称 | 字典类型代码 |
| 1 | CASE\_PERSON\_TYPE | 人员类型 |
| 2 | CASE\_PROPERTY | 案件性质 |
| 3 | CASE\_TYPE | 案件类型 |
| 4 | CERTIFICATE\_TYPE | 证件类型 |
| 5 | COMPARE\_MODE | 比对模式 |
| 6 | CRIMINAL\_PERSON\_TYPE | 建库人员类别 |
| 7 | DELETE\_FLAG | 删除标识 |
| 8 | GENE\_SYNC\_STATUS | 基因同步队列状态 |
| 9 | GENE\_SYNC\_TYPE | 基因同步队列类型 |
| 10 | GENE\_TYPE | 基因类型 |
| 11 | GROUP\_TYPE | 同型比对：分组类型 |
| 12 | MEMBER\_TYPE | 人员类型 |
| 13 | PERSON\_GENDER | 性别 |
| 14 | PERSON\_NATION | 国籍 |
| 15 | PERSON\_RACE | 民族 |
| 16 | PERSON\_RELATION\_TYPE | 人员亲缘身份 |
| 17 | QC\_PERSON\_TYPE | 质控人员类型 |
| 18 | QC\_SAMPLE\_TYPE | 质控样本类型 |
| 19 | SAMPLE\_TYPE | 样品类型 |
| 20 | TRANSFER\_NATION\_FLAG | 入国家库状态 |

8.2.4.3.2 入库数据类型接口

package: com.bazl.dna.database.core.controller.comm.instore;

Controller: InstoreDataManageController

接口路径：/comm/instore/instoreDataTypeList

请求方法：GET or POST

请求参数：无

返回数据：ResponseResult<List<InstoreDataType>>

数据类型：JSON

示例：

{

“code”: 200,

“msg”: “successc

“data”: [

{

“id“:“主键id“,

“typeCode“:“类型代码“,

“typeName“:“类型名称“,

“nationTypeCode“:“国家库类型代码“,

“nationTypeName“:“国家库类型名称“,

},...

]

}

8.2.4.1 种群频率接口

* 查询所有的种群频率方案集合

接口路径：/populationFrequency/list

请求方法：GET

请求参数：无

返回数据：ResponseResult<List<PopulationFrequencyInfo>>

数据类型：JSON

*示例：*

*{*

*“code”: 200,*

*“msg”: ”success”*

*“data”: [*

*{*

*“id”:1,*

*“defaultFlag”: 1,*

*“populationName”:”中国人群STR”,*

*“geneType”:”1”*

*...*

*},*

*...]*

*}*

* 根据基因类型查询默认的种群频率方案

接口路径：/populationFrequency/defaultOfGeneType

请求方法：GET

请求参数：String geneType

返回数据：ResponseResult<PopulationFrequencyInfo>

PopulationFrequencyInfo对象中包含对应的等位基因频率集合， 即：List<AlleleFrequency> alleleFrequencyInfoList

数据类型：JSON

*示例：*

*{*

*“code”: 200,*

*“msg”: ”success”*

*“data”: {*

*“id”:1,*

*“defaultFlag”: 1,*

*“populationName”:”中国人群STR”,*

*“geneType”:”1”*

*...*

*“*alleleFrequencyInfoList”:[{}]

*}*

*}*

* 根据种群id获取等位基因频率列表

接口路径：/listAlleleFrequency/{populationId}

请求方法：GET

请求参数：Integer populationId

返回数据：ResponseResult<<List<AlleleFrequencyInfo>>

返回匹配当前种群频率id的等位基因频率集合

数据类型：JSON

*示例：*

*{*

*“code”: 200,*

*“msg”: ”success”*

*“data”: [*

*{*

*“id”:1,*

*“populationFrequencyId”: 1,*

*“locusId”:1,*

*“alleleValue”:”13.3”,*

*“probability”: 0.0003*

*},*

*...*

*]*

*}*

8.2.4.2 Panel与基因座接口

* 查询指定panelType的试剂盒列表

接口路径：/panel/listByType/{panelType}

请求方法：GET

请求参数：String panelType

返回数据：ResponseResult<<List<DnaPanelInfo>>

返回指定类型（STR或Y-STR）的试剂盒集合。

数据类型：JSON

*示例：*

*{*

*“code”: 200,*

*“msg”: ”success”*

*“data”: [*

*{*

*“id”:1,*

*“panelType”: “1”,*

*“panelCode”:”ID-Plus”,*

*“panelName”:”Identifiler Plus”,*

*...*

*}, ...*

*]*

*}*

* 根据panelId查询符合条件的基因座有序集合

接口路径：/panel/locusListByPanel

请求方法：GET

请求参数：Integer panelId

返回数据：ResponseResult<<List<DnaLocusInfo>>

返回指定试剂盒id的基因座集合。

数据类型：JSON

*示例：*

*{*

*“code”: 200,*

*“msg”: ”success”*

*“data”: [*

*{*

*“id”:1,*

*“locusType”: “1”,*

*“locusName”:”D8S1179”,*

*“nationalLocusName”:”D8S1179”,*

*“locusAlias”:”D8S1179”*

*...*

*}, ...*

*]*

*}*

* 保存panel信息

接口路径：/panel/savePanelOnly

请求方法：POST

请求参数：DnaPanelInfo dnaPanelInfo

dnaPanelInfo对象中包含基因座列表 dnaLocusInfoList.

返回数据：ResponseResult<String>

返回是否保存成功的信息。

数据类型：JSON

*示例：*

*{ “code”: 200,“msg”: ”success”,“data”: null }*

* 保存panel信息和对应的基因座集合

接口路径：/panel/savePanelAndLocus

请求方法：POST

请求参数：DnaPanelInfo dnaPanelInfo

dnaPanelInfo对象中包含基因座列表 dnaLocusInfoList.

返回数据：ResponseResult<String>

返回是否保存成功的信息。

数据类型：JSON

*示例：*

*{“code”: 200,“msg”: ”success”,“data”: null }*

8.2.4.3 系统设置类数据接口

* 查询所有的入库数据类型

接口路径：/instore/listAllTypes

请求方法：GET

请求参数：无

返回数据：ResponseResult<<List<InstoreDataType>>

返回所有的入库数据类型集合。

数据类型：JSON

*示例：*

*{*

*“code”: 200,*

*“msg”: ”success”*

*“data”: [*

*{*

*“id”:1,*

*“typeCode”: “01”,*

*“typeName”:”现场物证”,*

*...*

*}, ...*

*]*

*}*

8.2.4.4 数据入库接口

8.2.4.5 快速比对接口

1同行比对

接口路径：/repidComparison/strCompare

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 描述 | 类型 | 是否必须 |
| sampleNo | 样本编号 | String | 否 |
| sampleName | 样本名称 | String | 否 |
| instoreDataType | 入库数据类型 | String | 否 |
| labServerInfo | 比对实验室范围 | String | 否 |
| lowestSameLimit | 匹配下线 | String | 否 |
| mostDiffLimit | 容差上线 | String | 否 |
| geneInfo | 基因信息 | String | 否 |
| populationName | 种群数据 | String | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dataList": {

"strCompare": [

{

"caseNo": "案件编号",

"labServerNo": "比对实验室名称",

"instoreDataType": "入库类型",

"sampleNo": "检材编号",

"caseName": "检材名称",

"matchLocusCount": "匹配位点个数",

"PI": "似然率",

}

],

}

},

"errorMessage": "成功"

}}

2亲缘比对

接口路径：/repidComparison/kinshipCompare

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 描述 | 类型 | 是否必须 |
| sampleNo | 样本编号 | String | 否 |
| sampleName | 样本名称 | String | 否 |
| instoreDataType | 入库数据类型 | String | 否 |
| labServerInfo | 比对实验室范围 | String | 否 |
| lowestSameLimit | 匹配下线 | String | 否 |
| mostDiffLimit | 容差上线 | String | 否 |
| motherGeneInfo | 母亲基因信息 | String | 否 |
| populationName | 种群数据 | String | 否 |
| fatherGeneInfo | 父亲基因信息 | String | 否 |
| childGeneInfo | 孩子基因信息 | String | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dataList": {

"strCompare": [

{

"caseNo": "案件编号",

"labServerNo": "比对实验室名称",

"instoreDataType": "入库类型",

"sampleNo": "检材编号",

"sampleName": "检材编号",

"caseName": "案件名称",

"matchLocusCount": "匹配位点个数",

"PI": "似然率",

}

],

}

},

"errorMessage": "成功"

}}

3 ystr快速比对接口

接口路径：/repidComparison/ystrCompare

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 描述 | 类型 | 是否必须 |
| sampleNo | 样本编号 | String | 否 |
| sampleName | 样本名称 | String | 否 |
| instoreDataType | 入库数据类型 | String | 否 |
| labServerInfo | 比对实验室范围 | String | 否 |
| lowestSameLimit | 匹配下线 | String | 否 |
| mostDiffLimit | 容差上线 | String | 否 |
| GeneInfo | 母亲基因信息 | String | 否 |
| populationName | 种群数据 | String | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dataList": {

"strCompare": [

{

"caseNo": "案件编号",

"labServerNo": "比对实验室名称",

"instoreDataType": "入库类型",

"sampleNo": "检材编号",

"sampleName": "检材编号",

"caseName": "案件名称",

"matchLocusCount": "匹配位点个数",

"PI": "似然率",

}

],

}

},

"errorMessage": "成功"

}}

8.2.4.6 查询比中结果接口

8.2.4.7 获取首页信息接口

接口路径：/currency/gethomePageInfo

请求方法：POST

请求参数：无

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dataList": {

（物证比人员）

"matterComparePersonList": [

{

"id": "主键",

"caseAcceptNo": "案件受理编号",

"caseName": "案件名称",

"caseUncover": "案件破获状态",

"labServerNo": "送检单位",

"sampleNo": "检材编号",

"sampleName": "检材名称",

"sampleType": "检材类型",

"matchRegion": "比中地区",

"persionType": 人员类型,

"personname":人员姓名,

"personIdcardNo": 身份证号,

"matchLocusCount": "比中位点个数"

}

],

（物证比物证）

"matterCompareMatter": [

{

"id": "主键",

"caseAcceptNo": "案件受理编号",

"caseName": "案件名称",

"": "案件破获状态",

"labServerNo": "送检单位",

"SampleNo": "检材编号",

"sampleName": "检材名称",

"matchRegion": "比中地区",

"persionType": 人员类型,

"personname":人员姓名,

"matchSampleNo": 比中检材编号,

"matchSampleName": 比中检材名称,

"matchLocusCount": "比中位点个数"

}

],

(待复核比中)

"toReviewCompare": {

"matterComparePersonCount": "物证比中人员",

"regionCompare": "本区比中",

"spanningArea": "跨区比中",

"matterComparematterCount": "物证比中物证",

"matterSameRegionCount": "本区比中",

"matterSpanningAreaCount": "跨区比中"

} ,

（待上报数据）

"toReportInfo": {

"toReportCaseCount": "待上报案件数",

"toReportMatterCount": "待上报物证数",

"toReportPersionCount": "带上报人员数"

},

（本地库数据情况）

"comprehensiveInfo": {

"caseCount": "案件数",

"matterCount": "物证数",

"personCount": "人员数",

"strCount": "str信息数",

"ystrCount": "ystr信息数",

"blendCount": "混合str信息数",

},

}

},

"errorMessage": "成功"

}}

8.2.4.8 获取案件信息接口

接口路径：/currency/currencyCaseQuery

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 描述 | 类型 | 是否必须 |
| offset | 起始页书 | Int | 否 |
| size | 每页条数 | Int | 否 |
| caseName | 案件名称 | String | 否 |
| caseType | 案件类型 | String | 否 |
| caseAcceptNo | 案件受理编号 | String | 否 |
| caseProperty | 案件性质 | String | 否 |
| sysXkNo | 案件现勘k号 | String | 否 |
| scenePlace | 案件地点 | String | 否 |
| sysCaseAno | 案件A号 | String | 否 |
| occurrenceStratDatetime | 案发开始时间 | Date | 否 |
| occurrenceEndDatetime | 案发结束时间 | Date | 否 |
| nationSysNo | 国家库案件编号 | String | 否 |
| isLifeCase | 是否命案 0 是 1不是 | String | 否 |
| consignOrgNameString | 委托单位 | String | 否 |
| consignPersonName | 委托人 | String | 否 |
| consignPersonPhone | 委托人电话 | String | 否 |
| sampleName | 检材名称 | String | 否 |
| sampleType | 检材类型 | String | 否 |
| sampleEvidenceNo | 现勘物证编号 | String | 否 |
| personType | 人员类型 | String | 否 |
| peresonName | 人员名称 | String | 否 |
| personIdcardNo | 人员身份证号 | String | 否 |
| personSex | 人员性别 1男 2女 | String | 否 |
| personRace | 民族 | String | 否 |
| serverName | 实验室名称 | String | 否 |
| acceptStartDate | 受理开始时间 | Date | 否 |
| acceptEndDate | 受理结束时间 | Date | 否 |
| acceptName | 受理人 | String | 否 |
| caseNameContain | 案件名称是否包含 0包含 1等于 | Int | 否 |
| caseAcceptNoContain | 案件受理编号是否包含 0包含 1等于 | Int | 否 |
| sysXkNoContain | 案件现场k号是否包含 0包含 1等于 | Int | 否 |
| sysCaseAnoContain | 案件A号是否包含 0包含 1等于 | Int | 否 |
| nationSysNoContain | 国家库案件编号是否包含 0包含 1等于 | Int | 否 |
| sampleNameContain | 建材名称是否包含 0包含 1等于 | Int | 否 |
| sampleEvidenceNoContain | 现场物证编号是否包含 0包含 1等于 | Int | 否 |
| peresonNameContain | 人员名称是否包含 0包含 1等于 | Int | 否 |
| personIdcardNoContain | 人员身份证是否包含 0包含 1等于 | Int | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dataList": {

（物证比人员）

"matterComparePersonList": [

{

"id": "主键",

"caseAcceptNo": "案件受理编号",

"caseName": "案件名称",

"sysXkNo": "K号",

"sysCaseAno": "A号",

"caseProperty": "案件性质",

"consignOrgNameString": "委托单位",

"acceptOrgNameString": "受理单位",

"acceptName": "受理人",

"": "破获状态",

}

],

}

},

"errorMessage": "成功"

}}

8.2.4.8 获取质控信息接口

接口路径：/currency/currencyCaseQuery

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 描述 | 类型 | 是否必须 |
| offset | 起始页书 | Int | 否 |
| size | 每页条数 | Int | 否 |
| qcSampleType | 质控样本类别 | String | 否 |
| qcSampleNo | 质控样本编号 | String | 否 |
| qcPersonType | 质控人员类别 | String | 否 |
| qcSampleName | 质控样本名称 | String | 否 |
| qcPersonName | 质控人员名称 | String | 否 |
| qcPersonIdcardNo | 人员身份证号 | String | 否 |
| qcPersonOrgName | 所属单位名称 | String | 否 |
| sex | 人员性别 1男 2女 | Int | 否 |
| qcSampleNoContain | 质控样本编号是否包含 0包含 1等于 | Int | 否 |
| qcPersonNameContain | 质控人员名称是否包含 0包含 1等于 | Int | 否 |
| qcSampleNameContain | 质控岩本名称是否包含 0包含 1等于 | Int | 否 |
| qcPersonIdcardNoContain | 质控人身份证是否包含 0包含 1等于 | Int | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dataList": {

（物证比人员）

"matterComparePersonList": [

{

"id": "主键",

"caseAcceptNo": "案件受理编号",

"caseName": "案件名称",

"sysXkNo": "K号",

"sysCaseAno": "A号",

"caseProperty": "案件性质",

"consignOrgNameString": "委托单位",

"acceptOrgNameString": "受理单位",

"acceptName": "受理人",

"": "破获状态",

}

],

}

},

"errorMessage": "成功"

}}

8.2.4.9 str基因信息管理接口

接口路径：/repidComparison/strAlleleQuery

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| name | 基因座名称 | String | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dnaLocusInfoList": [

{

"locusName": "基因座名称",

"otherName": "其他名称",

"coreLocusFlag": "核心基因座",

"valueScope": "基因座取值范围",

"locusOrd": "基因座顺序",

}

],

},

"errorMessage": "成功"

}}

8.2.5.0 ystr基因信息管理接口

接口路径：/repidComparison/ystrAlleleQuery

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| name | 基因座名称 | String | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dnaLocusInfoList": [

{

"locusName": "基因座名称",

"otherName": "其他名称",

"coreLocusFlag": "核心基因座",

"valueScope": "基因座取值范围",

"locusOrd": "基因座顺序",

}

],

},

"errorMessage": "成功"

}}

8.2.5.1 试剂盒信息管理接口

接口路径：/repidComparison/panelInfoQuery

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| panelName | 试剂盒名称 | String | 否 |
| panelType | 试剂盒类型 | String | 否 |
| aliasName | 试剂盒别名 | String | 否 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"panelInfoQueryList": [

{

"panelName": "试剂盒名称",

"locusOrd": "基因座顺序",

"coreLocusFlag": "核心基因座",

}

],

},

"errorMessage": "成功"

}}

8.2.5.2 查询案件性质

接口路径：/repidComparison/caseNatureQuery

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dictItems": [

{

"dictName": "案件性质",

}

],

},

"errorMessage": "成功"

}}

8.2.5.2 查询检材类型

接口路径：/repidComparison/sampleTypeQuery

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dictItems": [

{

"dictName": "检材类型",

}

],

},

"errorMessage": "成功"

}}

8.2.5.2 查询人员类型

接口路径：/repidComparison/personTypeQuery

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": [

{

"dictName": "检材类型",

}

],

"errorMessage": "成功"

}}

8.2.5.3 委托单位名称查询

接口路径：/repidComparison/consignOrgName

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": [

{

"consignOrgName": "委托单位名称",

}

],

},

"errorMessage": "成功"

}}

8.2.5.4 入库类型查询

接口路径：/repidComparison/instorDatetypeQuery

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"instoreDataTypes": [

{

"typeName": "入库数据类型",

}

],

},

"errorMessage": "成功"

}}

8.2.5.5 种群数据类型查询

接口路径：/repidComparison/populationQuery

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"populationFrequencyInfos": [

{

"populationName": "种群名称",

}

],

},

"errorMessage": "成功"

}}

8.2.5.5 查询试剂盒名称

接口路径：/repidComparison/panelNameQuery

请求方法：POST

请求参数：

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dnaPanelInfos": [

{

"panelName": "试剂盒名称",

}

],

},

"errorMessage": "成功"

}}

8.2.5.6 str复核标记

接口路径：/repidComparison/updateStrReviewFlag

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| id | id | String | 是 |
| reviewFlag | 1确认比中 2 解除关联 | String | 是 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": { },

"errorMessage": "成功"

}}

8.2.5.7 ystr复核标记

接口路径：/repidComparison/updateYstrReviewFlag

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| id | id | String | 是 |
| reviewFlag | 1确认比中 2 解除关联 | String | 是 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": { },

"errorMessage": "成功"

}}

8.2.5.8 亲缘复核标记

接口路径：/repidComparison/updateRelativeReviewFlag

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| id | id | String | 是 |
| reviewFlag | 1确认比中 2 解除关联 | String | 是 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": { },

"errorMessage": "成功"

}}

8.2.5.9 修改基因座管理数据

接口路径：/repidComparison/updateStrGeneInfo

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| id | id | String | 是 |
| locusName | 基因座名称 | String | 是 |
| locusAlias | 别名 | String | 是 |
| valueScope | 取值范围 | String | 是 |
| coreLocusFlag | 是否为核心基因座 |  | 是 |
| remark | 备注 | String | 是 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": { },

"errorMessage": "成功"

}}

8.2.6.0 试剂盒管理查看详情

接口路径：/repidComparison/updatepanelName

请求方法：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| id | id | String | 是 |

协议响应：

{

"code": 200,

"errorCode": 0,

"data": {

"dnaPanelInfos": [

{

"panelName": "试剂盒名称",

"aliasName": "试剂盒别名",

"panelCode": "试剂盒编号",

"panelType": "试剂盒类型",

}

],

},

"errorMessage": "成功"

}}

**8.3 bazl-dna-database-sync**

8.3.1 说明

用于将新增、修改、删除的样本基因数据同步到缓存数据库Redis中。

8.3.2

(备忘：亲属库写入同步队列时，只写入一条队列数据即可)

**8.4 bazl-dna-database-compare**

8.4.1 说明

处理数据比对定时任务的组件，以线程池调度的方式处理不同类型的数据比对任务，包括：同型自动比对、亲缘自动比对、Y-STR自动比对、快速比对、混合比对等；比对目标数据从缓存数据库Redis中获取。

**8.5 bazl-dna-database-transfer**

8.5.1 说明

处理数据上报国家库定时任务的组件，上报数据包括案件数据和建库人员数据。由于国家库上报工具在一台服务器上只能部署一个的约束，该组件部署时需分别在各分局实验室服务器上进行部署。