# 一、puri-api 项目规范

1、请求路径禁止驼峰命名，全部用小写。正确写法例如：historydetail

2、SR 层只做参数的校验。如果存在不合法参数，应该抛出异常 IllegalArgumentException("OriginRankDetailSR 参数异常");

3、逻辑处理全部放到service层。

4、ArrayList 类型的对象命名后缀统一用List

5、业务相关的工具类应放到业务模块中，只有所有通用的工具类放到对应工程的common中。非必要，不要往puri-common中添加类或方法

6、定义枚举类型，必须要定义描述字段。当关键字比较长时，应使用数据code进行映射。正确写法如：

|  |
| --- |
| *//榜单名枚举* **public enum** RankEnum {   *//Zili Star 活动榜* ***RANKING\_OF\_ZILI\_STAR\_NEWFANS***(**"1"**, **"rankingOfZiliStarNewFans"**, **"Zili Star 活动榜榜单名"**),  *//原创总粉丝榜* ***RANKING\_OF\_USER\_TOTAL\_FANS***(**"2"**, **"rankingOfUserTotalFans"**, **"用户总粉丝量排行榜榜单名"**),  *//原创涨粉丝榜* ***RANKING\_OF\_USER\_NEW\_FANS***(**"3"**, **"rankingOfUserNewFans"**, **"用户新增粉丝排行榜榜单名"**),  *//原创内容榜* ***RANKING\_OF\_VIDEO\_NEW\_LIKE***(**"4"**, **"rankingOfVideoNewLike"**, **"视频新增点赞排行榜榜单名"**);   **private static** Map<String, RankEnum> *rankMap*;  **private** String **code**; *//榜单名code* **private** String **rankName**;*//榜单名* **private** String **desc**; *//描述* } |

7、枚举对象可以直接用==。一般对象的比较都是使用equals。在枚举对象内部，equals方法中比较直接使用的==来判断两个对象是否相等。

8、LOGGER.error 打印错误日志时，必须打出Exception e。样例如下：

|  |
| --- |
| ***LOGGER***.error(**"addRankHistory exception, "**, e); |

# 二、puri-mt项目规范

1、如非必要，新增的thrift 数据模型都定义在vo.thrift中

2、远端传输的thrift 数据模型命名均以Result结尾，形式为：

|  |
| --- |
| struct XXXResult {  1:optional i32 code; //状态码。 0：正常； 1：异常  2:optional string msg; //状态消息  3:optional T xxx; //返回的数据  } |

3、在service.thrift中定义接口方法使，第一个参数必须为1:string traceId

4、在PuriServiceImpl中重写接口方法时有以下几点规范:

（1）需要写方法注释。快捷方式： /\*\* 回车

（2）需要对传入的参数做为空校验

（3）需要 new 返回结果

（4）需要try-catch 业务逻辑

样例如下：

|  |
| --- |
| */\*\*  \** ***@param traceId*** *\** ***@param historyTime*** *历史日期，格式如"20210125  \** ***@return*** *在history\_rank表中添加该日期的历史数据  \** ***@throws*** *TException  \*/* @Override **public** BooleanResult addRankHistory(String traceId, String historyTime) **throws** TException {  BooleanResult booleanResult = **new** BooleanResult();  **try** {  booleanResult.setCode(0);  **if** (StringUtils.*isBlank*(historyTime)) {  booleanResult.setResult(**false**);  } **else** {  booleanResult.setResult(**originRankHistoryService**.addRankHistory(historyTime));  }  } **catch** (Exception e) {  booleanResult.setCode(1);  booleanResult.setMsg(e.getMessage());  ***LOGGER***.error(**"addRankHistory exception, "**, e);  }   **return** booleanResult; } |

5、在PuriServceImpl中禁止直接引用 Dao 或 Mapper。应当引用包装Dao 或 Mapper的Service 方法。

6、使用GsonUtils.parse方法解析对象时，如果这个对象是一个thrift对象，那么该对象的非string字段的值会在解析时因解析失败从而丢失。

# 三、数据库相关

1、mysql表的除了更新时间用mysql的时间（xml 中使用Mysql函数NOW()）外，其余的时间都用服务器时间(System.*currentTimeMillis*())。

# 四、其他

1.问题：在启动puri-api 与 puri-mt项目时，出现Log日志循环引用的问题。

解决：使用 mvn clean install -Dmaven.test/skip=true **-Pdev (使用-P指定环境)**

2.问题：在启动puri-mt项目时，出现数据库读库异常，具体报错信息如下：

文本

描述已自动生成

解决：查看本机物联网，应该使用MIOffice-5G网络。切换网络即可。

3.测试包命名规范：例如：2.18.0.1-wx-SNAPSHOT

4.问题：在Mysql数据库插入表情失败

图形用户界面, 文本, 应用程序

描述已自动生成

解决：<https://www.jb51.net/article/152223.htm>。

在dev环境修改表编码为utf8mb4后，可能依然无法插入表情。这是因为mysql数据库还需要做编码修改。直接在测试环境进行表情插入测试。

5.问题：puri-api项目在本地编译失败，具体报错信息如下：

文本

描述已自动生成

解决：切换到master分支仍然编译失败说明是本地仓库出现问题。直接删除对应的包目录（多半是/com/xiaomi/data），重新拉取新包即可。

如果拉取新包后，仍然存在某些类找不到的情况，可直接使用mvn clean install 安装一下即可。