# 接口公共参数：request\_ip(server\_ip)、main\_account(userCode)、sub\_account、mac、areaId(areaID)

# 公共返回参数：retCode、retDesc、data

共用方法判断传入参数是否为空

judgeParams(model,list){

判断逻辑

retCode=-1;

retDesc=”参数为空”

Retrun obj;

}

每个方法try catch

# 1：分类查询GetVodClass

ApkServiceImpl中的GetVodClass从缓存中取，初始化数据在initserviceimpl中进行（二级分类以children[]形式存放）



# 2：二级分类查询GetSecondLevelClass

去掉（Controller、service、dao、mybatis）

# 3：查包组GetVodFromFLClass、GetVodFromSLClass

参数保留：QueryFrom、QueryTo、Class\_id（二级是SecondClass\_id）

initserviceimpl中分别按热度初始化各一级分类为list，由小到大。Order by hot\_deg,id

接口请求时从缓存中拿，不查询数据库。要判断list的idx是否越界

查询二级分类时带上where SecondClass\_id=？

返回值：bg\_url、packages[{pkg\_id、pkg\_poster、pkg\_name}]

热搜中不带Class\_id，进行综合热度排序，取前几个。

前端每次查询5页数据。

# **4：查详情GetVodDetail**

初始化包组详情参数，放入MAP，以pkg\_id为key

传入参数：pkg\_id

Order\_turn排序方式（后端排序）

返回参数：

pkg\_total\_count总集数

pkg\_finish\_status完结状态0否 1是

pkg\_director导演

pkg\_id包组id

pkg\_introduce包组介绍

pkg\_leader主演

pkg\_poster海报地址

pkg\_name包组名称

pkg\_play\_count播放次数

pkg\_score评分

pkg\_type包组类型

pkg\_programs[program\_id,program\_idx,program\_play\_addr,program\_name, program\_code]分集

pkg\_bcharging收费标志0不收费 1收费

pkg\_play\_time\_limit预览时间限制

pkg\_free\_count免费观看集数（不鉴权）

pkg\_break\_program\_id最新断点分集

pkg\_break\_program\_time最新断点时间（秒）

pkg\_book\_mark是否收藏0否 1是

pkg\_relatives[pkg\_id、pkg\_name、pkg\_poster]推荐

pkg\_book\_mark需查询数据库

# **5：专题接口GetSubject**

# **6:：GetSoapOperaAssets、GetSoapOperaAssetsBreak**

去掉（Controller、service、dao、mybatis）

# **7：AddBookmark、DelBookmark**

参数pkg\_id

# **8：PlayFrom改成StartPlay**

前端传入：pkg\_id、program\_id，异步调用

# **9：GetFilterParams、FilterVods、SearchVods、SearchVodsDetail、DelWatchRecord、DelBookmarkAll、GetSoapOperaUpdateInfo、getInitalChar、addToBlackList、removeOutBlackList**

去掉（Controller、service、dao、mybatis）

# **10：SearchFuzzyVods搜索**

传入参数：SearchFrom、SearchTo、chars

返回参数：pkg\_id、pkg\_poster、pkg\_name、result\_count

# **11：GetWatchedVods、GetBookmarkVods**

传入参数：SearchFrom、SearchTo

返回参数：packages:[{pkg\_id、pkg\_name、pkg\_poster}]，result\_count

# **12：增加login接口**

返回request\_ip，前端注意覆盖的问题

# **13：reportErrorMessage**

前端try catch

传入参数：url(页面地址)、error\_desc(错误描述)

# **14：getSendnoti通知**

支持消息类型配置，前端支持点击跳转

# **15：uploadBreakPoint**

不计算播放百分比，只插入，后台定时清理之前的播放记录（存储过程），前后台只保留一个月的。（定时任务进行解析存储）

# **16：登录接口**

返回是否白名单用户

# **17：接口响应速度**

使用map进行存储，以方法名作为key，再开放接口查询整个map，get请求

# **18：鉴权、订购加密接口（独立）**

后台实现

# **19：订购信息插入接口（独立）**

后端提供给前端用于数据插入

# **20：启动加载**

通知消息：NOTIFY\_MAP

一级分类：CLASSIC\_VOD\_MAP

二级分类：CLASSIC\_SECOND\_VOD\_MAP

全部包组：PACKAGES\_MAP(全部包组，按热度排序)

一级分类包组：PACKAGES\_CLASSIC\_VOD\_MAP(多个一级分类下的包组，按热度排序)

二级分类包组：PACKAGES\_CLASSIC\_SECOND\_VOD\_MAP(多个一级分类下的包组，按热度排序)

首页配置：PAGEINFO\_MAP

包组详情：PACKAGE\_DETAIL\_MAP

专题加载：FEATURES\_MAP

都是静态变量static

# **21：定时任务**

pv、uv、vv、vv详情、vv端口数据合并、play\_count、专题pv、专题uv

更新缓存，5分钟一次

注意拆分成多个Scheduled

订购统计都进行管理台查询

# **22：页面生成**

定时任务生成

生成页面：按一级分类配置生成页面，包括专题作为一级分类

<html>

<body>

#replace

</body>

<script></script>

</html>

<div>电视剧</div>

<div>电影</div>

<div>综艺</div>

# **23：专题作为一级分类时的pv、uv统计接口**

**24：开放接口向外暴露接口**

后台管理系统通过调用几十个服务的接口进行展示

在线用户、并发数

查询并发执行：Runtime.*getRuntime*().exec("netstat -nat | grep -i ‘8082’|wc -l");

在线用户数采取心跳方式，单独建立服务，单独部署，对外暴露接口，存储在缓存中（main\_account为key自动覆盖），30分钟有效期，超过后清理。

前端30分钟请求一次