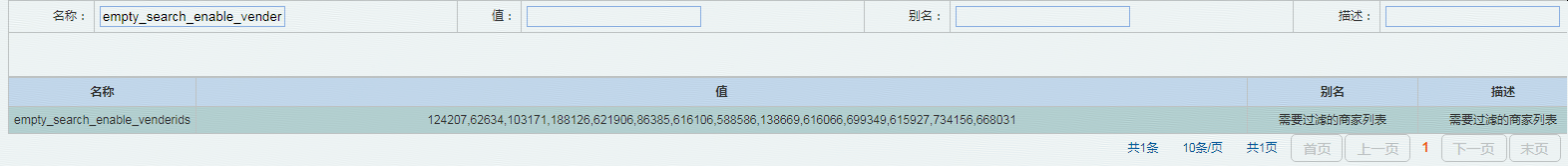
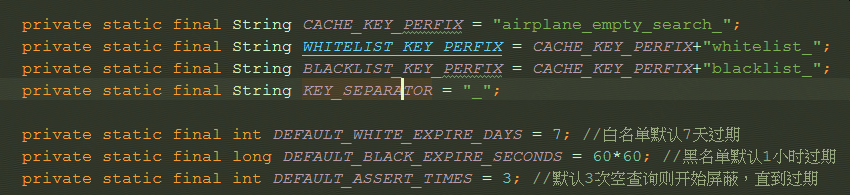
# 一期：迁页面



两个配置：

empty\_search\_enable\_venderids需要过滤空查询的商家列表





Key前缀用的是原来的

*/\*\*  
 \* 查询航线黑名单key前缀  
 \*/*private final String AIRPLANE\_EMPTY\_SEARCH\_BLACK = "airplane\_empty\_search\_blacklist\_";  
  
*/\*\*  
 \* 查询航线白名单key前缀  
 \*/*private final String AIRPLANE\_EMPTY\_SEARCH\_WHITE = "airplane\_empty\_search\_whitelist\_";

## 加入白名单的hash结构hSet

1. 页面中操作
2. Gw代码中自动拉白，当接口返回有航班数据时，加入白名单

key：

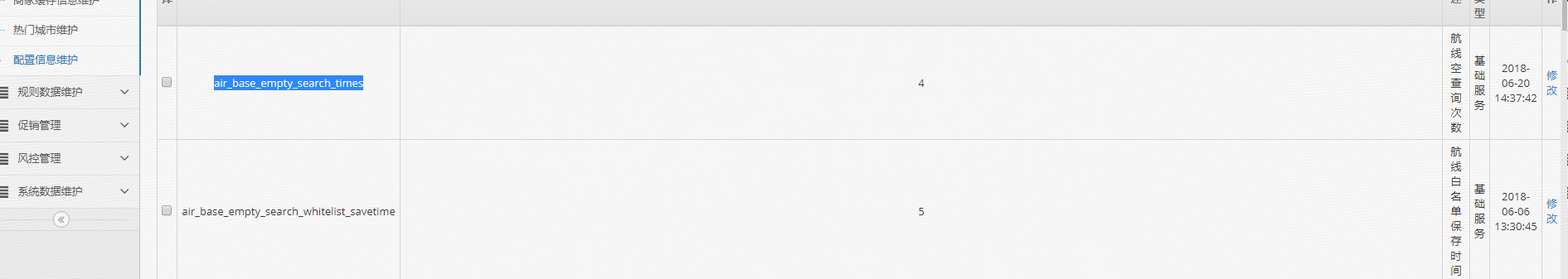
airplane\_empty\_search\_whitelist\_venderId

field:

depAirportCode\_arrAirportCode

value：

过期时间。默认7天，config中配置 改成airmis中air\_base\_empty\_search\_whitelist\_savetime（airmis）



## 加入黑名单的结构setEx（网关中）

Key:

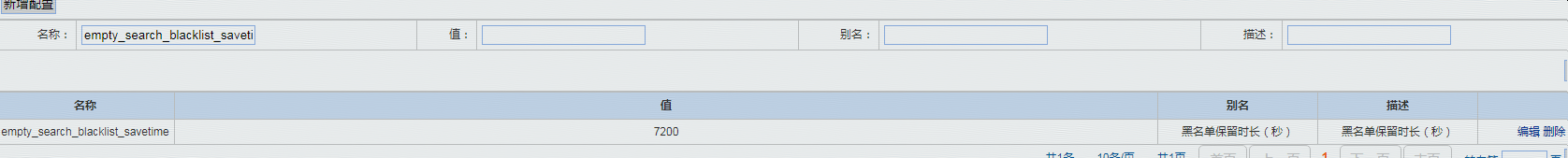
airplane\_empty\_search\_blacklist\_ venderId\_depAirportCode\_arrAirportCode

value:

次数，初始为1，每添加一次累加（incr，不会改变缓存有效时间）。默认3次空查询则开始屏蔽，直到过期，config中配置改成airmis：air\_base\_empty\_search\_times（airmis）

过期时间：

默认是1小时，在config中配置empty\_search\_blacklist\_savetime（保存时长，gw）



航班数据存储时间不超过10天，缓存的有效时间不超过5小时。

当查询工单没有数据时，可以先使用无效数据。

当航线在白名单中时，直接continue。不在白名单再判断是不是在黑名单中。所以白名单优先级高于黑名单。

当查询结果为空时，将空缓存

airplane\_empty\_search\_whitelist\_67742

## 黑名单批量查询

### 方案一：Scan命令

查询指定前缀的key，然后使用pipeline操作，特别慢。放弃

### 方案二：关联key

使用set结构：

|  |  |
| --- | --- |
| Key | Value |
| VenderId | VenderId\_depAirportCode\_arrAirportCode |

# 二期：迁代码

Common🡪baseServer

## 现状：

### Gw中调用common中

Tts：若返回航班数据不为空，加入白名单。否则缓存空数据，然后加入黑名单

加白名单的两种方法

1. 页面中操作
2. Gw代码中自动拉白，当接口返回有航班数据时，加入白名单

### Gw中调用

添加黑名单白名单

AbstractVenderProxy类

TTSHandlerUtil类

判断是否是黑名单：

DataCenterGroup类emptySearchFliter方法

判断是否过滤空查询：

DataCenterGroup类

AbstractVenderProxy类

TTSHandlerUtil类

## 改进

### Gw中调用base-server

通过RuleFacadeServiceI的emptySearchQuery方法

|  |  |
| --- | --- |
| 入参 | 参数说明 |
| EmptySearchEntity | venderId、depAirportCode、arrAirportCode  type：  为空则是查询黑白名单  为EmptySearchEnum.WHITE则是加入白名单（异步）  为EmptySearchEnum.BLACK则是增加空查询次数（异步） |

加入白名单时，直接加入，并删除该商家\_航线下的黑名单。加入前在调用处判断城市信息

加入黑名单时，若现在在白名单中则不加入（空查询次数一直累加直到过期，若移除了白名单则立刻回加入黑名单，），若不在白名单中则直接加入

加入黑白名单时，同时写关联key

### Airmis中

白名单不能拉黑

黑名单可以拉白（批量删除黑名单）

RuleFacadeServiceClient类

方法：

加入白名单insertEmptySearchWhiteList

|  |  |
| --- | --- |
| 参数 | 说明 |
| EmptySearchEntity | venderId、depAirportCode、arrAirportCode  必须不为空 |

查询黑白名单selectEmptySearchList

查询维度: 1.商家 2.航线(出发+到达) 3.商家+航线，分页

商家查询时分页，通过关联key查出符合score条件的pageSize条数据。

航线查询时不分页

|  |  |
| --- | --- |
| 参数 | 说明 |
| EmptySearchEntity | Type参数一定不为空，只能查黑名单和白名单的一种   1. venderId、depAirportCode、arrAirportCode三个参数都不为空，则是查询具体的商家、航线是否在黑白名单中。 2. 商家维度：venderId不为空，查询一个商家的名单 3. 航线维度：depAirportCode、arrAirportCode不为空 |

批量删除黑名单deleteEmptySearchList

不区分黑名单还是白名单

|  |  |
| --- | --- |
| 参数 | 说明 |
| String[] keys | venderId\_depAirportCod\_arrAirportCode三个参数拼接的字符串  直接删除，不关心结果 |
| String type | 操作类型，黑名单或白名单 |

Mis中配置air\_base\_empty\_search\_blacklist\_savetime

air\_base\_empty\_search\_times

air\_base\_empty\_search\_whitelist\_savetime

/\*\*

\* 默认构造函数，JSON反序列化用

\*/

public EmptySearchEntity() {

}

EmptySearchEntity emptySearchEntity = JSON.*parseObject*(entityJson, EmptySearchEntity.class);

### Ump监控：

1. 代码中打标

@FlightMonitor("flight.rule.pipeLine.emptySearchQuery")

1. Ump中新建key

