|  |  |
| --- | --- |
|  | modle |
| 页面-jipiao H5-jipiao | FlightQueryBean （airplane-model） |

需要将FlightQueryBean转换成新定义的model，然后调用flightServer的jsf接口

## Model定义

Model的IR—grab的OR，grab--工单

工单—grab，将grab的OS转换成model的IS（model中是对外的，jipiao）

备注：

1. Model参数尽量打平，source，userPin，来源
2. 查询中使用城市三字码，机场三字码，不适用城市中文名
3. 指定查询的model使用routeKey、flightKey、cacheKey
4. 指定查询中UUID，userPin，来源（sourceId）
5. 使用航线key，航班key，舱位key。航班返回参数中可以不用policyId、fareItemId、uniqueKey等参数。指定查询和退改签查询的时候通过入参的舱位key取缓存，缓存中取不到（1. 页面停留时间过长，退改签和指定查询都有可能，缓存过期。2. Redis挂掉）则通过航线key走dataHandler，通过航班key找到航班再找舱位，获得舱位数据后拼数据调用工单接口查询。可能会出现走dataHandler舱位不存在
6. 航线key：出发\_到达\_日期
7. 航班key：flightKey: "PEK#PVG#CA155#2018-11-21"
8. 舱位key：policyId、fareItemId、uniqueKey、uuId、productCode、customCode
9. 在13个舱位过滤逻辑之后，写缓存，缓存是散列的结构，key是MD5加密后的key（可以压缩成指定长度，16位或者32位），value是具体的舱位model。
10. 舱位key的加密在13个舱位循环的打标中做。
11. 现有的查询舱位detail，若缓存中为null，不会走dataHandler
12. IsSpecialClass，classText，classNoCn

可以这样：

cabinClassMap: {

PEK#PVG#CA#Y: {

subType: "经济舱",

classInfos: null,

majorType: "经济舱",

majorCode: "EC",

fullPrice: 1360

}

},

1. 接口

Map<FlightNo，List<Flight>>兼容了中转

Map<FlightNo，List<Cabin>>

Map<airPortCode，Airport>

Map<cityCode，City>

Map<airwayCode，Airways>

1. 指定查询，退改签，舱位展开共用一份缓存。

结构：String

Key: flightKey+UUID

Value:Json串，对象如下：

Object{

Flight

Map<cabinKey,Cabin>

}

1. Key里边是否要uuid：要。

## 注意：

Jimdb申请新集群，并行一段时间

订单中使用的促销信息是，指定查询时重新走了一遍ruleHandler

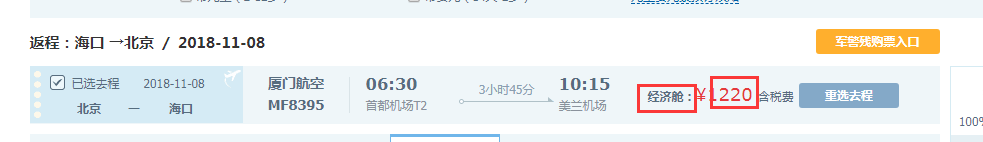
exsitsActivity， activityUserType，maxReducePrice，hasVenderCouponFlag

机场、城市、航站楼、日期、航班号

经停城市：

IsStop。为0则null，为1则有

|  |  |  |  |
| --- | --- | --- | --- |
| stops: [{ | |  |  |
|  | airportCode: "LYA", | | |
|  | cityCode: "LYA", | |  |
|  | arrvDate: "2018-11-18", | | |
|  | arrvTime: "09:55", | | |
|  | deptDate: "2018-11-18", | | |
|  | deptTime: "10:40", | | |
|  | duration: 45, | |  |
|  | through: null | |  |
| }], |  |  |  |

bingoLeastClassInfo的carrierAirwaysCn，price往返用

price，航班打标用，sourceId是经停用

过滤时间段，航司，机型，仓等，机场 弄成map结构

军警残和其他合成一个model

## 航班维度：

航班号，maxReducePrice和lowPrice，是否共享航班bollean，实际承运航班号

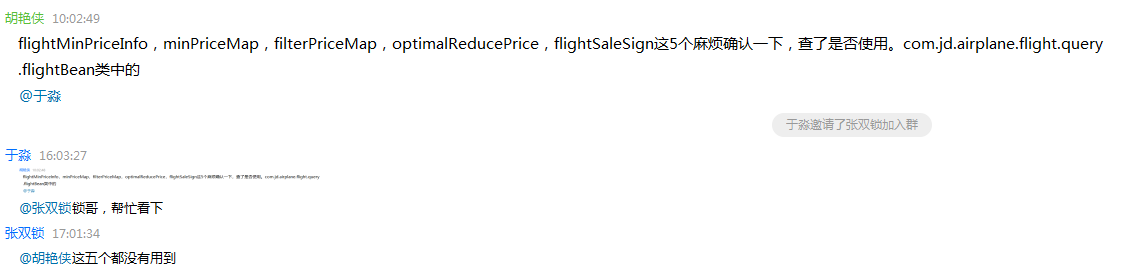
接口中不用有，调用者计算：DepTimeType，ArrTimeType，AddDay

是否经停（不用经停次数）

里程，机型（机型中文名，机型编码，查询不用PlaneStyleType），共享航班号

机型作为map，可以使航班上只有机型编码，不用机型中文名，只传一份

YseatPrice（y舱价格），餐食，



## 舱位维度

行李额， exsitsActivity，

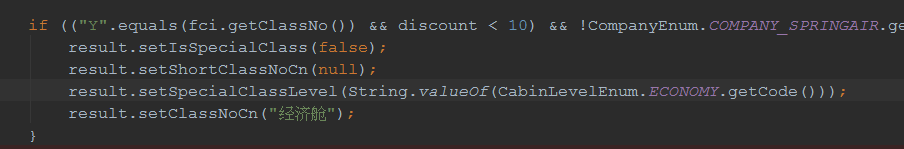
ActivityUserType，activityId





可以删除shortClassNoCn

除春秋外，经济舱都显示经济舱discount折



IsSpecialClass，classText（文案公用），classNoCn，可以这样：



cabinClassMap: {

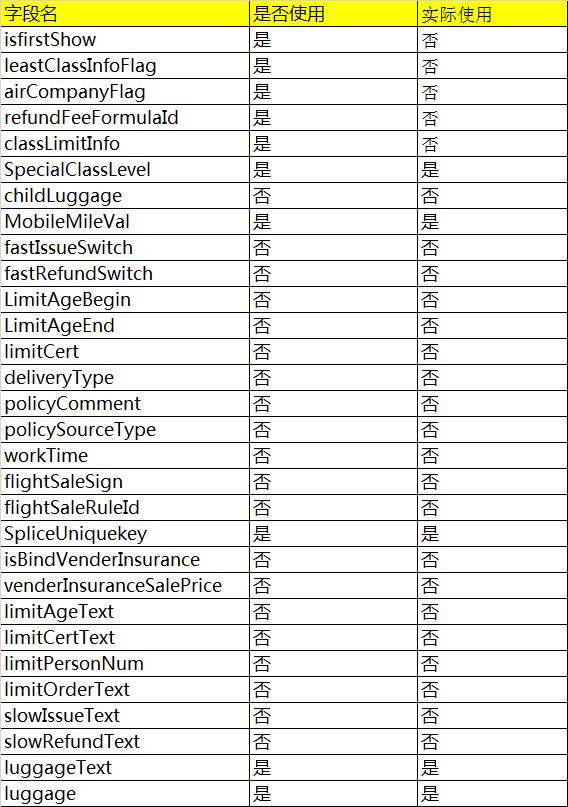
CA#G: {

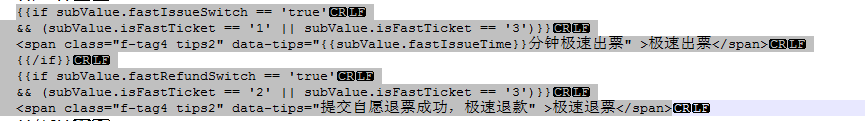
code: "G",

level: “Y”,

type: "超级经济舱",

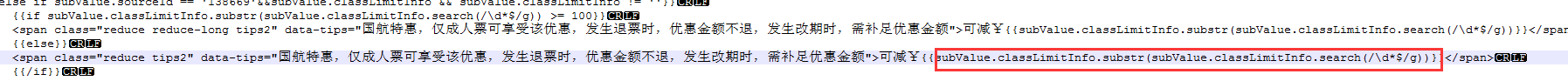
}





## 不用：

### 属性：

1. OptimalReducePrice（pc优惠码和普通立减取最大的），vm中没用，vm中pc优惠码用舱位的classLimitInfo
2. flightSaleSign（买赠）
3. flightBean中

ifSharedFlight不用（共享航班，查询页面实际用isCodeShare（int））

com.jd.airplane.domain.interfacedata.flightInfo中isCodeShare不用，订单用ifSharedFlight

共享航班是否也做异步接口（返回参数包变小）

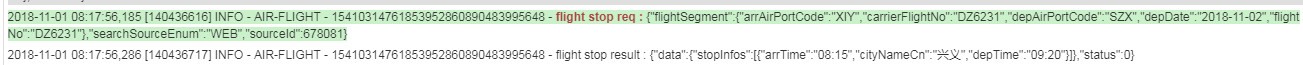
1. 航班的ShareFlightNo
2. FlightSaleSign

### 类：

## 经停：

除了九元和春秋走航司，其他走tts

com.jd.orderfuse.service.JdOrerFuseService



## 促销活动：

Sku促销

航班

exsitsActivity

ActivityUserType

MaxReducePrice

MaxPromotionBeans，

整合成一个map<活动针对的用户类型，活动的立减金额>

舱位问题

exsitsActivity

ActivityUserType

ReducePrice

ActivityId

PromotionBeans，

Pc码：PcCodesVal

买赠：sale

优惠券：coupon

ClassLimitInfo：gw的discountId--interface的classLimitInfo

### Map

航班

*/\*\*  
 \* 促销map，promotionMap：买赠“sale”,值0或1，优惠券“coupon”,值0或1，sku促销“skuPromotion”,值是skuPromotion类对象，pc码“pcCode”,值0或1  
 \*/*private Map<String, T> promap;

舱位

*/\*\*  
 \* 促销map，promotionMap：买赠“sale”,值saleId，优惠券“coupon”,值true，sku促销“skuPromotion”,值SKUPromotion对象，pc码“pcCode”,值pc码  
 \*/*private Map<String, T> promap;

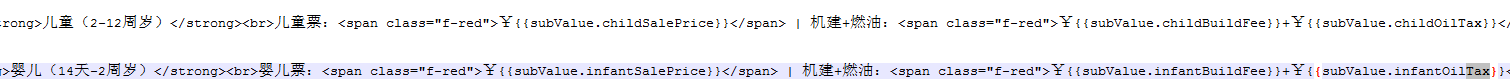
## 机建燃油：

一个航班下的基建费和燃油费应该一样。下面是返回错误的（旗舰店直接返回的）



航班打标：leastClassInfo

舱位打标：ClassInfo

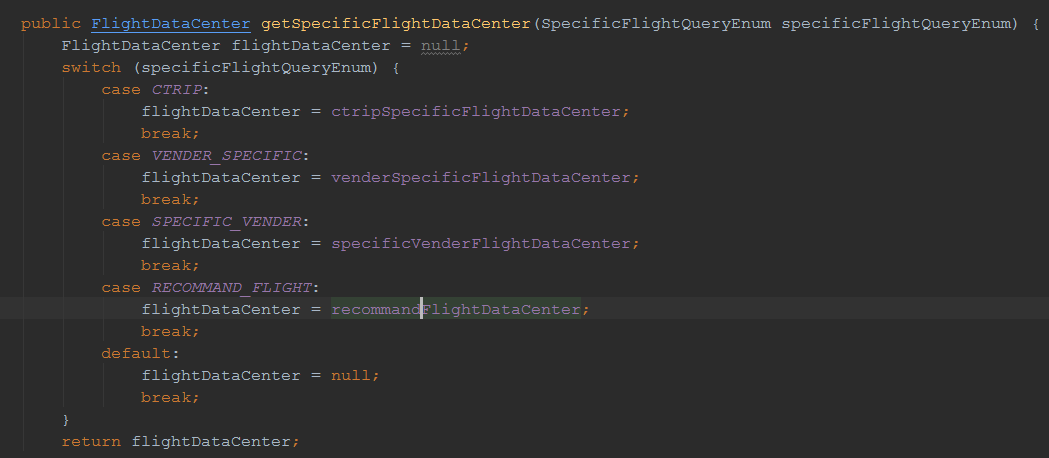


可删除的方法：

setSelfFlag：自营打标（将东鹰打标成京东自营）、

## 指定查询

VenderSpecific和SpecificVender合成一个接口，flightServer对外提供一个，内部决定走哪几步。前台预定调用的地方，适配一下，适配两套（携程，另一套）。



### 调用的地方：

#### 代码层面

FlightQueryServiceImpl中

getOrderFlightInfo

下游：

getFlightPriceInfo

getFlightPriceNewInfo（缓存flightSearchService.getOrderFlightInfo+指定查询接口）

0003接口

订单预定里边

#### 业务场景

往返查询的去程信息取缓存specificVender

预定取缓存走接口specificVender、venderSpecific

提交订单走验价接口venderSpecific

### 适配：

## 退改签：

现在前台调用退改签的地方参数不变，适配时其他参数赋值为空，拼key调用flightServer的新退改签接口，返回不变。

H5现在是直接调用flightServer的，可以让其先用旧的接口，当切成新的查询接口时候，退改签和指定查询接口也用新的

## 打标（包括携程）：

打标的给下游原始数据。

speTicketRemarks包机切位出票慢

private Integer limitTicketTime;对应打标文案slowRefundText（页面），打标slowIssueText; （页面）

private Integer orderMaxNum;对应浮层文案limitOrderText（页面），打标文案LimitPersonNum（页面）

private Integer orderMinNum; 对应浮层文案limitOrderText（页面），打标文案LimitPersonNum（页面）

limitAgeBegin

limitAgeEnd对应文案limitAgeText（页面）及classLimit中的ageTextLimit文案，ageLimit

limitCert对应文案LimitCertText（页面）及classLimit中的certTextLimit文案，certLimit

luggage打标，限。。kg（页面）

luggageText对应文案弹层（页面）

private RefundRemark refundRemark（页面）

private Double mile；？倍里程（页面）

isFastTicket

fastIssueTime

舱位：

private Map<String, String> classLimit；//携程对应Interface的ctripLimitInfoMap。Key是常数，value是对应的值。其他的对应

getAirFlightLimitMap:

年龄ageLimit、ageTextLimit，

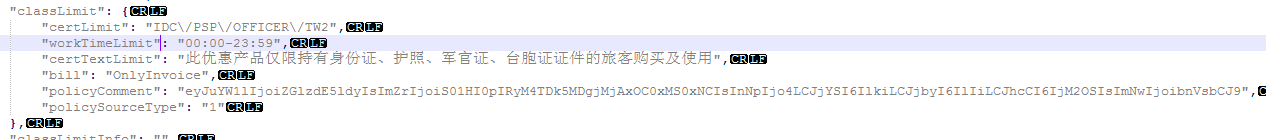
证件certLimit、certTextLimit，

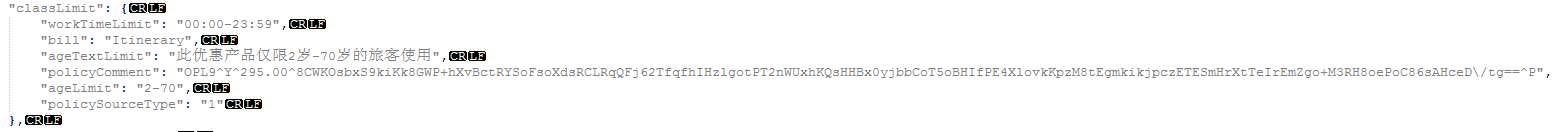
行程单bill，（页面）

政策备注policyComment，

政策来源policySourceType（1-TTS直连 2-TTS政策上传 3-自营），

出票时间workTimeLimit





ctripLimitInfoMap：

slow出票慢 （页面）

traveler成人>=12（页面）

bill 报销类型 （页面）

mp N人飞享（页面）

classLimitInfo：1.国航的pc码信息，gw的discountId--interface的classLimitInfo

2.携程的销售限制classInfo.setClassLimitInfo("您预订的是特价机票产品" + slow.toString() + travelerCategory.toString()

+ ageLimitInfo.toString() + billType.toString() + minPassenger.toString());

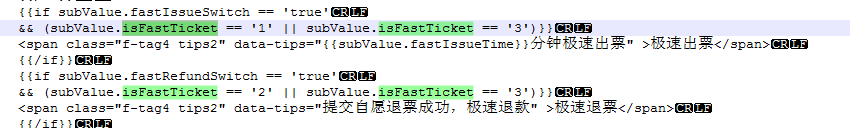
出票慢浮层文案、退票慢浮层文案、行李额文案initLuggageText

3. 行李额：成人行李额、文案（Luggage、LuggageText），儿童行李额（ChildLuggage

）；//这四个参数补充

4. 急速出退票，不用把开关字段传到前台，通过一个字段决定是0：非极速出退票1.急速出票2.急速退票3.急速出退票

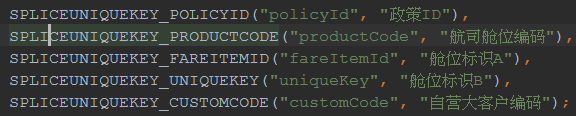
现状：



改成：

一个字段

## 待确认：

SpliceUniquekey 

## FlightQueryBean

|  |
| --- |
| private String arrCity;//删除，改用三字码 // 抵达城市 |
| private String arrTime; //不用，cabinkey// 抵达时间 |
| private String classNo;//删除 // 舱位代码 |
| private String depCity;//删除，改用三字码 // 出发城市 |
| private String depTime; // 不用，出发时间 |
| private String sourceId;//不用，往返的回程时，去程的值。cabinKey// 商家id |
| private String goTime;//不用// 出发时间 |
| private String backTime;//不用// 回程时间 |
| private String flightNo;//不用，国航，携程，普通和指定，展开舱位用，cabinkey// 2次查询航班号 |
| private String bookingClass; //不用，国航，普通和指定，cabinkey// 2次查询预订的仓位 |
| private boolean secondQueryFlag;//国航，普通和指定，cabinkey// 2次查询标识 |
| private boolean queryAdtFlag;//不用，国航默认true // 国航查询成人价格标识 |
| private int queryAdtNum;//不用，国航默认1 // 国航查询成人人数 |
| *private* String uniqueKey;//不用，cabinkey 选择去程的产品,此参数对往返的回程查询有效(携程) |
| private String backAirways;//不用，推荐航班 //返程航司 |
| private Map<String, Object> extendParams;//不用,用于存储额外需要传递的参数。 |
| //订单来源：mobile –手机m端、 app---移动端app、web---网站、travel---企业差旅、vip – 大客户 h5---H5页面 |
| private String sourceType;//二选一，已经用了source |
| private Boolean splitFlag;//不用，预定用, *拆分标识 true：是往返拆单* |
| private String limitKey;//删除，有cabinKey,舱位限制条件 |

private String arrCity;//删除，改用三字码 // 抵达城市

private String arrTime; //不用，cabinkey// 抵达时间

private String classNo;//删除 // 舱位代码

private String depCity;//删除，改用三字码 // 出发城市

private String depTime; // 不用，出发时间

private String sourceId;//不用，往返的回程时，去程的值。cabinKey// 商家id

private String goTime;//不用// 出发时间

private String backTime;//不用// 回程时间

private String flightNo;//不用，国航，携程，普通和指定，展开舱位用，cabinkey// 2次查询航班号

private String bookingClass; //不用，国航，普通和指定，cabinkey// 2次查询预订的仓位

private boolean secondQueryFlag;//国航，普通和指定，cabinkey// 2次查询标识

private boolean queryAdtFlag;//不用，国航默认true // 国航查询成人价格标识

private int queryAdtNum;//不用，国航默认1 // 国航查询成人人数

private String uniqueKey;//不用，cabinkey 选择去程的产品,此参数对往返的回程查询有效(携程)

private String backAirways;//不用，推荐航班 //返程航司

private Map<String, Object> extendParams;//不用,用于存储额外需要传递的参数。

private String sourceType;//二选一，已经用了source

private Boolean splitFlag;//不用，预定用, 拆分标识 true：是往返拆单

private String limitKey;//删除，有cabinKey,舱位限制条件

## FlightBean

\* 航空公司二字码

\*/

private String airways;

/\*\*

\* 航空公司中文名

\*/

private String airwaysCn;//从map中取

/\*\*

\* 到达机场中文名

\*/

private String arrAirdrome;//从map中取

/\*\*

\* 到达机场三字码

\*/

private String arrCity;

/\*\*

\* 到达日期

\*/

private String arrDate;

/\*\*

\* 到达航站楼

\*/

private String arrTerminal;

/\*\*

\* 到达时间

\*/

private String arrTime;

/\*\*

\* 舱位列表

\*/

private List<FlightClassBean> bingoClassInfoList;//不用，舱位放在map中

/\*\*

\* 起飞机场中文名

\*/

private String depAirdrome;//从map中取

/\*\*

\* 起飞机场三字码

\*/

private String depCity;

/\*\*

\* 起飞日期

\*/

private String depDate;

/\*\*

\* 起飞航站楼

\*/

private String depTerminal;

/\*\*

\* 起飞时间

\*/

private String depTime;

/\*\*

\* 起飞时间类型

\* 0-凌晨(0:00-6:00)、1-上午(6:00-12:00)、2-下午(12:00-18:00)、3-晚上(18:00-24:00)

\*/

private String depTimeType;//flightServer中不用，调用方计算，公共的，应该放到服务提供方

/\*\*

\* 到达时间类型 (同起飞一致)

\*/

private String arrTimeType;//flightServer中不用，调用方计算，应该放到服务提供方

/\*\*

\* 是否跨天到达 0 当天 1 跨天

\*/

private String addDay;//flightServer中不用，调用方计算，应该放到服务提供方

/\*\*

\* 航班号

\*/

private String flightNo;

/\*\*

\* 是否共享航班(1-是 0-否)

\*/

private int isCodeShare;//和ifSharedFlight重复，二选一

/\*\*

\* 是否经停

\*/

private String isStop;

/\*\*

\* 最低价舱位信息

\*/

private FlightClassBean bingoLeastClassInfo;//不用，舱位在cabinMap中

/\*\*

\* 餐食

\*/

private String meal;

/\*\*

\* 里程

\*/

private String milage;

/\*\*

\* 机型编号

\*/

private String planeStyle;

/\*\*

\* 机型中文名，是基础数据中的描述，大型机，小型机，中型机

\*/

private String planeStyleCN;//取map

/\*\*

\* 机型类型（1 大机型 2 中机型 3 小机型 4 其他机型）

\*/

private String planeStyleType;//取map

/\*\*

\* 共享航班号

\*/

private String shareFlightNo;//不用，二选一

/\*\*

\* 机建税 //机建费

\*/

private long tax;

/\*\*

\* Y舱价格

\*/

private long yseatPrice;

/\*\*

\* 是否俩小时内起飞 1-是 0-否

\*/

private int islimit2HoursFlight;//不用，可以前端做

/\*\*

\* 有无vip休息室，1：有，0：无

\*/

private int vipLoungeFlag;//不用

/\*\*

\* 飞行时间

\*/

private String airTime;

/\*\*

\* 门票信息 arrCityCode-达到城市三字码 scenicName-景点名 scenicDes-景点详情 detailURL-链接

\* picture-图片

\*/

private Map<String, String> ticketMap;//门票废弃，pc码可能用

/\*\*\*

\* 是否含有可领商家优惠券

\*/

private String hasVenderCouponFlag;

private Sale sale;//买赠打标

//是否命中促销

private boolean exsitsActivity;//待定

//活动针对的用户类型,0->所有用户,1->新用户

private String activityUserType;//待定

private long maxPromotionBeans; //最大京豆数量

private long maxReducePrice; //最大立减金额

private long optimalReducePrice;//pc、h5不用，差旅已经确认，所以删掉 //最优立减金额(6.2)

private int flightSaleSign;//pc、h5不用，差旅已经确认，所以删掉 //买赠打标(6.2)

/\*\*

\* 销售打标

\*/

//是否共享航班

private boolean ifSharedFlight;//二选一

//实际承运航班号

private String carrierFlightNo;

//实际承运航空公司中文名称

private String carrierAirwaysCn;//不用通过map取

/\*\*婴儿基建费\*/

private long infantBuildFee;

//航班最低价格相关信息

private FlightMinPriceInfo flightMinPriceInfo;//H5、pc不用，差旅已经确认 ，所以删除

//航班最低价信息,

" private Map<PriceComposeType,PriceItem> minPriceMap;//pc端不用，h5用，差旅已经确认，所以可以删除，将具体逻辑放在H5

pc端的details不进行MinPriceMap价格处理"

/\*\*\*

\* 舱等过滤报价信息,PC列表过滤用

\* key:

\* ECONOMY("Y","1", "Economy", "经济舱"),

\* BUSINESS("C", "3", "Business", "商务舱"),

\* FIRST("F", "2","First", "头等舱");

\*/

private Map<String,PriceItem> filterPriceMap;/低价列表页筛选仓等用

## FlightClassBean

|  |
| --- |
| public class FlightClassBean implements Serializable { |
| */\*\** |
| *\* 儿童燃油税* |
| *\*/* |
| private long childOilTax;//放航班上 |
| */\*\** |
| *\* 舱位代码* |
| *\*/* |
| private String classNo; |
| */\*\** |
| *\* 舱位中文描述* |
| *\*/* |
| private String classNoCn;//通过map读取，如大众游经济舱 |
| */\*\** |
| *\* 舱位等级（1 经济舱 2 头等 3 商务）* |
| *\*/* |
| private String classLevel; |
| */\*\** |
| *\* 儿童舱位代码* |
| *\*/* |
| private String childSeatCode; |
| */\*\** |
| *\* 舱位标识* |
| *\*/* |
| private String fareItemId;//用于退改签查询 |
| */\*\** |
| *\* 燃油税* |
| *\*/* |
| private long oilTax;//放航班上 |
| */\*\** |
| *\* 京东销售价* |
| *\*/* |
| private long price; |
| */\*\** |
| *\* 商家id* |
| *\*/* |
| private String sourceId; |
| */\*\** |
| *\* 剩余座位数A或9充足，1-8，显示座位数，0或其他值不显示该舱位* |
| *\*/* |
| private String seatNum; |
| */\*\** |
| *\* 儿童销售价* |
| *\*/* |
| private long childSalePrice; |
| */\*\** |
| *\* 优先展示标识（0是隐藏 1是显示）* |
| *\*/* |
| private String isfirstShow;//原来的第一屏展示，待确认差旅 |
| */\*\** |
| *\* 航司舱位标识* |
| *\*/* |
| private int airCompanyFlag;//可以删除，应该不用，待差旅确认 |
| */\*\** |
| *\* 京东特享标识-----l 去掉京东特享标示* |
| *\*/* |
| private String leastClassInfoFlag;//删除不用，这个其他商家展示成京东自营的打标 |
| */\*\** |
| *\* 航司舱位编码* |
| *\*/* |
| private String productCode;//待定，退改签用。航司会员产品 |
| */\*\** |
| *\* 政策id* |
| *\*/* |
| private String policyId;//待定，退改签用。航司的退改签、保险、行李额、预定销售控制 |
| */\*\** |
| *\* 退票费id* |
| *\*/* |
| private int refundFeeFormulaId;//没有赋值的地方，直接删除 |
| */\*\** |
| *\* 舱位标识* |
| *\*/* |
| private String uniqueKey;//待定，携程退改签缓存key，指定查询 |
| */\*\** |
| *\* 携程限制信息 slow 表示出票慢 只要有类就表示为出票慢产品 traveler "1"表示限成人购买，"0"表示没有限制 bill* |
| *\* 行程单类型限制 值包括Itinerary 行程单 OnlyInvoice仅发票 InvoiceOrItinerary发票(可提供行程单+差额发票)* |
| *\* mp 最小成行人数* |
| *\*/* |
| private Map<String, String> classLimit;//携程getCtripLimitMap，其他的限制getAirFlightLimitMap。页面中用：key不为null，则打标，打标的浮层是classLimitInfo，例如出票慢等 |
| */\*\** |
| *\* 携程限制文案* |
| *\*/* |
| private String classLimitInfo;//1.国航的pc码信息，gw的discountId--interface的classLimitInfo  2.携程的销售限制classInfo.setClassLimitInfo("您预订的是特价机票产品" + slow.toString() + travelerCategory.toString()  + ageLimitInfo.toString() + billType.toString() + minPassenger.toString()); |
|  |
| */\*\** |
| *\* 折扣* |
| *\*/* |
| private String discount;//销售价/Y舱全价：price/fullPrice |
| //是否命中促销 |
| private boolean exsitsActivity;//待定 |
| //活动id |
| private long activityId;//待定 |
| //立减金额 |
| private long reducePrice;//待定 |
|  |
| */\*\** |
| *\* 商家成本价* |
| *\*/* |
| private long venderPrice; |
|  |
| */\*\** |
| *\* 票面价* |
| *\*/* |
| private long originalPrice; |
|  |
| */\*\** |
| *\* 舱位类型，运价类型* |
| *\** |
| *\** ***@return*** |
| *\*/* |
| private String seatType; |
|  |
| */\*\** |
| *\* 国航PC优惠码 格式:HB108/HB109/50 无使用为空* |
| *\*/* |
| private String pcCodesVal;//待定 |
|  |
| */\*\** |
| *\* 成人行李 luggage =0 时鼠标悬浮显示提示文案* |
| *\*/* |
| private String luggage;//待定，设计 |
|  |
| */\*\** |
| *\* 成人行李文案* |
| *\*/* |
| private String luggageText;//待定，设计 |
|  |
| /\*儿童行李\*/ |
| private String childLuggage;//待定，设计，可能不需要 |
|  |
| /\*列表页-退改签信息标签\*/ |
| private TextRemark refundRemark;//退改。。。起 |
|  |
| /\*若有里程，则显示xx倍里程\*/ |
| private Double mile;//待定，设计 |
|  |
| */\*\** |
| *\* 是否特殊舱位* |
| *\*/* |
| private Boolean isSpecialClass;//待定，页面中用，除了经济舱，商务舱，头等舱，都是特殊舱位 |
|  |
| */\*\** |
| *\* 特殊舱位文案 特殊舱位鼠标悬浮文案展示提示信息* |
| *\*/* |
| private String classText;//使用map，1.20KG 免费托运行李额；\r\n2.一次免费信息变更服务(在符合更改条件的情况下，名字或者证件号码可以免费变更一次)；\r\n3.专享呼叫电话；\r\n4.前排座位；\r\n5.毛毯；\r\n6.餐食；\r\n7.报纸；\r\n8.饮品服务。 |
|  |
| */\*\** |
| *\* 特殊舱位等级* |
| *\*/* |
| private String specialClassLevel;//前台不用 |
|  |
| */\*\** |
| *\* 特殊舱位短名称* |
| *\*/* |
|  |
| private String shortClassNoCn;//可以删除，直接给全名称，舱位优化后没有赋值 |
|  |
| */\*\** |
| *\* 里程(移动端专用)* |
| *\*/* |
| private String mobileMileVal;//确定h5，差旅是否用 |
| */\*\*\** |
| *\* 是否含有可领商家优惠券* |
| *\*/* |
| private String hasVenderCouponFlag; |
| //活动针对的用户类型,0->所有用户,1->新用户 |
| private String activityUserType;//待定，设计 |
| //是否极速出退票 0：非极速出退票、1：仅极速出票、2：仅极速退票、3：极速出退票 |
| private String isFastTicket;//留着，在后端做逻辑后，直接给前端结果。只有TTS使用 |
| //极速出票开关 0-关、1-开 |
| private String fastIssueSwitch;//删除，开关没必要，代码中判断关闭则不赋值，isFastTicket |
| //极速退票开关 0-关、1-开 |
| private String fastRefundSwitch;//删除，开关没必要，代码中判断关闭则不赋值，isFastTicket |
| //出票极速时间 单位：分钟 |
| private String fastIssueTime; |
| //年龄段限制起 |
| private Integer limitAgeBegin;//待定，待确认，vm中有data- |
| //年龄段限制止 |
| private Integer limitAgeEnd;//待定，待确认，vm中有data- |
| //证件类型限制 |
| private String limitCert;//待定，待确认，vm中有data- |
| //配送类型：1-配送行程单 2-配送发票 3-配送行程单和发票 |
| private Integer deliveryType;//待定，待确认，vm中有data- |
| //政策备注（特殊说明） |
| private String policyComment;//在classLimit中有，不用单独设置，vm中没有 |
| //政策来源类型 1-TTS直连 2-TTS政策上传 |
| private Integer policySourceType;//待定，待确认，vm中有data- |
| //出票时间 |
| private String workTime;//待定，待确认，vm中有data- |
| //限制年龄文案 |
| private String limitAgeText;//在classLimit中有，不用单独设置，弹层 |
| //限制证件文案 |
| private String limitCertText;//在classLimit中有，不用单独设置，弹层 |
|  |
| */\*\** |
| *\* 销售打标* |
| *\*/* |
| private Sale sale;//待定，待设计 |
|  |
| private int flightSaleSign;//删除 //买赠打标(6.2) |
| private int flightSaleRuleId;//删除 //买赠打标活动ID(6.2) |
|  |
|  |
| //促销返京豆 |
| private long promotionBeans;//待定，待设计 |
|  |
| private String spliceUniquekey;//pc不用，差旅和H5待确认 //组装标识信息:Map<String,String>转成Json |
|  |
| //是否绑定商家保险产品销售,1:强制绑定保险销售 0：不强制商家保险销售 |
| private String isBindVenderInsurance = "0";//删除 |
| //商家保险售价 |
| private String venderInsuranceSalePrice = "";//删除 |
|  |
| */\*\** |
| *\* 儿童基建费* |
| *\*/* |
| private long childBuildFee;//不用，航班上 |
|  |
| */\*\** |
| *\* 婴儿基建费* |
| *\*/* |
| private long infantBuildFee;//不用，航班上 |
|  |
| */\*\** |
| *\* 婴儿燃油税* |
| *\*/* |
| private long infantOilTax;//不用，航班上 |
|  |
| */\*\** |
| *\* 婴儿销售价* |
| *\*/* |
| private long infantSalePrice; |
|  |
| */\*\** |
| *\* 婴儿成本价* |
| *\*/* |
| private long infantVenderPrice; |
|  |
| */\*\** |
| *\* 婴儿舱位代码* |
| *\*/* |
| private String infantSeatCode; |
|  |
| //产品名称 |
| private String productName = "";//和宾哥确认 |
| //产品描述 |
| private String productDesc = "";//和宾哥确认 |
|  |
| //大客户编码（商旅平台使用） |
| private String customCode="";//和宾哥确认，页面不用，可以作为舱位key的一部分 |
|  |
| //预定人数打标，包机切位 |
| private String limitPersonNum;//文案，限1-8人购票 |
|  |
| //预定人数限制浮层文案 |
| private String limitOrderText;//2-9人同时购买可享受此价格 |
|  |
| //出票慢文案 |
| private String slowIssueText;//特殊产品出票慢，航班起飞前2天/12小时出票，支付成功后保障出行 |
|  |
| //退票慢文案 |
| private String slowRefundText;//特殊产品，退票审核需3-5天，请耐心等待 |
| //展示优先级,1到13 |
| private String showPriority;//告诉前端坑位 |
|  |
| //最低价标识 0-不是 1-是 |
| private String lowestPriceFlag;//飞哥添加，可能多个舱位此标志是1 |
| //最小预定人数限制,等于null代表无限制 |
| private Integer orderMinNum; |
| //最大预订人数限制,等于null代表无限制 |
| private Integer orderMaxNum; |
| //政策类型23和24返回list列表，目前有1001，1002，1003，1004，1005，1006 |
| //1001 特殊产品：出票慢，最晚出票时间为航班起飞前2.5小时，保证100%出行 |
| private List<String> speTicketRemarks; |
| */\*\** |
| *\* 最晚出票时限（小时）* |
| *\*/* |
| private Integer limitTicketTime; |
| //舱位限制条件 |
| private String limitKey; |

## 评审问题:

共享航班

经停次数

vender优惠券

requestTime：时间唯一标识。

行李额与仓等

lableMap舱位标签Map

isSpecialClass

saleLimitMap

deliverType:h5打标怎么用

customCode

构建值key去掉

舱位的起飞时间去掉

缓存时间

client命名

调研订阅主从，或jimdb双写，sku促销老的

风控放在jipiao不动

VOP调用JDOS

航班列表，航班详情

pc、H5、差旅商家开关

版本号相关的代码可以去掉（老版本）

swagger

object序列化问题

## 适配逻辑

### 指定查询

Jipiao中订单调用的接口

下游调用的接口，返回结果是价格信息FlightPriceInfo+销售限制

### 退改签

用cabinKey、flightKey、rountKey替换现有的FlightClassBena中指定查询要用的字段，调用方无感知，直接调用，在服务方内部解析

## 响应码：

public class AirResponse<T> implements Serializable {  
 */\*\*  
 \* 返回状态码 0 代表成功，－1 代表失败，查看errorMessage获取具体错误消息  
 \*/* private Integer status;  
  
 */\*\*  
 \* 返回数据结果，只有当状态码为0-成功的情况下具有数据  
 \*/* private T data;  
  
 */\*\*  
 \* 详细错误消息，当状态码为－1失败情况下具有数据  
 \*/* private String errorMessage;

## 缓存策略

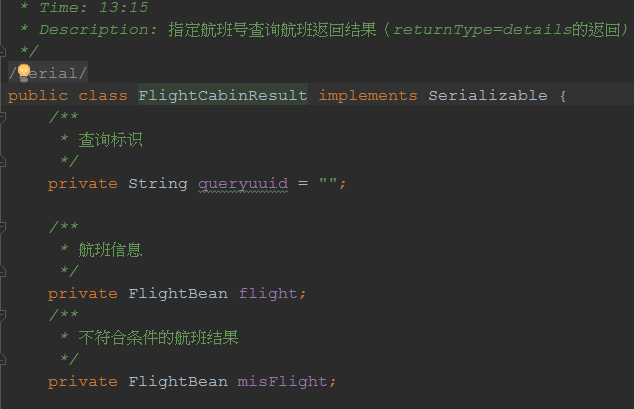
缓存1：航班查询公用的航班+舱位缓存：BASECACHE:depCity:arrCity:depDate:venderId

存储：

缓存2：航班下的舱位（13个舱位后）：QUERY\_FLIGHT\_CABINS\_uuid\_flightNo

存储：已完成，DetailsReturnHandler中15分钟





缓存时间延长：QUERY:uuid

LowPriceReturnHandler中



