|  |
| --- |
| **2013-12** |



**酒店应用接入说明文档V2.0**

版权申明：

本文件属于机密文件，所有权属于携程计算机技术（上海）有限公司。任何接触此件的人必须遵守以下规定：

1. 本文件及包含的内容均属于携程计算机技术（上海）有限公司资产。
2. 未经携程计算机技术（上海）有限公司的书面同意，任何人不得对本文件的全部和/或部分内容进行任何形式的复制或传播扩散。

**版本历史**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **修订章节** | **修改原因** | **更新日期** | **API版本号** |
| V1.0 | Daniel |  |  | 2013-01-22 |  |
| V2.0 | 崔继涛 |  | 增加接入模式说明，包括跳转下单、预付等 | 2013.12.27 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

[1. 准备工作 4](#_Toc376274917)

[1.1申请联盟Allianceid,Sid,Key 4](#_Toc376274918)

[1.2业务分析 4](#_Toc376274919)

[1.3API接入模式选择 4](#_Toc376274920)

[1.4 API分析 4](#_Toc376274921)

[1.5开发和测试 4](#_Toc376274922)

[1.6申请正式上线 5](#_Toc376274923)

[2.酒店接入模式 5](#_Toc376274924)

[2.1跳转至携程下单 5](#_Toc376274925)

[2.1.1应用场景描述 5](#_Toc376274926)

[2.1.2酒店产品的获取 7](#_Toc376274927)

[2.1.2.1酒店基本信息获取 7](#_Toc376274928)

[2.1.2.2酒店房态信息获取 9](#_Toc376274929)

[2.1.3酒店产品数据展示 10](#_Toc376274930)

[2.1.4到携程下单 11](#_Toc376274931)

[2.1.4.1跳转到携程酒店列表页 11](#_Toc376274932)

[2.1.4.2跳转到携程酒店详情页 11](#_Toc376274933)

[2.1.4.3跳转到携程酒店订单填写页 12](#_Toc376274934)

[2.2通过接口下单 12](#_Toc376274935)

[2.2.1现付接入流程 12](#_Toc376274936)

[2.2.2预付接入流程 26](#_Toc376274937)

[3.优化实践 38](#_Toc376274938)

[增量更新 38](#_Toc376274939)

[提升API调用性能 39](#_Toc376274940)

[详细的日志记录 39](#_Toc376274941)

[良好的错误处理 39](#_Toc376274942)

# 准备工作

欢迎大家使用携程的OPEN API接口，构建丰富多彩的应用，接口可以应用在pc端也可以用在无线端应用的开发。

## 1.1申请联盟Allianceid,Sid,Key

在做接入之前，您需要成为我们的开发者用户，登录<http://u.ctrip.com> 同时请加入携程联盟的QQ群181603509

成为我们联盟的一员，点击“管理我的帐户”

选择“联盟站点列表”，这里可以增加“未来你做产品展示的网站地址”，我们会根据你填写的信息给你分配SID，SiteKey(Allianceid 就是登录后在左侧“我的帐户”下面那个括号内的数据-AID).

点击”站点别名”会显示出来站点的基本信息,其中有APIKey的数据。

## 1.2业务分析

在进行API集成前，根据自己要做得产品的情况，先要做充分的业务分析：能否使用营销的工具（图片、文字、橱窗广告），能否使用头尾替换网站？能否使用PHP整站，如果以上不能够解决你的需求，考虑使用API，API集成到什么程度，是跳转到携程下单还是在自己的站点上通过接口完成下单，如果确定要使用API，请看API接入模式选择。

## 1.3 API接入模式选择

跳转到携程下单： 这种模式相对简单，相当于在联盟的网站上只是展示酒店的信息，用户在联盟网站上点击预定后，会跳转到携程的主站完成下单工作，主要流程是通过接口把酒店的基本信息和房态信息落地到本地数据库中，然后自己组织搜索和展示，根据携程给出的跳转规则拼接跳转的地址，然后跳转到携程，后续工作联盟不需要关心。

通过调用接口下单：这种模式适合需要完全埋名，做自己私有品牌的联盟，这比较复杂，需要开发的工作量相对比较大，所有的工作都是在联盟网站上完成。

## API分析

经过业务分析后，需要知道API能否满足业务需求，在每个业务场景中会使用到哪些API的方法，并考虑API方法的调用方式及数据的处理方式。

## 1.5开发和测试

酒店预订过程需要至少的页面：酒店列表(搜索)页、酒店详情页、订单填写页、订单成功页。可以学习携程网站的对应的页面上有哪些的元素及相关流程处理方法。根据api和给出的业务流程，自己开发或者利用携程提供的SDK进行程序设计，设计完成后，和我们的业务联系配合测试工作完成。

## 1.6申请正式上线

联盟开发完成并测试后，和我们技术人员联系，技术人员会根据先前的流程来审查您的产品是否满足上线的要求。

# 2.酒店接入模式

目前通过我们的API接口可以获取携程常规酒店的数据，数量大概是50000多家，其中价格计划包括了预付和现付两种，请根据具体的需求接入对应的产品！

## 2.1跳转至携程下单

这种模式是联盟展示酒店信息，客户点击后跳转的携程，由于跳转到携程下单，不会涉及到复杂的收款流程，而且我们的预付和一些优惠促销政策会有很多低价，所以建议联盟同时获取预付和现付的价格计划，这样可以提高您的用户的成交率，为您带来更多的订单。

### 2.1.1应用场景描述

下图是能帮助您完成酒店跳转到携程下单的API 使用场景。接口中请求体及返回体的字段描述详见“API2.4详细说明文档”。



在以上接口的调用和返回数据的处理中，请记得带上gzip。

以下是以**PHP**代码为例描述设置**gzip**和返回数据的处理：

|  |
| --- |
| 在请求接口时设置 |
| **$this**->setRequestHeader("Accept","\*/\*");  **$this**->setRequestHeader("Accept-Language","zh-cn");  **$this**->setRequestHeader("Accept-Encoding","gzip, deflate");//设置gzip模式请求  **$this**->setRequestHeader("User-Agent","HttpRequest Class 1.0"); |
| 返回数据后，做判断 |
| **public function** getResponseBody() (){  **if**(**$this**->getResponseHeader("Content-Encoding")=="gzip" && **$this**->getResponseHeader("Transfer-Encoding")=="chunked"){//判断返回数据是否是gzip压缩的  **return** gzdecode\_1(transfer\_encoding\_chunked\_decode(**$this**->responseBody));  }**else if**(**$this**->getResponseHeader("Content-Encoding")=="gzip"){  **return** gzdecode\_1(**$this**->responseBody);  }**else**{  **return $this**->responseBody;  }  } |

### 2.1.2酒店产品的获取

酒店的产品数据，分为静态基础酒店数据和动态房态数据。这些数据都是支持落地到本地数据库，然后联盟根据最终客户的查询条件，组织数据给最终客户。

#### 2.1.2.1酒店基本信息获取

联盟用户先根据接口的说明文档，编制本地数据库，然后编写自动定时同步的程序，例如JOB，定时来访问我们的接口。

**数据打底（第一次全量获取所有的酒店基本信息和指定天数内的房态数据）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 酒店的基本信息  OTA\_HotelSearch | 30--120次 | Y | 变化比较少，可以一周更新一次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  //构造权限头部  $headerRight=getRightString($AllianceID,$SID,$KEYS,$RequestType);  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelSearchRQ Version="0.0" PrimaryLangID="zh" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA\_HotelSearchRQ.xsd" xmlns="http://www.opentravel.org/OTA/2003/05">  **<ns:Criteria AvailableOnlyIndicator="false">// AvailableOnlyIndicator设置成false可以获取7天等**  <ns:Criterion>  <ns:HotelRef HotelCityCode="2" HotelName="上海"/>//hotelcityCode是城市的ID，从静态信息中可以获取，HotelName这个属性点，在做全量获取数据时，去掉  <ns:Award Provider="HotelStarRate" Rating="5"/>//如果全量获取数据，可以将此节点去掉，更多限制规则请看文档  </ns:Criterion>  </ns:Criteria>  </ns:OTA\_HotelSearchRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 酒店的房型信息  OTA\_HotelDescriptiveInfo | 30--120次 | Y | 变化比较少，可以一周更新一次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <OTA\_HotelDescriptiveInfoRQ Version="1.0" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA\_HotelDescriptiveInfoRQ.xsd" xmlns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <HotelDescriptiveInfos>  <HotelDescriptiveInfo HotelCode="625">//这里可以同时请求多条数据，最佳条数10条  <HotelInfo SendData="true"/>  <FacilityInfo SendGuestRooms="true"/>  <AreaInfo SendAttractions="true" SendRecreations="true"/>  <MultimediaObjects SendData="true"/>  </HotelDescriptiveInfo>  <HotelDescriptiveInfo HotelCode="11952">  <HotelInfo SendData="true"/>  <FacilityInfo SendGuestRooms="true"/>  <AreaInfo SendAttractions="true" SendRecreations="true"/>  <MultimediaObjects SendData="true"/>  </HotelDescriptiveInfo>  </HotelDescriptiveInfos>  </OTA\_HotelDescriptiveInfoRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 3 | 价格计划  OTA\_HotelRatePlan | 30--120次 | Y | 全量更新  注意频次  每次可以请求28天内的价格计划数据；接口可以提供90天内的价格计划（可通过3次请求获取） | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelRatePlanRQ TimeStamp="2012-05-01T00:00:00.000+08:00" Version="1.0">  <ns:RatePlans>  <ns:RatePlan>  <ns:DateRange Start="2013-01-06" End="2013-01-09"/>  <ns:RatePlanCandidates>  **<ns:RatePlanCandidate AvailRatesOnlyInd="true" >//如果是落地价格计划，请按照酒店来获取数据，一次可以支持10个酒店**  **<ns:HotelRefs>**  **<ns:HotelRef HotelCode="671"/>**  **</ns:HotelRefs>**  **</ns:RatePlanCandidate>**  </ns:RatePlanCandidates>  **<ns:TPA\_Extensions RestrictedDisplayIndicator="true"/>//为true时查询所有价格计划**  </ns:RatePlan>  </ns:RatePlans>  </ns:OTA\_HotelRatePlanRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 备注： | | | | | |

#### 2.1.2.2酒店房态信息获取

**数据更新（编制定时执行的JOB程序，通过OPEN API 接口获取房态数据）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 酒店价格缓存变化  OTA\_HotelCacheChange | 500--1000次 | Y | 可以每隔半个小时进行缓存变化获取 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelCacheChangeRQ Version="1.0">  <ns:CacheSearchCriteria CacheFromTimestamp="2013-01-17T00:00:00.000+08:00">  <ns:CacheSearchCriterion HotelCityCode="2" HotelCode="625"/>//可以同时请求多个酒店的数据，或者HotelCode去掉请求一个城市的变化。  <ns:CacheSearchCriterion HotelCityCode="2" HotelCode="671"/>//可以同时请求多个酒店的数据  </ns:CacheSearchCriteria>  </ns:OTA\_HotelCacheChangeRQ>  </RequestBody>  </HotelRequest>  </Request>  **对于价格缓存变化返回的信息，有些时间可能已经超出了你所落地的时间数据，所以这部分变化就不用考虑了，例如：现在是2014-1-1，你落地了30天的数据到2014-1-31号，那么你在请求价格缓存变化的时候，可能会返回提示你2014-4-1日这个酒店有变化，这个时间已经超出了你落地数据的时间，所以你应该忽略它，而是只关心你落地数据的时间内的，也就是当前时间到2014-1-31号的变化。** | | | | | |
| 2 | 价格计划  OTA\_HotelRatePlan | 30--120次 | Y | 根据1中返回的，发生价格变化的酒店信息，获取。  注意频次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| 如果通过价格计划接口返回的酒店价格数据，出现时间不连续，说明中间断掉的日期，价格几计划不存在了，需要在联盟的数据库中，删除这些数据。  **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelRatePlanRQ TimeStamp="2012-05-01T00:00:00.000+08:00" Version="1.0">  <ns:RatePlans>  <ns:RatePlan>  <ns:DateRange Start="2013-01-06" End="2013-01-06"/>  <ns:RatePlanCandidates>  **<ns:RatePlanCandidate AvailRatesOnlyInd="true" RatePlanCode="278550">//主要是根据价格缓存变化接口的返回数据，来更新落地价格计划，加上价格计划码。**  **<ns:HotelRefs>**  **<ns:HotelRef HotelCode="625"/>**  **</ns:HotelRefs>**  **</ns:RatePlanCandidate>**  </ns:RatePlanCandidates>  **<ns:TPA\_Extensions RestrictedDisplayIndicator="true"/>//为true时查询所有价格计划**  </ns:RatePlan>  </ns:RatePlans>  </ns:OTA\_HotelRatePlanRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

### 2.1.3酒店产品数据展示

联盟站长可以根据自己的业务需求，从落地的数据中，组织酒店的列表、子房型、详细等数据，展示给用户。具体的组织样式可以参看携程官网。

### 2.1.4到携程下单

具体的跳转规则可以和我们的业务索取，我们有跳转到PC上的规则，也有跳转到HTML5站点的规则，无论您是哪种，只要和我们的业务沟通即可，如果不满足，我们的业务会根据您的需求，评估是否有必要为您重新开发跳转连接。下面列出部分国内酒店PC上的跳转规则如下：

#### 2.1.4.1跳转到携程酒店列表页

按如下规则，拼接跳转的URL地址：

|  |
| --- |
| 酒店  a.请求URL  <http://u.ctrip.com/union/CtripRedirect.aspx?TypeID=10&CityName=%25E5%2593%2588%25E5%25B0%2594%25E6%25BB%25A8&CheckInDate=2012-05-30&CheckOutDate=2012-06-01&Starts=&Room=2&MinPirce=100&MaxPirce=500&HotelName=%25E5%25A6%2582%25E5%25AE%25B6%20&sid=xxx&allianceid=xxx>  b.参数说明  TypeID = 10 （10=酒店）  CityName 城市名称  CheckInDate 入住日期  CheckOutDate 退房日期  Starts 星级(可选)  Room 房间数(可选)  MinPirce 最低价(可选)  MaxPirce 最高价(可选)  HotelName 酒店名称(可选)  allianceid 联盟ID  sid 站点ID  ouid 联盟自定义参数  KeyWord 酒店位置  注意：参数中所有中文参数需要进行两次utf8编码后才可使用 |

#### 2.1.4.2跳转到携程酒店详情页

|  |
| --- |
| 酒店详情页  a.请求URL  <http://u.ctrip.com/union/CtripRedirect.aspx?TypeID=60&CheckInDate=2013-01-10&CheckOutDate=2013-01-11&sid=1000967&allianceid=3052&HotelID=126374>  b.参数说明  TypeID = 60 （60=酒店详情页）  HotelID 酒店ID  allianceid 联盟ID  sid 站点ID  ouid 联盟自定义参数  CheckInDate 入店时间  CheckOutDate 离店时间 |

#### 2.1.4.3跳转到携程酒店订单填写页

**请求URL**

<http://u.ctrip.com/union/CtripRedirect.aspx?TypeID=11&AllianceID=1&SID=1&Ouid=&SSOh=790dcdb4156db43f5b29c9afc2ea9347&SSOt=20130419161230&HotelID=436187&RoomID=2267629&Paymentterm=PP&StartDate=2013-09-19&depDate=2013-09-20>

**参数说明：**

|  |  |  |
| --- | --- | --- |
| **序号** | **参数名称** | **参数说明** |
| 1 | TypeID | TypeID = 11（固定） |
| 2 | SSOh | 登录平台参数（固定） |
| 3 | SSOt | 登录平台参数（固定） |
| 4 | HotelID | 酒店ID（必填） |
| 5 | RoomID | 房型ID（必填） |
| 6 | Paymentterm | 预付PP、现付FG（必填）价格计划中RatePlanCategory字段 |
| 7 | StartDate | 入住日期（必填） |
| 8 | depDate | 离店日期（必填） |
| 9 | allianceid | 联盟ID(可选) |
| 10 | sid | 站点ID(可选) |
| 11 | ouid | 联盟自定义参数(可选) |

返回的价格计划中RatePlanCategory字段RatePlanCategory=501和502为预付的价格计划，RatePlanCategory=14和16现付的价格计划

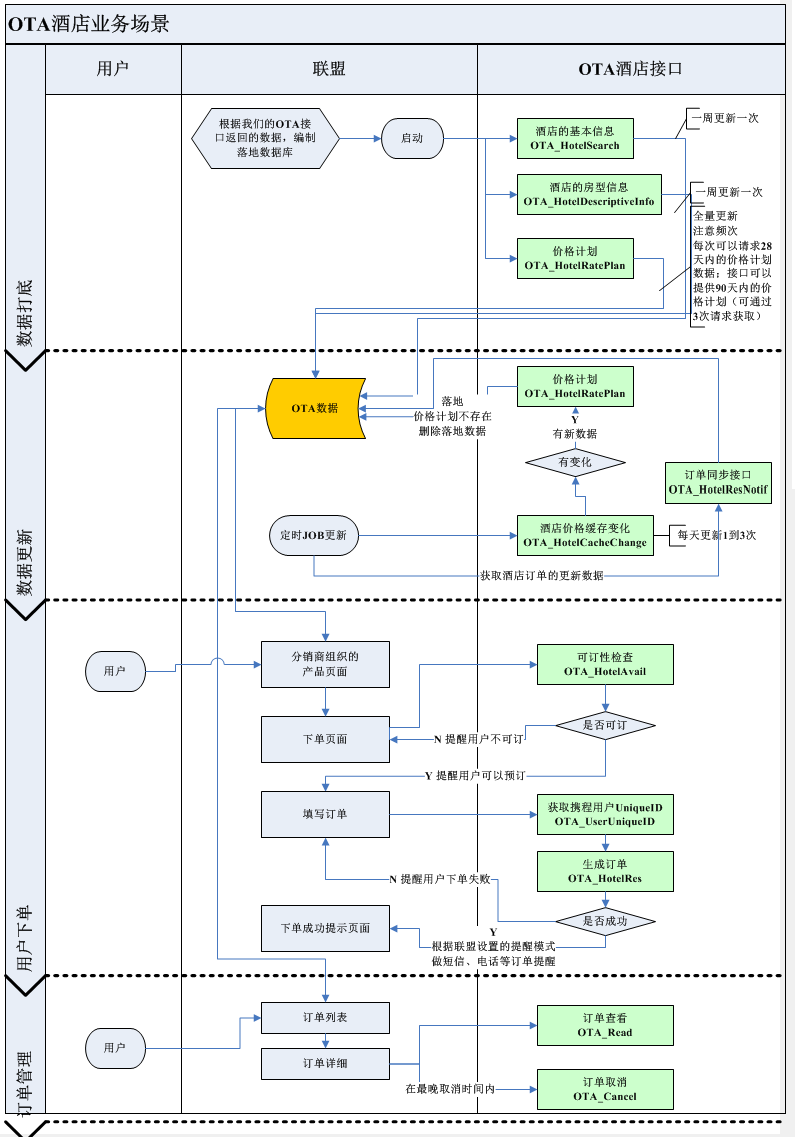
## 2.2通过接口下单

由于通过接口下单，涉及到的接口相对复杂，而且预付和现付的产品差别比较大，所以下面分别来进行阐述如何接入预付的产品和现付的产品。返回的价格计划中RatePlanCategory字段RatePlanCategory=501和502为预付的价格计划，RatePlanCategory=14和16现付的价格计划。

### 2.2.1现付接入流程

**1、现付应用场景描述**

下图是能帮助您完成酒店现付价格计划通过接口下单的接入API 使用场景。接口中请求体及返回体的字段描述详见“携程网站联盟OPENAPI2.4OTA(酒店)接口说明”。



在以上接口的调用和返回数据的处理中，请记得带上gzip。

以下是以**PHP**代码为例描述设置**gzip**和返回数据的处理：

|  |
| --- |
| 在请求接口时设置 |
| **$this**->setRequestHeader("Accept","\*/\*");  **$this**->setRequestHeader("Accept-Language","zh-cn");  **$this**->setRequestHeader("Accept-Encoding","gzip, deflate");//设置gzip模式请求  **$this**->setRequestHeader("User-Agent","HttpRequest Class 1.0"); |
| 返回数据后，做判断 |
| **public function** getResponseBody() (){  **if**(**$this**->getResponseHeader("Content-Encoding")=="gzip" && **$this**->getResponseHeader("Transfer-Encoding")=="chunked"){//判断返回数据是否是gzip压缩的  **return** gzdecode\_1(transfer\_encoding\_chunked\_decode(**$this**->responseBody));  }**else if**(**$this**->getResponseHeader("Content-Encoding")=="gzip"){  **return** gzdecode\_1(**$this**->responseBody);  }**else**{  **return $this**->responseBody;  }  } |

**2、酒店产品的获取**

酒店的产品数据，分为静态基础酒店数据和动态房态数据。这些数据都是支持落地到本地数据库，然后联盟根据最终客户的查询条件，组织数据给最终客户。

**2.1、酒店基本信息获取**

联盟用户先根据接口的说明文档，编制本地数据库，然后编写自动定时同步的程序，例如JOB，定时来访问我们的接口。

**数据打底（第一次全量获取所有的酒店基本信息和指定天数内的房态数据）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 酒店的基本信息  OTA\_HotelSearch | 30--120次 | Y | 变化比较少，可以一周更新一次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  //构造权限头部  $headerRight=getRightString($AllianceID,$SID,$KEYS,$RequestType);  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelSearchRQ Version="0.0" PrimaryLangID="zh" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA\_HotelSearchRQ.xsd" xmlns="http://www.opentravel.org/OTA/2003/05">  **<ns:Criteria AvailableOnlyIndicator="false">// AvailableOnlyIndicator设置成false**  <ns:Criterion>  <ns:HotelRef HotelCityCode="2" HotelName="上海"/>//hotelcityCode是城市的ID，从静态信息中可以获取，HotelName这个属性点，在做全量获取数据时，去掉  <ns:Award Provider="HotelStarRate" Rating="5"/>//如果全量获取数据，可以将此节点去掉，更多限制规则请看文档  </ns:Criterion>  </ns:Criteria>  </ns:OTA\_HotelSearchRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 酒店的房型信息  OTA\_HotelDescriptiveInfo | 30--120次 | Y | 变化比较少，可以一周更新一次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <OTA\_HotelDescriptiveInfoRQ Version="1.0" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA\_HotelDescriptiveInfoRQ.xsd" xmlns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <HotelDescriptiveInfos>  <HotelDescriptiveInfo HotelCode="625">//这里可以同时请求多条数据，最佳条数10条  <HotelInfo SendData="true"/>  <FacilityInfo SendGuestRooms="true"/>  <AreaInfo SendAttractions="true" SendRecreations="true"/>  <MultimediaObjects SendData="true"/>  </HotelDescriptiveInfo>  <HotelDescriptiveInfo HotelCode="11952">  <HotelInfo SendData="true"/>  <FacilityInfo SendGuestRooms="true"/>  <AreaInfo SendAttractions="true" SendRecreations="true"/>  <MultimediaObjects SendData="true"/>  </HotelDescriptiveInfo>  </HotelDescriptiveInfos>  </OTA\_HotelDescriptiveInfoRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 3 | 价格计划  OTA\_HotelRatePlan | 30--120次 | Y | 全量更新  注意频次  每次可以请求28天内的价格计划数据；接口可以提供90天内的价格计划（可通过3次请求获取） | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelRatePlanRQ TimeStamp="2012-05-01T00:00:00.000+08:00" Version="1.0">  <ns:RatePlans>  <ns:RatePlan>  <ns:DateRange Start="2013-01-06" End="2013-01-09"/>  <ns:RatePlanCandidates>  <ns:RatePlanCandidate AvailRatesOnlyInd="true" >**//如果是落地价格计划，请按照酒店来获取数据，一次可以支持10个酒店**  <ns:HotelRefs>  <ns:HotelRef HotelCode="671"/>  </ns:HotelRefs>  </ns:RatePlanCandidate>  </ns:RatePlanCandidates>  **<ns:TPA\_Extensions RestrictedDisplayIndicator="true/>//为true时查询所有的价格计划，如果你只做现付的价格计划，请在返回的价格计划中要找到RatePlanCategory字段过滤掉RatePlanCategory=501和502的预付的价格计划。**  </ns:RatePlan>  </ns:RatePlans>  </ns:OTA\_HotelRatePlanRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 备注： | | | | | |

**2.2、酒店房态信息获取**

**数据更新（编制定时执行的JOB程序，通过OPEN API 接口获取房态数据）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 酒店价格缓存变化  OTA\_HotelCacheChange | 500--1000次 | Y | 可以每隔半个小时进行缓存变化获取 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelCacheChangeRQ Version="1.0">  <ns:CacheSearchCriteria CacheFromTimestamp="2013-01-17T00:00:00.000+08:00">  <ns:CacheSearchCriterion HotelCityCode="2" HotelCode="625"/>//可以同时请求多个酒店的数据，或者HotelCode去掉请求一个城市的变化。  <ns:CacheSearchCriterion HotelCityCode="2" HotelCode="671"/>//可以同时请求多个酒店的数据  </ns:CacheSearchCriteria>  </ns:OTA\_HotelCacheChangeRQ>  </RequestBody>  </HotelRequest>  </Request>  **对于价格缓存变化返回的信息，有些时间可能已经超出了你所落地的时间数据，所以这部分变化就不用考虑了，例如：现在是2014-1-1，你落地了30天的数据到2014-1-31号，那么你在请求价格缓存变化的时候，可能会返回提示你2014-4-1日这个酒店有变化，这个时间已经超出了你落地数据的时间，所以你应该忽略它，而是只关心你落地数据的时间内的，也就是当前时间到2014-1-31号的变化。** | | | | | |
| 2 | 价格计划  OTA\_HotelRatePlan | 30--120次 | Y | 根据1中返回的，发生价格变化的酒店信息，获取。  注意频次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| 如果通过价格计划接口返回的酒店价格数据，出现时间不连续，说明中间断掉的日期，价格几计划不存在了，需要在联盟的数据库中，删除这些数据。  **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelRatePlanRQ TimeStamp="2012-05-01T00:00:00.000+08:00" Version="1.0">  <ns:RatePlans>  <ns:RatePlan>  <ns:DateRange Start="2013-01-06" End="2013-01-06"/>  <ns:RatePlanCandidates>  **<ns:RatePlanCandidate AvailRatesOnlyInd="true" RatePlanCode="278550">//主要是根据价格缓存变化接口的返回数据，来更新落地价格计划，加上价格计划码RatePlanCode。**  **<ns:HotelRefs>**  **<ns:HotelRef HotelCode="625"/>**  **</ns:HotelRefs>**  **</ns:RatePlanCandidate>**  </ns:RatePlanCandidates>  **<ns:TPA\_Extensions RestrictedDisplayIndicator="true/>//为true时会查询所有的价格计划，如果你只做现付的价格计划，请在返回的价格计划中要找到RatePlanCategory字段过滤掉RatePlanCategory=501和502的预付的价格计划。**  </ns:RatePlan>  </ns:RatePlans>  </ns:OTA\_HotelRatePlanRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

**2.3、本系统下单**

如果根据业务需要，联盟站长可以编制程序，在自己的系统上实现下单工作。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 可订性检查  OTA\_HotelAvail | 500--3000次 | Y | 直接调用 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  **<ns:OTA\_HotelAvailRQ Version="3.0" TimeStamp="2012-04-20T00:00:00.000+08:00">//注意使用版本为3.0**  <ns:AvailRequestSegments>  <ns:AvailRequestSegment>  <ns:HotelSearchCriteria>  <ns:Criterion>  <ns:HotelRef HotelCode="671"/>  **//注意支持凌晨下单，例如：现在是26号的晚上，无论是24点前预订，还是过了24点到了27号的凌晨预订，那么这个start都应该是26号而不是27号，因为凌晨左右定的房仍然是26号的（26号入住，27号离开）。**  **<ns:StayDateRange Start="2013-02-08T13:00:00.000+08:00" End="2013-02-09T15:00:00.000+08:00"/>**  <ns:RatePlanCandidates>  <ns:RatePlanCandidate RatePlanCode="13471"/>  </ns:RatePlanCandidates>  <ns:RoomStayCandidates>  <ns:RoomStayCandidate Quantity="2">  <ns:GuestCounts IsPerRoom="true">  <ns:GuestCount Count="4"/>  </ns:GuestCounts>  </ns:RoomStayCandidate>  </ns:RoomStayCandidates>  <ns:TPA\_Extensions>  **//最迟时间支持凌晨6点前下单**  **<ns:LateArrivalTime>2013-02-07T15:00:00.000+08:00</ns:LateArrivalTime>**  </ns:TPA\_Extensions>  </ns:Criterion>  </ns:HotelSearchCriteria>  </ns:AvailRequestSegment>  </ns:AvailRequestSegments>  </ns:OTA\_HotelAvailRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 生成新订单  OTA\_HotelRes | 500--3000次 | Y | 根据1中返回的数据，可以做如下下单请求：现付、信用卡担保、押金担保 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  **现付订单：**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  **<ns:OTA\_HotelResRQ TimeStamp="2012-07-28T13:06:30.000+08:00" Version="3.0" PrimaryLangID="zh" >//注意使用版本为3.0**  <ns:UniqueID Type="504" ID="100000"/>//固定标记  <ns:UniqueID Type="28" ID="1"/>//联盟的ID，Allianceid  <ns:UniqueID Type="503" ID="50"/>//联盟的站点ID,SID  <ns:UniqueID Type="1" ID="159c906a-aa28-4f54-b609-59d2c105fde2"/>//用户的携程UniqueID 获取方式见第3步  <ns:HotelReservations>  <ns:HotelReservation>  <ns:RoomStays>  <ns:RoomStay>  <ns:RoomTypes>  <ns:RoomType NumberOfUnits="1"/>  </ns:RoomTypes>  <ns:RatePlans>  <ns:RatePlan RatePlanCode="13471"/>  </ns:RatePlans>  <ns:BasicPropertyInfo HotelCode="671"/>  </ns:RoomStay>  </ns:RoomStays>  <ns:ResGuests>  **//最早到店时间支持凌晨6点前下单**  **<ns:ResGuest ArrivalTime="10:00:00+08:00">**  <ns:Profiles>  <ns:ProfileInfo>  <ns:Profile><ns:Customer>  <ns:PersonName>  <ns:Surname>test1</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>test2</ns:Surname>  </ns:PersonName>  <ns:ContactPerson ContactType="tel">  <ns:PersonName>  <ns:Surname>contact</ns:Surname>  </ns:PersonName>  <ns:Telephone PhoneNumber="34064880" PhoneTechType="1"/>  <ns:Email>XXXXX@ctrip.com</ns:Email></ns:ContactPerson>  </ns:Customer>  </ns:Profile>  </ns:ProfileInfo>  </ns:Profiles>  <ns:TPA\_Extensions>  **//最迟时间支持凌晨6点前下单**  **<ns:LateArrivalTime>2012-12-15T13:00:00+08:00</ns:LateArrivalTime>**  </ns:TPA\_Extensions>  </ns:ResGuest>  </ns:ResGuests>  <ns:ResGlobalInfo>  <ns:GuestCounts IsPerRoom="false">  <ns:GuestCount Count="2"/>  </ns:GuestCounts>  **//注意支持凌晨下单，例如：现在是26号的晚上，无论是24点前预订，还是过了24点到了27号的凌晨预订，那么这个start都应该是26号而不是27号，因为凌晨左右定的房仍然是26号的（26号入住，27号离开）。**  <ns:TimeSpan Start="2012-12-15T00:00:00+08:00" End="2012-12-16T00:00:00+08:00"/>  <ns:SpecialRequests>  <ns:SpecialRequest>  <ns:Text>test order ,please cancel it,thanks</ns:Text>  </ns:SpecialRequest>  </ns:SpecialRequests>  <ns:Total AmountBeforeTax="1800" CurrencyCode="CNY"/>  </ns:ResGlobalInfo>  </ns:HotelReservation>  </ns:HotelReservations>  </ns:OTA\_HotelResRQ>  </RequestBody>  </HotelRequest>  </Request>  **信用卡担保：**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelResRQ TimeStamp="2012-04-21T00:00:00.000+08:00" **Version="3.0"** PrimaryLangID="zh">  <ns:UniqueID ID="100000" Type="504"/>  <ns:UniqueID Type="28" ID="1"/>  <ns:UniqueID Type="503" ID="50"/>  <ns:UniqueID Type="1" ID="159c906a-aa28-4f54-b609-59d2c105fde2"/>  <ns:HotelReservations>  <ns:HotelReservation>  <ns:RoomStays>  <ns:RoomStay>  <ns:RoomTypes>  <ns:RoomType NumberOfUnits="8"/>  </ns:RoomTypes>  <ns:RatePlans>  <ns:RatePlan RatePlanCode="13471"/>  </ns:RatePlans>  <ns:BasicPropertyInfo HotelCode="671"/>  </ns:RoomStay>  </ns:RoomStays>  <ns:ResGuests>  **//最早到店时间支持凌晨6点前下单**  <ns:ResGuest ArrivalTime="13:00:00.000+08:00">  <ns:Profiles>  <ns:ProfileInfo>  <ns:Profile>  <ns:Customer>  <ns:PersonName>  <ns:Surname>王长春1</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春2</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春3</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春4</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春5</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春6</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春7</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王长春8</ns:Surname>  </ns:PersonName>  <ns:ContactPerson ContactType="TEL">  <ns:PersonName>  <ns:Surname>王长春</ns:Surname>  </ns:PersonName>  <ns:Telephone PhoneNumber="18599999999" PhoneTechType="1"/>  <ns:Email>Test@Ctrip.Com</ns:Email>  </ns:ContactPerson>  </ns:Customer>  </ns:Profile>  </ns:ProfileInfo>  </ns:Profiles>  <ns:TPA\_Extensions>  **//最迟时间支持凌晨6点前下单**  <ns:LateArrivalTime>2012-12-15T15:00:00.000+08:00</ns:LateArrivalTime>  </ns:TPA\_Extensions>  </ns:ResGuest>  </ns:ResGuests>  <ns:ResGlobalInfo>  <ns:GuestCounts IsPerRoom="false">  <ns:GuestCount Count="8"/>  </ns:GuestCounts>  **//注意支持凌晨下单，例如：现在是26号的晚上，无论是24点前预订，还是过了24点到了27号的凌晨预订，那么这个start都应该是26号而不是27号，因为凌晨左右定的房仍然是26号的（26号入住，27号离开）。**  <ns:TimeSpan Start="2012-12-15T13:00:00.000+08:00" End="2012-12-16T15:00:00.000+08:00"/>  <ns:SpecialRequests>  <ns:SpecialRequest>  <ns:Text>要一个电视</ns:Text>  </ns:SpecialRequest>  </ns:SpecialRequests>  <ns:DepositPayments>  <ns:GuaranteePayment GuaranteeType="CC/DC/Voucher">  <ns:AcceptedPayments>  <ns:AcceptedPayment>  <ns:PaymentCard CardType=" uTsv3Mbxikg=" CardNumber="XXXvnxg5aF+WeMRTjkL6JHkdDXK3b+J9" SeriesCode="cbyXXXJ06Bx8=" EffectiveDate="XX6by52NpPA=" ExpireDate="IX6by52NpPA=">  <ns:CardHolderName> z9OKRg9ysfP5BrpZ5FTJ4Q==</ns:CardHolderName>  <ns:CardHolderIDCard>bt4m5XXyKP8B64cqC4LQYgPRx9Lah+DJ</ns:CardHolderIDCard>  </ns:PaymentCard>  </ns:AcceptedPayment>  </ns:AcceptedPayments>  <ns:AmountPercent Amount="14400"/>  </ns:GuaranteePayment>  </ns:DepositPayments>  <ns:Total AmountBeforeTax="14400" CurrencyCode="CNY"/>  <ns:CancelPenalties>  <ns:CancelPenalty Start="2012-12-15T13:00:00.000+08:00" End="2012-12-16T15:00:00.000+08:00">  <ns:AmountPercent Amount="14400"/>  </ns:CancelPenalty>  </ns:CancelPenalties>  </ns:ResGlobalInfo>  </ns:HotelReservation>  </ns:HotelReservations>  </ns:OTA\_HotelResRQ>  </RequestBody>  </HotelRequest>  </Request>  **押金担保：**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelResRQ TimeStamp="2012-04-21T00:00:00.000+08:00" **Version="3.0">**  <ns:UniqueID ID="100000" Type="504"/>  <ns:UniqueID Type="28" ID="1"/>  <ns:UniqueID Type="503" ID="50"/>  <ns:UniqueID Type="1" ID="159c906a-aa28-4f54-b609-59d2c105fde2"/>  <ns:HotelReservations>  <ns:HotelReservation>  <ns:RoomStays>  <ns:RoomStay>  <ns:RoomTypes>  <ns:RoomType NumberOfUnits="2"/>  </ns:RoomTypes>  <ns:RatePlans>  <ns:RatePlan RatePlanCode="185940"/>  </ns:RatePlans>  <ns:BasicPropertyInfo HotelCode="16100"/>  </ns:RoomStay>  </ns:RoomStays>  <ns:ResGuests>  <ns:ResGuest ArrivalTime="13:00:00.000+08:00">  <ns:Profiles>  <ns:ProfileInfo>  <ns:Profile>  <ns:Customer>  <ns:PersonName>  <ns:Surname>王长春</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>王鑫</ns:Surname>  </ns:PersonName>  <ns:ContactPerson ContactType="SMS">  <ns:PersonName>  <ns:Surname>王长春</ns:Surname>  </ns:PersonName>  <ns:Telephone PhoneNumber="18599999999" PhoneTechType="5"/>  <ns:Email>Test@Ctrip.Com</ns:Email>  </ns:ContactPerson>  </ns:Customer>  </ns:Profile>  </ns:ProfileInfo>  </ns:Profiles>  <ns:TPA\_Extensions>  <ns:LateArrivalTime>2012-11-26T15:00:00.000+08:00</ns:LateArrivalTime>  </ns:TPA\_Extensions>  </ns:ResGuest>  </ns:ResGuests>  <ns:ResGlobalInfo>  <ns:GuestCounts IsPerRoom="false">  <ns:GuestCount Count="2"/>  </ns:GuestCounts>  <ns:TimeSpan Start="2012-11-22T13:00:00.000+08:00" End="2012-11-24T15:00:00.000+08:00"/>  <ns:SpecialRequests>  <ns:SpecialRequest>  <ns:Text>要一个电视</ns:Text>  </ns:SpecialRequest>  <ns:SpecialRequest>  <ns:Text>要一个电视2</ns:Text>  </ns:SpecialRequest>  </ns:SpecialRequests>  <ns:DepositPayments>  <ns:GuaranteePayment GuaranteeType="Deposit">  <ns:AcceptedPayments>  <ns:AcceptedPayment/>  </ns:AcceptedPayments>  <ns:AmountPercent Amount="876.00" CurrencyCode="CNY"/>  </ns:GuaranteePayment>  </ns:DepositPayments>  <ns:Total AmountBeforeTax="1752.00" CurrencyCode="CNY"/>  </ns:ResGlobalInfo>  </ns:HotelReservation>  </ns:HotelReservations>  </ns:OTA\_HotelResRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 3 | 获取携程UniqueID  OTA\_UserUniqueID | 500--5000次 | Y | 直接调用 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0" encoding="utf-8"?>  <Request>  <Header AllianceID="联盟ID" SID="联盟站点的ID" TimeStamp="时间错" RequestType="OTA\_UserUniqueID" Signature="XXXXXXXXXXXXXXXXXXXXXXX" />  <UserRequest>  <AllianceID>联盟ID </AllianceID>  <SID>联盟站点的ID</SID>  <UidKey>admin</UidKey>//用户在联盟站长系统中的用户标识，例如用户名  </UserRequest>  </Request> | | | | | |

**6、订单操作**

如果我们的客户是通过链接跳转到携程系统，做下单操作的，那么客户可以直接在携程系统上进行订单的管理，联盟站长无需操心订单的问题。

如果客户是在联盟站长的系统上进行下单的，那么联盟站长需要编制JOB程序，定时的获取到订单的变更信息，更新到联盟自己的系统中，提供联盟用户进行使用。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 订单变更  OTA\_HotelResNotif | 100--500次 | Y | 变化比较频繁，可以30分钟更新一次 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelResNotifRQ Version="1.0">  <ns:HotelReservations>  <ns:HotelReservation LastModifyDateTime="2013-01-04T00:00:00.000+08:00">  <ns:UniqueID Type="28" ID="1"/>  <ns:UniqueID Type="503" ID="50"/>  </ns:HotelReservation>  </ns:HotelReservations>  </ns:OTA\_HotelResNotifRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 订单详情查询  OTA\_Read | 100--500次 | Y | 可以直接调用 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <ns:OTA\_ReadRQ Version="1.0">  <ns:UniqueID Type="28" ID="XX"/>//联盟的ID，Allianceid  <ns:UniqueID Type="503" ID="XX"/>//联盟的SID  <ns:UniqueID Type="1" ID="XXXXXXX"/>//用户的携程UniqueID  <ns:UniqueID Type="501" ID="XXXXXX"/>//订单号  </ns:OTA\_ReadRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 3 | 订单取消接口  OTA\_Cancel | 100--500次 | Y | 可以直接调用 | 在订单允许的最晚取消时间内，可以通过接口取消订单。  数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <ns:OTA\_CancelRQ TimeStamp="2012-04-20T00:00:00.000+08:00" Version="1.0">  <ns:UniqueID ID="159c906a-aa28-4f54-b609-59d2c105fde2" Type="1"/>//用户的携程UniqueID  <ns:UniqueID ID="XX" Type="28"/>//联盟的ID，Allianceid  <ns:UniqueID ID="XXX" Type="503"/>//联盟的SID  <ns:UniqueID ID="XXXXX" Type="501"/>//订单号  <ns:Reasons>  <ns:Reason Type="506"/>//订单取消  </ns:Reasons>  </ns:OTA\_CancelRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

### 2.2.2预付接入流程

由于预付酒店是需要联盟自行收取酒店的费用，携程会根据联盟的押金和信用额度来处理预定工作，其中涉及到很多资金方面的业务，所以请联盟务必联系我们的业务人员，按照要求提供资料方便进行配置工作。

**1应用场景描述**

下图是能帮助您完成预付酒店预订全流程的API 使用场景。接口中请求体及返回体的字段描述详见酒店API的说明文档。





在以上接口的调用和返回数据的处理中，请记得带上gzip。

以下是以**PHP**代码为例描述设置**gzip**和返回数据的处理：

|  |
| --- |
| 在请求接口时设置 |
| **$this**->setRequestHeader("Accept","\*/\*");  **$this**->setRequestHeader("Accept-Language","zh-cn");  **$this**->setRequestHeader("Accept-Encoding","gzip, deflate");//设置gzip模式请求  **$this**->setRequestHeader("User-Agent","HttpRequest Class 1.0"); |
| 返回数据后，做判断 |
| **public function** getResponseBody() (){  **if**(**$this**->getResponseHeader("Content-Encoding")=="gzip" && **$this**->getResponseHeader("Transfer-Encoding")=="chunked"){//判断返回数据是否是gzip压缩的  **return** gzdecode\_1(transfer\_encoding\_chunked\_decode(**$this**->responseBody));  }**else if**(**$this**->getResponseHeader("Content-Encoding")=="gzip"){  **return** gzdecode\_1(**$this**->responseBody);  }**else**{  **return $this**->responseBody;  }  } |

**2.酒店数据的获取和更新**

酒店的产品数据，分为静态基础酒店数据和动态房态数据。这些数据都是支持落地到本地数据库，然后联盟根据最终客户的查询条件，组织数据给最终客户。

2.2.1酒店基本信息获取

联盟用户先根据接口的说明文档，编制本地数据库，然后编写自动定时同步的程序，例如JOB，定时来访问我们的接口。

**数据打底（第一次全量获取所有的酒店基本信息和指定天数内的房态数据）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 酒店的基本信息  OTA\_HotelSearch | 30--120次 | Y | 变化比较少，可以一周更新一次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  //构造权限头部  $headerRight=getRightString($AllianceID,$SID,$KEYS,$RequestType);  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelSearchRQ Version="0.0" PrimaryLangID="zh" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA\_HotelSearchRQ.xsd" xmlns="http://www.opentravel.org/OTA/2003/05">  **<ns:Criteria AvailableOnlyIndicator="false">// AvailableOnlyIndicator设置成false**  <ns:Criterion>  <ns:HotelRef HotelCityCode="2" HotelName="上海"/>//hotelcityCode是城市的ID，从静态信息中可以获取，HotelName这个属性点，在做全量获取数据时，去掉  <ns:Award Provider="HotelStarRate" Rating="5"/>//如果全量获取数据，可以将此节点去掉  </ns:Criterion>  </ns:Criteria>  </ns:OTA\_HotelSearchRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 酒店的房型信息  OTA\_HotelDescriptiveInfo | 30--120次 | Y | 变化比较少，可以一周更新一次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <OTA\_HotelDescriptiveInfoRQ Version="1.0" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA\_HotelDescriptiveInfoRQ.xsd" xmlns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <HotelDescriptiveInfos>  <HotelDescriptiveInfo HotelCode="625">//这里可以同时请求多条数据，最佳条数10条  <HotelInfo SendData="true"/>  <FacilityInfo SendGuestRooms="true"/>  <AreaInfo SendAttractions="true" SendRecreations="true"/>  <MultimediaObjects SendData="true"/>  </HotelDescriptiveInfo>  <HotelDescriptiveInfo HotelCode="11952">  <HotelInfo SendData="true"/>  <FacilityInfo SendGuestRooms="true"/>  <AreaInfo SendAttractions="true" SendRecreations="true"/>  <MultimediaObjects SendData="true"/>  </HotelDescriptiveInfo>  </HotelDescriptiveInfos>  </OTA\_HotelDescriptiveInfoRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 3 | 价格计划  OTA\_HotelRatePlan | 30--120次 | Y | 全量更新  注意频次  每次可以请求28天内的价格计划数据；接口可以提供90天内的价格计划（可通过3次请求获取） | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值，注意具体获取多少天的价格计划，请根据自己网站的需求而定，一般预定会发生在一个月内。 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelRatePlanRQ TimeStamp="2012-05-01T00:00:00.000+08:00" Version="1.0">  <ns:RatePlans>  <ns:RatePlan>  <ns:DateRange Start="2013-01-06" End="2013-01-09"/>  <ns:RatePlanCandidates>  **<ns:RatePlanCandidate AvailRatesOnlyInd="true" >//如果是落地价格计划，请按照酒店来获取数据，一次可以支持10个酒店**  **<ns:HotelRefs>**  **<ns:HotelRef HotelCode="671"/>**  **</ns:HotelRefs>**  **</ns:RatePlanCandidate>**  </ns:RatePlanCandidates>  **<ns:TPA\_Extensions RestrictedDisplayIndicator="true"/>//为true时会查询所有价格计划，请根据不同的价格计划类型做不同的流程处理。返回体中RatePlanCategory字段RatePlanCategory=501和502为预付的价格计划，RatePlanCategory=14和16现付的价格计划。**  </ns:RatePlan>  </ns:RatePlans>  </ns:OTA\_HotelRatePlanRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 备注： | | | | | |

2.2.2酒店房态信息更新

**数据更新（编制定时执行的JOB程序，通过OPEN API 接口获取房态数据）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 酒店价格缓存变化  OTA\_HotelCacheChange | 500--1000次 | Y | 可以每隔一个小时进行缓存变化获取 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式，访问频次可以提高到最大值 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelCacheChangeRQ Version="1.0">  <ns:CacheSearchCriteria CacheFromTimestamp="2013-01-17T00:00:00.000+08:00">  <ns:CacheSearchCriterion HotelCityCode="2" HotelCode="625"/>//可以同时请求多个酒店的数据，也可以把HotelCode去掉，请求一个城市的变化数据  <ns:CacheSearchCriterion HotelCityCode="2" HotelCode="671"/>//可以同时请求多个酒店的数据  </ns:CacheSearchCriteria>  </ns:OTA\_HotelCacheChangeRQ>  </RequestBody>  </HotelRequest>  </Request>  **对于价格缓存变化返回的信息，有些时间可能已经超出了你所落地的时间数据，所以这部分变化就不用考虑了，例如：现在是2014-1-1，你落地了30天的数据到2014-1-31号，那么你在请求价格缓存变化的时候，可能会返回提示你2014-4-1日这个酒店有变化，这个时间已经超出了你落地数据的时间，所以你应该忽略它，而是只关心你落地数据的时间内的，也就是当前时间到2014-1-31号的变化。** | | | | | |
| 2 | 价格计划  OTA\_HotelRatePlan | 30--120次 | Y | 根据1中返回的，发生价格变化的酒店信息，获取。  注意频次 | 数据的总量比较大，请一定要用gzip模式获取。同时加了gzip模式，访问频次可以提高到最大值 |
| 如果通过价格计划接口返回的酒店价格数据，出现时间不连续，说明中间断掉的日期，价格几计划不存在了，需要在联盟的数据库中，删除这些数据。  **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelRatePlanRQ TimeStamp="2012-05-01T00:00:00.000+08:00" Version="1.0">  <ns:RatePlans>  <ns:RatePlan>  <ns:DateRange Start="2013-01-06" End="2013-01-09"/>  <ns:RatePlanCandidates>  **<ns:RatePlanCandidate AvailRatesOnlyInd="true" RatePlanCode="278550">//主要是根据价格缓存变化接口的返回数据，来更新落地价格计划，加上价格计划码RatePlanCode。**  **<ns:HotelRefs>**  **<ns:HotelRef HotelCode="625"/>**  **</ns:HotelRefs>**  **</ns:RatePlanCandidate>**  </ns:RatePlanCandidates>  **<ns:TPA\_Extensions RestrictedDisplayIndicator="true"/>//为true时会查询所有价格计划，请根据不同的价格计划类型做不同的流程处理。返回体中RatePlanCategory字段RatePlanCategory=501和502为预付的价格计划，RatePlanCategory=14和16现付的价格计划。**  </ns:RatePlan>  </ns:RatePlans>  </ns:OTA\_HotelRatePlanRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

2.3酒店产品数据展示

联盟站长可以根据自己的业务需求，从落地的数据中，组织酒店的列表、子房型、详细等数据，展示给用户。具体的展示形式由自己而定。

2.4订单填写及联盟收款

当用户从联盟的网站上选定酒店和房间后，点击“预订”则会进入联盟自己组织的订单填写页面。用户在订单填写页面中要选择房间数目，甚至修改入住的日期。当所有的填写工作完成后，用户则会“下一步”，这个时候需要调用可订性检查（OTA\_HotelAvail）接口，可定性检查会返回预定的信息，如果返回提示不可以预定，则联盟需要提示客户重新选择酒店，如果可以预定，则联盟可以提示客户抓紧时间支付，因为房间数量是变化的，以免到联盟收款后，调用携程下单接口没有房的情况。联盟可以以各种方式收取房价金额，当联盟完成收款后，则会生成一个联盟本地的订单，然后会进入下一步。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 可订性检查  OTA\_HotelAvail | 500--3000次 | Y | 直接调用 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| **请求体样例：（相关函数请参考2.7常用函数）**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelAvailRQ **Version="2.0"** TimeStamp="2012-04-20T00:00:00.000+08:00">**//注意使用版本为2.0**  <ns:AvailRequestSegments>  <ns:AvailRequestSegment>  <ns:HotelSearchCriteria>  <ns:Criterion>  <ns:HotelRef HotelCode="671"/>  **//注意支持凌晨下单，例如：现在是26号的晚上，无论是24点前预订，还是过了24点到了27号的凌晨预订，那么这个start都应该是26号而不是27号，因为凌晨左右定的房仍然是26号的（26号入住，27号离开）。**  <ns:StayDateRange Start="2013-02-08T13:00:00.000+08:00" End="2013-02-09T15:00:00.000+08:00"/>  <ns:RatePlanCandidates>  **<ns:RatePlanCandidate RatePlanCode="13471" RatePlanCategory="501"/>//RatePlanCategory=501或者502都是预付**可订检查  </ns:RatePlanCandidates>  <ns:RoomStayCandidates>  <ns:RoomStayCandidate Quantity="2">  <ns:GuestCounts IsPerRoom="true">  <ns:GuestCount Count="4"/>  </ns:GuestCounts>  </ns:RoomStayCandidate>  </ns:RoomStayCandidates>  <ns:TPA\_Extensions>  <ns:LateArrivalTime>2013-02-07T15:00:00.000+08:00</ns:LateArrivalTime>  </ns:TPA\_Extensions>  </ns:Criterion>  </ns:HotelSearchCriteria>  </ns:AvailRequestSegment>  </ns:AvailRequestSegments>  </ns:OTA\_HotelAvailRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

2.5确定下单

联盟完成收款并生成本地订单以后，需要调用携程的下单接口完成预定，我们在接到联盟的下单请求后，内部会有一个防抵赖检查收款流程，需要联盟提供一个支付确认接口网址供携程这边调用，我们会传递一些收到的下单参数，当我们调用的时候，联盟需要根据我们传递的参数判断是否这个订单已经收款，并返回确认信息，我们会根据返回的确认信息决定能否生成新订单。注意，还需要开发分销商订单状态回传接口，这样可以确保订单状态推送，请联盟按照以下参数的格式开发，并把接受的网址提供给我们的业务注册。

**分销商支付确认接口：**

**提交格式：**

<http://www.yoursite.com/yourweb.aspx>

我们调用你提供的地址并附加上参数如下：http://www.yoursite.com/yourweb.aspx?order\_type=1&order\_id=test\_007

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 格式 |
| order\_type | 订单类型 | Int |
| order\_id | 分销商订单号 | String |

**接口需要返回的数据格式如下：**

{

    "is\_success": true,

    "pay\_amount": "15.00",

    "pay\_currency": "CNY",

    "remark": "备注写这里",

    "err\_code": 0,

    "err\_msg": "支付成功"

}

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 格式 |
| is\_success | 是否调用成功，成功：true；失败：false | Bool |
| pay\_amount | 支付金额（如果未支付，返回0） | Decimal |
| pay\_currency | 支付币种 | String |
| remark | 备注 | String |
| err\_code | 错误代码（调用失败时需要设置） | String |
| err\_msg | 错误描述（调用失败时需要设置） | String |

**分销商订单状态回传接口**

**提交格式：**

<http://www.yoursite.com/yourweb.aspx>

我们调用你提供的地址并附加上参数如下：http://www.yoursite.com/yourweb.aspx ?order\_type=1&alliance\_id=1&sid=50&distributor\_order\_id=10005&order\_status=1&remark=”下单成功”[&ctrip\_order\_id=212964584](http://202.103.25.41:6668/hotelorder/order/ctripConfrim?order_type=1&alliance_id=5308&sid=436599&distributor_order_id=tcp708009591696&order_status=0&remark=%e4%b8%8b%e5%8d%95%e6%88%90%e5%8a%9f&ctrip_order_id=212964584)

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 格式 |
| order\_type | 订单类型 | Int |
| alliance\_id | 联盟Id | String |
| sid | 分销商站点Id | Int |
| distributor\_order\_id | 分销商订单号 | String |
| order\_status | 订单状态（0-下单成功，1-下单失败，2-退单成功，3-退单失败） | Int |
| remark | 备注 | String |
| ctrip\_order\_id | 携程订单号 | Int |

**接口需要返回的数据格式如下：**

{

    "is\_success": true,

   "err\_code": 0,

    "err\_msg": "支付成功"

}

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 格式 |
| is\_success | 是否调用成功，成功：true；失败：false | Bool |
| err\_code | 错误代码，成功：0，失败：-1 | Int |
| err\_msg | 错误描述（调用失败时需要设置） | String |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 生成新订单  OTA\_HotelRes | 500--3000次 | Y | 如果支付确认成功则会允许下单 | **预付订单中RatePlanCategory只能为501：普通预付专用价格体系或者502：促销预付专用价格体系，对应价格计划返回体中的RatePlanCategory属性** |
| **请求体样例：（相关函数请参考2.7常用函数）**  **预付订单：**  <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelResRQ TimeStamp="2013-08-28T13:06:30.000+08:00" **Version="2.0"** PrimaryLangID="cn" **RatePlanCategory="501"**>//**预付订单中RatePlanCategory只能为501：普通预付专用价格体系或者502：促销预付专用价格体系，对应价格计划返回体中的RatePlanCategory属性，注意使用版本为2.0**    <ns:UniqueID Type="504" ID="test\_030"/>    <ns:UniqueID Type="28" ID="130"/>    <ns:UniqueID Type="503" ID="260"/>    <ns:UniqueID Type="1" ID="159c906a-aa28-4f54-b609-59d2c105fde2"  />    <ns:HotelReservations>      <ns:HotelReservation>        <ns:RoomStays>          <ns:RoomStay>            <ns:RoomTypes>              <ns:RoomType NumberOfUnits="1"/>            </ns:RoomTypes>            <ns:RatePlans>              <ns:RatePlan RatePlanCode="7635"/>            </ns:RatePlans>            <ns:BasicPropertyInfo HotelCode="40"/>          </ns:RoomStay>        </ns:RoomStays>        <ns:ResGuests>  **//最早到店时间一般是预订日期的24时之前**  **<ns:ResGuest ArrivalTime="10:00:00+08:00">**  <ns:Profiles>  <ns:ProfileInfo>  <ns:Profile><ns:Customer>  <ns:PersonName>  <ns:Surname>test1</ns:Surname>  </ns:PersonName>  <ns:PersonName>  <ns:Surname>test2</ns:Surname>  </ns:PersonName>  <ns:ContactPerson ContactType="tel">  <ns:PersonName>  <ns:Surname>contact</ns:Surname>  </ns:PersonName>  <ns:Telephone PhoneNumber="34064880" PhoneTechType="1"/>  <ns:Email>XXXXX@ctrip.com</ns:Email></ns:ContactPerson>  </ns:Customer>  </ns:Profile>  </ns:ProfileInfo>  </ns:Profiles>  <ns:TPA\_Extensions>  **//最迟时间支持凌晨6点前下单**  **<ns:LateArrivalTime>2012-12-15T13:00:00+08:00</ns:LateArrivalTime>**  </ns:TPA\_Extensions>  </ns:ResGuest>  </ns:ResGuests>  <ns:ResGlobalInfo>  <ns:GuestCounts IsPerRoom="false">  <ns:GuestCount Count="2"/>  </ns:GuestCounts>  **//注意支持凌晨下单，例如：现在是26号的晚上，无论是24点前预订，还是过了24点到了27号的凌晨预订，那么这个start都应该是26号而不是27号，因为凌晨左右定的房仍然是26号的（26号入住，27号离开）。**  <ns:TimeSpan Start="2012-12-15T00:00:00+08:00" End="2012-12-16T00:00:00+08:00"/>          <ns:SpecialRequests>            <ns:SpecialRequest>              <ns:Text>测试订单，请取消，谢谢</ns:Text>            </ns:SpecialRequest>          </ns:SpecialRequests>          <ns:Total AmountBeforeTax="400" CurrencyCode="CNY"/>        </ns:ResGlobalInfo>      </ns:HotelReservation>    </ns:HotelReservations>  </ns:OTA\_HotelResRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 获取携程UniqueID  OTA\_UserUniqueID | 500--5000次 | Y | 直接调用 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0" encoding="utf-8"?>  <Request>  <Header AllianceID="联盟ID" SID="联盟站点的ID" TimeStamp="时间错" RequestType="OTA\_UserUniqueID" Signature="XXXXXXXXXXXXXXXXXXXXXXX" />  <UserRequest>  <AllianceID>联盟ID </AllianceID>  <SID>联盟站点的ID</SID>  <UidKey>admin</UidKey>//用户在联盟站长系统中的用户标识，例如用户名  </UserRequest>  </Request> | | | | | |

2.6订单操作

2.6.1订单号获取

由于订单号开始是由接口推送的，这种推送机制可能因为暂时的网络、联盟接受服务器宕机、携程推送服务器宕机等原因导致推送不成功，对于没有携程订单号的订单，联盟可以定时的调用此接口获取携程的订单号，说明如下：

http://**{API\_Url}**/ hotel/D\_GetCtripOrderID.asmx

<?xml version="1.0"?>

<Request>

<Header $headerRight/>

<DomesticGetCtripOrderIDRequest>

<!--分销商订单号列表-->

<DistributorOrderIDList>

<DistributorOrderIDDetail>

<!--分销商订单号-->

<DistributorOrderID>test122</DistributorOrderID>

</DistributorOrderIDDetail>

<DistributorOrderIDDetail>

<DistributorOrderID>100ddddd000</DistributorOrderID>

</DistributorOrderIDDetail>

</DistributorOrderIDList>

</DomesticGetCtripOrderIDRequest>

</Request>

响应如下：

<?xml version="1.0"?>

<Response>

<Header ShouldRecordPerformanceTime="False" Timestamp="2014-03-20 10:35:30:67647" ReferenceID="2f5fa701-d685-4849-a80a-20221eac8c2e" ResultCode="Success" />

<DomesticGetCtripOrderIDResponse>

<CtripOrderIDList>

<CtripOrderIDDetail>

<!--分销商订单号-->

<DistributorOrderID>test122</DistributorOrderID>

<!--携程订单号-->

<CtriporderID>100682946</CtriporderID>

</CtripOrderIDDetail>

<CtripOrderIDDetail>

<DistributorOrderID>100ddddd000</DistributorOrderID>

<CtriporderID>100682947</CtriporderID>

</CtripOrderIDDetail>

</CtripOrderIDList>

</DomesticGetCtripOrderIDResponse>

2.6.2定时同步订单状态

联盟需要编写job定时的通过订单变更（OTA\_HotelResNotif）接口同步订单的状态，有订单变更返回时，需要逐条的调用订单详情查询（OTA\_Read）接口来更新本地数据库中的订单，如果有取消成功的订单则应该进入联盟的退款流程。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 订单变更  OTA\_HotelResNotif | 100--500次 | Y | 变化比较频繁，可以30分钟更新一次 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <RequestBody xmlns:ns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <ns:OTA\_HotelResNotifRQ Version="1.0">  <ns:HotelReservations>  <ns:HotelReservation LastModifyDateTime="2013-01-04T00:00:00.000+08:00">  <ns:UniqueID Type="28" ID="1"/>  <ns:UniqueID Type="503" ID="50"/>  </ns:HotelReservation>  </ns:HotelReservations>  </ns:OTA\_HotelResNotifRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |
| 2 | 订单详情查询  OTA\_Read | 100--500次 | Y | 可以直接调用 | 数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <ns:OTA\_ReadRQ Version="2.0">//请使用2.0版本  <ns:UniqueID Type="28" ID="XX"/>//联盟的ID，Allianceid  <ns:UniqueID Type="503" ID="XX"/>//联盟的SID  <ns:UniqueID Type="1" ID="XXXXXXX"/>//用户的携程UniqueID  <ns:UniqueID Type="501" ID="XXXXXX"/>//订单号  </ns:OTA\_ReadRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

2.6.3查看订单详情

客户在联盟站点查看订单详情的时候，联盟即可以展示自己本地系统的订单，也可以通过携程提供的订单详情查询（OTA\_Read）接口来获取详情。

2.6.4订单取消

如果客户是在联盟的站点申请取消订单，则联盟需要调用携程提供的订单取消接口（OTA\_Cancel），接口会返回取消的信息，如果不能取消，联盟需要立即反馈给用户，如果可能允许取消，联盟需要提示用户取消审核中，但是能否取消成功，以携程以后返回的订单状态为准。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **步骤** | **接口** | **接口频次** | **gzip** | **更新规则** | **注意事项** |
| 1 | 订单取消接口  OTA\_Cancel | 100--500次 | Y | 可以直接调用 | 在订单允许的最晚取消时间内，可以通过接口取消订单。  数据的返回格式比较整齐，用gzip模式获取，压缩比非常大，速度响应会很快。强烈建议用gzip模式。 |
| <?xml version="1.0"?>  <Request>  <Header $headerRight/>  <HotelRequest>  <ns:OTA\_CancelRQ TimeStamp="2012-04-20T00:00:00.000+08:00" Version="1.0">  <ns:UniqueID ID="159c906a-aa28-4f54-b609-59d2c105fde2" Type="1"/>//用户的携程UniqueID  <ns:UniqueID ID="XX" Type="28"/>//联盟的ID，Allianceid  <ns:UniqueID ID="XXX" Type="503"/>//联盟的SID  <ns:UniqueID ID="XXXXX" Type="501"/>//订单号  <ns:Reasons>  <ns:Reason Type="506"/>//订单取消  </ns:Reasons>  </ns:OTA\_CancelRQ>  </RequestBody>  </HotelRequest>  </Request> | | | | | |

2.6.5联盟退款

如果联盟在进行订单状态同步时，携程有返回订单取消成功的记录，说明携程已经成功退款到联盟在携程的账户上，那么联盟应该进入自己的退款流程将金额返还给用户并更新本地的订单状态。

# 3.优化实践

## 增量更新

把增量更新接口返回的数据放到一个队列里面，有多个线程访问队列更新价格计划。这样可以保持最大限度及时更新价格计划，同时防止价格计划更新被遗漏。

## 提升API调用性能

API调用是跨公网完成的，需要尽量优化降低网络上消耗的时间。可以启用对内容的GZIP压缩来减少数据包大小，启用HTTP链接的keep-alive特性减少网络等待时间。

## 详细的日志记录

系统一定得在关键点上记录详细的日志，日志是可以区分等级，最详细的基本应该记录发送请求的数据和返回的数据。出现错误调用的日志应该记录最详细的信息。当出现异常情况的时候，分析日志是非常有效的方法。

## 良好的错误处理

API调用参数都有相关的校验规则，再进行调用前须本地完成参数的校验，更好是用户输入或选择的时候进行前端校验，不要出现等待费时的远程错误。错误的提示，可以根据错误码区分不同的错误类型，更人性化的提示用户。