OTA Model
Getting Started Guide

Version 1.1

11/16/2016

Contents

REVISION HISTORY

Date	Version	Revisions
2016/10/03	1.0	Initial version for OTA Model Getting Started Guide
2016/11/16	1.1	Update benefits information in Certification Checklist

1. Overview

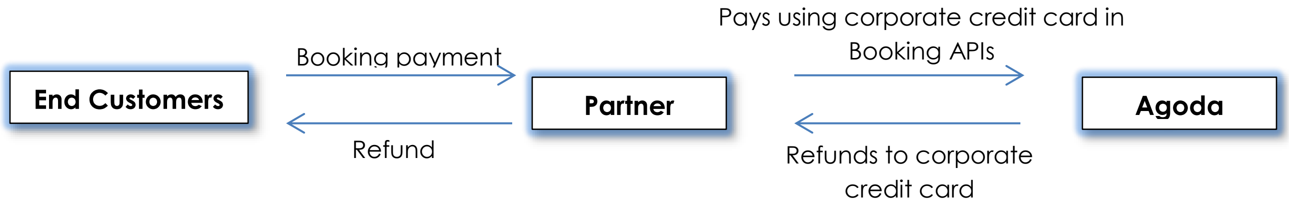
Agoda is a pure online retailer and is self-service oriented. Under OTA Model integration model, all of our self-service functionalities are provided to partner via XML APIs in order to resemble functionalities on agoda.com website. By using these APIs, partner will be able to process the payment and provide post-booking customer service directly to the end-customers.

Important
Integrations of all Search/Book/Customer Services APIs are mandatory.

1.1 Payment

Partner processes all charges and refunds from and to customers.

Agoda charges and refunds partner's Corporate Credit Card, which is submitted via XML API for each booking.

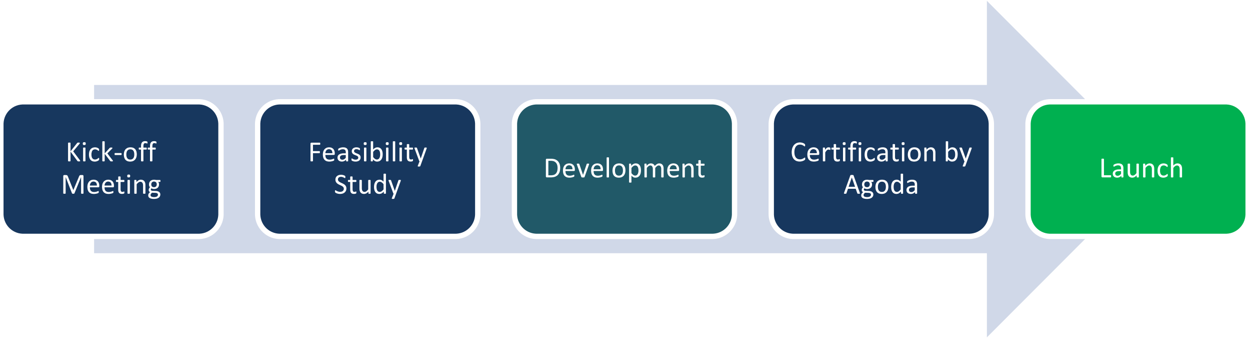

End Customers
Booking payment
Refund
Partner
Pays using corporate credit card in Booking APIs
Refunds to corporate credit card
Agoda

1.2 Customer Services and Customer Voucher

Under OTA model, partner sends out Customer Voucher. To avoid problems during check-in, customer voucher needs to follow Agoda policy and to be explicitly approved by Agoda prior to launch.
Please refer to section 3 of this documentation for our Customer Voucher policy.

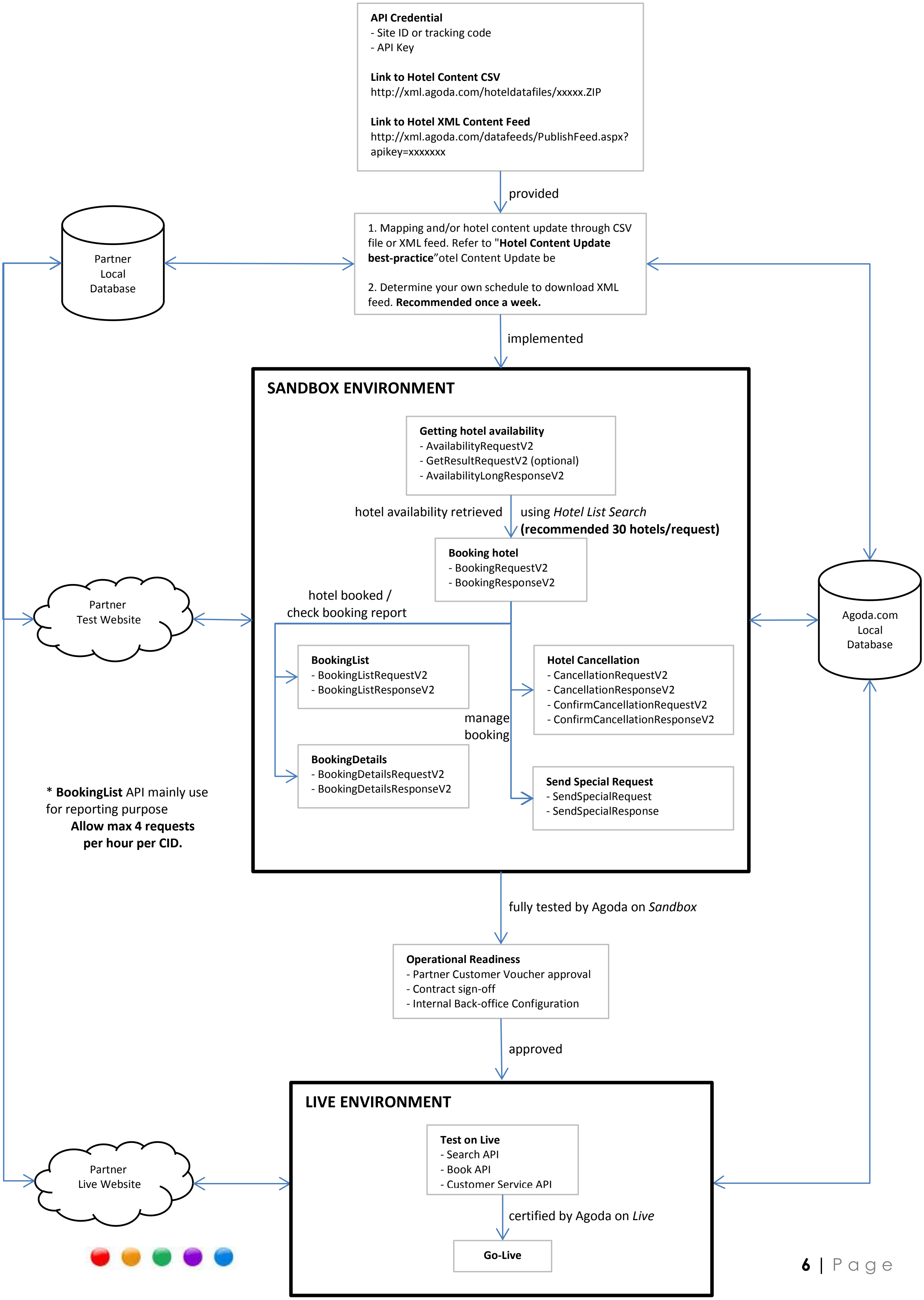
1.3 Implementation Plan

The flow of implementation can be depicted in the diagram below.



1. Kickoff Meeting – where scope, project timeline, and any other matters are discussed
2. Feasibility Study – this allows partner to perform some analysis and raise queries to Agoda
3. Development – the development phase by Partner. We stand by to support any enquiries
4. Agoda Certification: Pre-Live Testing and Voucher Approval – Agoda will also co-perform UAT to ensure the integration works correctly
5. Launch – the usual lead time between the completion of Certification and launch is about one week

Below flow chart illustrates step-by-step implementation processes.



API Credential
- Site ID or tracking code
- API Key

Link to Hotel Content CSV
http://xml.agoda.com/hoteldatafiles/xxxxx.ZIP

Link to Hotel XML Content Feed
http://xml.agoda.com/datafeeds/PublishFeed.aspx?apikey=xxxxxxx

provided

1. Mapping and/or hotel content update through CSV file or XML feed. Refer to "**Hotel Content Update best-practice**"otel Content Update be

2. Determine your own schedule to download XML feed. **Recommended once a week.**

Partner
Local
Database

implemented

SANDBOX ENVIRONMENT

Getting hotel availability
- AvailabilityRequestV2
- GetResultRequestV2 (optional)
- AvailabilityLongResponseV2

hotel availability retrieved

using *Hotel List Search*
(recommended 30 hotels/request)

Booking hotel
- BookingRequestV2
- BookingResponseV2

hotel booked /
check booking report

BookingList
- BookingListRequestV2
- BookingListResponseV2

BookingDetails
- BookingDetailsRequestV2
- BookingDetailsResponseV2

manage
booking

Hotel Cancellation
- CancellationRequestV2
- CancellationResponseV2
- ConfirmCancellationRequestV2
- ConfirmCancellationResponseV2

Send Special Request
- SendSpecialRequest
- SendSpecialResponse

Partner
Test Website

Agoda.com
Local
Database

* **BookingList** API mainly use
for reporting purpose
**Allow max 4 requests
per hour per CID.**

fully tested by Agoda on *Sandbox*

Operational Readiness
- Partner Customer Voucher approval
- Contract sign-off
- Internal Back-office Configuration

approved

LIVE ENVIRONMENT

Test on Live
- Search API
- Book API
- Customer Service API

certified by Agoda on *Live*

Go-Live

Partner
Live Website

2. IMPLEMENTATION

2.1 API Credentials

Agoda will give every partner their own unique key (AID-affiliate ID, Site ID, and API Key) to access our APIs. You can retrieve hotel information, make bookings, retrieve booking detail information, send special requests to properties, cancel bookings, and so on using XML requests.

Your Site ID and API Key will be provided by Agoda Account Manager prior to the implementation:

Your Affiliate ID (AID)	xxxxxx

Site ID (CID)	**API Key**
xxxxxxxx	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

2.2 Hotel Data

Refer to "**Hotel Content Update best-practice**” document. There are two options available:

XML Content Feeds (Support multiple languages and currencies)

http://xml.agoda.com/datafeeds/PublishFeed.aspx?apikey=xxxxxxxxxxxxxxxxxxx

Instructions on how to pull these content feeds as well as example requests and responses can be found in the link above.

Static CSV File

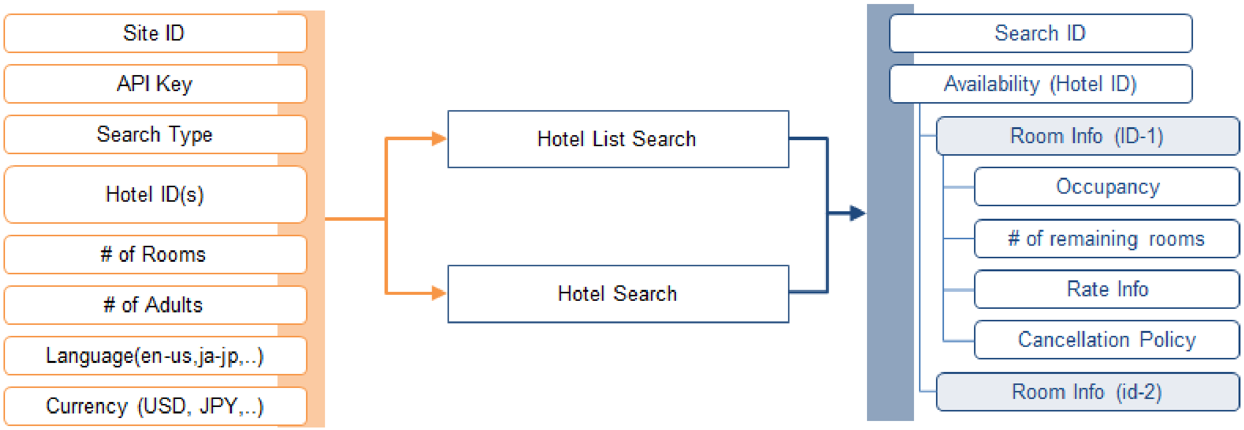
The hotel data file is used as a static base point, used by the partner to incorporate Agoda's inventory into their existing DB. This hotel data file is being refreshed once a week and needs to be updated periodically.

http://xml.agoda.com/hoteldatafiles/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx_EN.zip

This section of the document explains Agoda’s XML APIs, which is illustrated by Agoda’s booking workflow as below:

2.3 Availability Search

Availability Search APIs are used for retrieving hotel and room information before bookings are made. At this step the Partner is expected to pass information such as hotel IDs, number of rooms and guests, etc. in form of an XML request. Agoda will then process the request and response back with hotel and room information. Details can be found in API specification document – Affiliate API Availability Services.



2.3.1 Hotel List Search

Hotel List Search provides concurrent searches for a list of hotel IDs, which can be obtained from the data feeds/files. Hotel List Search is Agoda's preferred method for search requests as it gives partners as well as end customers' optimal performance.

Search type ID: 6

Example of Hotel List Search

```
<?xml version="1.0" encoding="utf-8" ?>
<AvailabilityRequestV2 siteid="xxxxxx" apikey="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
async="false" waittime="12" xmlns="http://xml.agoda.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Type>6</Type>
<Id>xxxxx,xxxxx,xxxxx</Id> <- List of Agoda hotel IDs within the same city, separated by comma.
<Radius>0</Radius>
<Latitude>0</Latitude>
<Longitude>0</Longitude>
<CheckIn>2012-09-21</CheckIn>
<CheckOut>2012-09-25</CheckOut>
<Rooms>1</Rooms>
<Adults>2</Adults>
<Children>0</Children>
<Language>th-th</Language>
<Currency>USD</Currency>
</AvailabilityRequestV2>
```

Practical use of "Hotel List Search"

1. Sending 30 hotel IDs per single search request using *AvailabilityRequestV2* API. This is to optimize the servers' performance and to give you the satisfied response time.
2. If you do wish to get an availability of more hotels (e.g. 300 hotels), it is highly recommended that you break the hotel list into many smaller requests (30 hotels per each request, all sent to the API in parallel). For each search request, you will receive "searchid", which is a unique reference for each search request/response. When you send multiple search requests to get availability of e.g. 300 hotels, you can use each XML response (which links to each searchid) to form a hotel results on your end. After that when a customer wants to make a booking on a hotel, please ensure that the booking belongs to a correct searchid.
3. Number of hotels returned is subjected to hotels availability put in a search criteria – meaning there may be less than 30 hotels being returned in single search result.
4. Searches with the same criteria does not need to be sent multiple times within a few minutes. XML API error rate is low and response time also fast (for hotel list search) so one request per criteria is fine.

2.3.2 Hotel Search

Hotel Search provides a result of a particular hotel as defined by Agoda hotel ID.

Search type ID: 4

Example of Hotel Search

```
<?xml version="1.0" encoding="utf-8" ?>
<AvailabilityRequestV2 siteid="xxxxxx" apikey="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
async="false" waittime="12" xmlns="http://xml.agoda.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Type>4</Type>
<Id>xxxxx</Id>  <- Agoda Hotel ID
<Radius>0</Radius>
<Latitude>0</Latitude>
<Longitude>0</Longitude>
<CheckIn>2012-09-21</CheckIn>
<CheckOut>2012-09-25</CheckOut>
<Rooms>1</Rooms>
<Adults>2</Adults>
<Children>0</Children>
<Language>th-th</Language>
<Currency>USD</Currency>
</AvailabilityRequestV2>
```

Important

Please ensure your website displays the correct room rates. If incorrect room rates are being displayed, partners will still be charged the correct rate by Agoda

Note that Mandatory surcharges are not included in the Rate inclusive tag; however, the mandatory surcharges will be automatically added into the total amount after the booking form has been submitted by Partner.

Example of rate with surcharge:

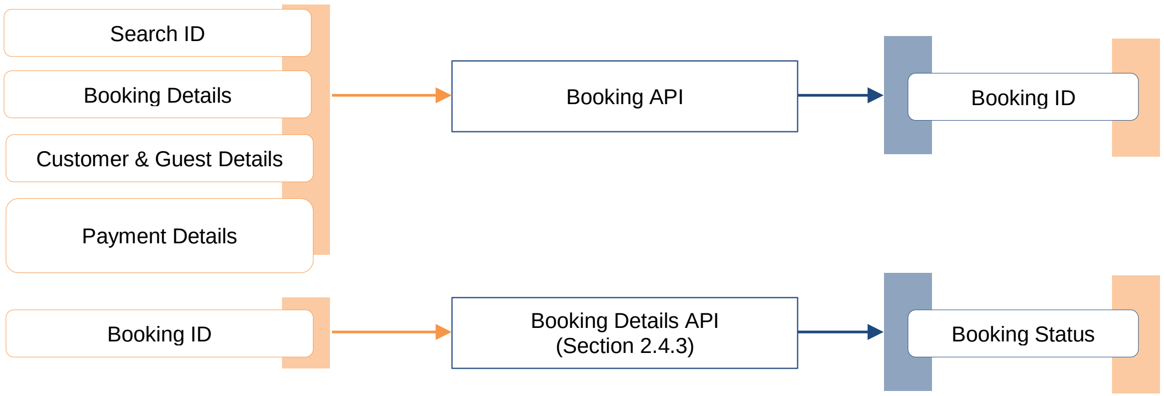
```
<RateInfo>
<Rate exclusive="214.99" tax="12.90" fees="4.30" inclusive="232.19"/>
..
<Included>Breakfast</Included>
<Surcharges>
    <Surcharge id="76" method="PAPN" charge="Mandatory" margin="n">
    <Name>Bed Tax 3 night(s) (Adult x 3)</Name>
    <Rate exclusive="44.46" tax="3.54" fees="0.00" inclusive="48.00"/>
</Surcharge>
</Surcharges>
```

Total Price = (Inclusive Per Room Per Night Rate x No. of Rooms x No. of nights)
+ Inclusive Surcharge Rates

2.4 Booking

2.4.1 Booking Request

To make a booking, the request sends Search ID together with Room Information, Customer and Guest details to Agoda's Booking functionality. The system then returns Booking ID.



Important
Customer information (i.e. name, email, and phone number) must be submitted via *BookingRequestV3* in <CustomerDetail> section.

Under OTA model, partner will pass customer information to Agoda via Booking Request API as part of reservation process. Required information consists of:

- **Language** – A language that a customer prefers to receive marketing communication
- **Title/First name/Last name** – Customer's title, first name, last name
- **Email** – Customer's email address
- **Phone** – Customer's phone number
- **Newsletter** – True/False; Customer "opt-in/opt-out" must be implemented in accordance with following scenario

Use Case	**Wording**	**<Newsletter> flag**
Tick to opt-out	I prefer not to receive promotions and updates from Agoda right now.	False
Tick to opt-in	Please send me updates on exclusive prices and last-minute deals from Agoda.	True
Dual provider tick to opt-out	I prefer not to receive promotions and updates from Agoda and *[partner name]* right now.	False
Dual provider tick to opt-in	Please send me updates on exclusive prices and last-minute deals from Agoda and *[partner name]*.	True

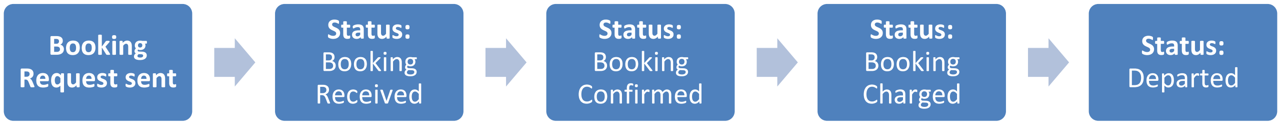
Example of Booking Request API Request

```
<?xml version="1.0" encoding="UTF-8"?>
<BookingRequestV3 siteid="123456" apikey="00000000-0000-0000-0000-000000000000"
xmlns="http://xml.agoda.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xml.agoda.com BookingRequestV3.xsd">
    … … (message hidden)
  <CustomerDetail>
    <Language>en-us</Language>
    <Title>Mr.</Title>
    <FirstName>John</FirstName>
    <LastName>Doe</LastName>
    <Email>johndoe@mail.com</Email>
    <Phone>
      <CountryCode>66</CountryCode>
      <AreaCode>2</AreaCode>
      <Number>6629231024</Number>
    </Phone>
    <Newsletter>true</Newsletter>
    <IpAddress>127.0.0.1</IpAddress>
  </CustomerDetail>
  <PaymentDetails>
    … … (message hidden)
  </PaymentDetails>
</BookingRequestV3>
```

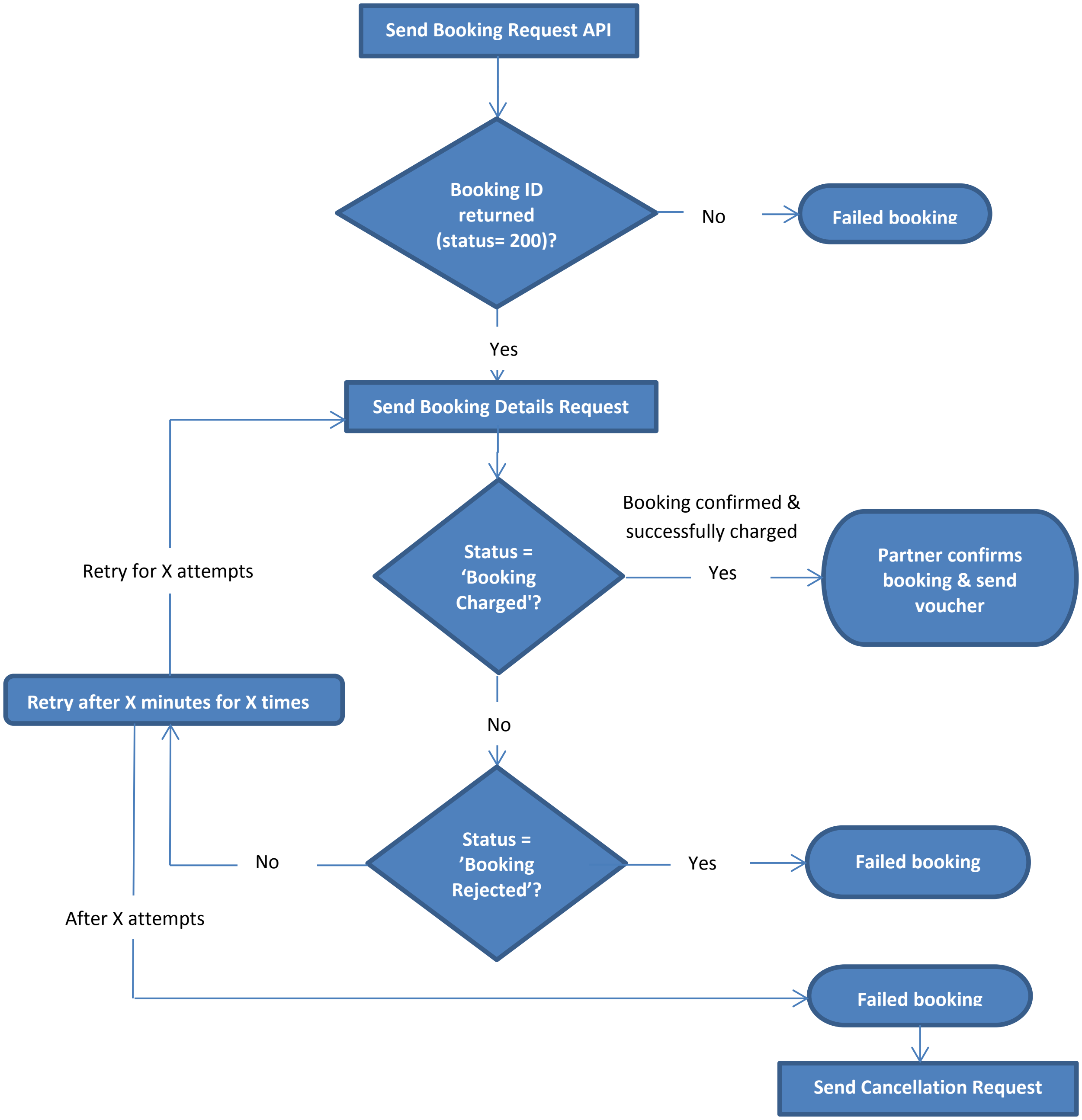
Booking ID

Booking ID is an identification number which is unique to each booking. This number is used for reference by Agoda, its partners, as well as hotels. Booking ID is located on the customer voucher and needs to be presented to the hotel by the customer upon arrival.

When a booking has been received by Agoda at the Booking Details step, the Booking Status begins as 'Booking Received' which means that the booking has been received by the system and is still being processed through the following steps.


Booking Request sent
Status: Booking Received
Status: Booking Confirmed
Status: Booking Charged
Status: Departed

For this reason, a ‘Booking Confirmation flow’, as illustrated in the below flowchart, will also need to be implemented.


Send Booking Request API
Booking ID returned (status= 200)?
No
Failed booking
Yes
Send Booking Details Request
Status = ‘Booking Charged'?
Booking confirmed & successfully charged
Yes
Partner confirms booking & send voucher
Retry for X attempts
Retry after X minutes for X times
No
Status = ’Booking Rejected’?
No
Yes
Failed booking
After X attempts
Failed booking
Send Cancellation Request

Booking Status

Booking Status is an indication of where each booking is within the process, and can be one of the values below:

- Booking Received
- Booking Charged
- Booking Confirmed
- Departed
- Booking Rejected
- Booking Cancelled
- Booking Cancelled By Customer
- Booking Test
- Technical Error

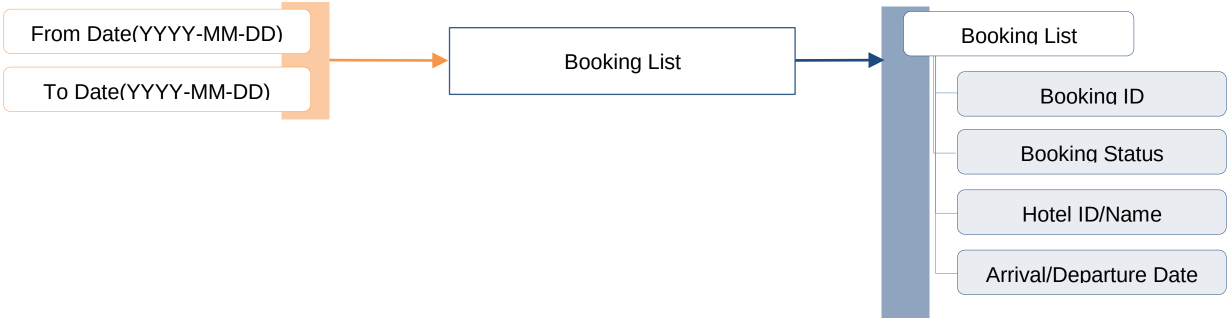
This is not to be confused with Booking Response Status, which serves a different purpose.

Booking Response Status

Booking Response Status (e.g. 200 – OK) is a response back from the system to indicate whether or not the request sent has been successfully accepted and processed on the system or not.

Please refer to the Appendix of the Technical Specification Document for more details in Booking Response Status.

2.4.2 Booking List



Booking List is used for retrieving a list of bookings within a specified range (From Date – To Date), the system returns Booking IDs, Booking Status, Hotel ID/Name, and Arrival/Departure Date.

Practical use of “Booking List” API

BookingList API mainly use for reporting purpose. Allowing only 4 requests per hour per CID – meaning that partner can call this API in every 15 minutes and expected specific date range should **NOT** more than 2 days per single request.

Example of Booking List API Request

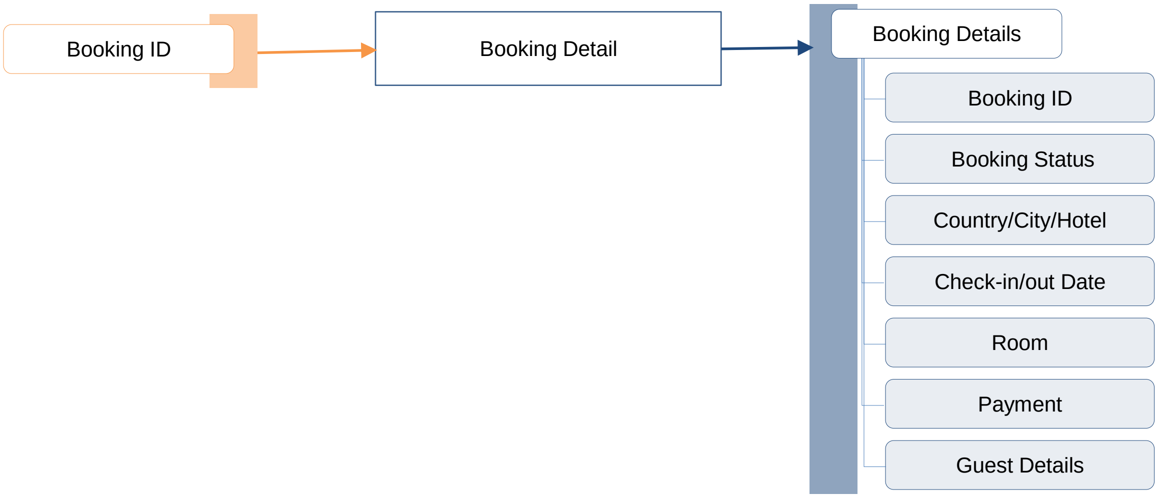
```
<?xml version="1.0" encoding="utf-8" ?>
<BookingListRequestV2 siteid=" xxxxxx " apikey=" xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
xmlns="http://xml.agoda.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <DateRange>
        <FromDate>2013-12-07</FromDate>
        <ToDate>2013-12-08</ToDate>
     </DateRange>
</BookingListRequestV2>
```

Example of Booking List API Response

```
<?xml version="1.0" encoding="utf-8"?>
<BookingListResponseV2 status="200" xmlns="http://xml.agoda.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <Bookings>
        <Booking id=" xxxxxx " status="BookingTest" tag="00000000-0000-0000-0000-000000000000"
        hotelid="71656" hotelname="X Test Hotel &amp; Spa - Do Not Book" cityname="Not Specified"
        received="2013-01-10" lastmodified="2013-01-10" arrival="2013-01-14" departure="2013-01-15"
        usdamount="0.00" selfservice="http://www.agoda.com/mybooking/customer-
        request.aspx?MjU2NDgwNzAequAL-VwWEzMMjS2gequAL" />
        <Booking id=" xxxxxx " status="BookingCancelledByCustomer" tag="" hotelid="159266"
        hotelname="Partners Test Hotel - DO NOT BOOK" cityname="Not Specified" received="2013-02-
        08" lastmodified="2013-02-08" arrival="2013-03-16" departure="2013-03-17" usdamount="0.00"
        selfservice="http://www.agoda.com/mybooking/customer-request.aspx?MjY3NDI5NzkequAL-
        EXrODAf+NNMequAL" />
    </Bookings>
</BookingListResponseV2>
```

2.4.3 Booking Details

To obtain details of each booking that has already been created in Agoda's system, e.g. Booking Status, Location, Check-in/out dates, Room, Guest Details, partners can send a booking detail request



Example of Booking Detail Request

```
<?xml version="1.0" encoding="UTF-8"?>
<BookingDetailsRequestV2 siteid="xxxxxx " apikey=" xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx "
xmlns="http://xml.agoda.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <BookingID> xxxxxxxxx</BookingID>
    <BookingID> xxxxxxxxx</BookingID>
</BookingDetailsRequestV2>
```

Example of Booking Detail API Response

```
<?xml version="1.0" encoding="utf-8"?>
<BookingDetailsResponseV2 status="200" xmlns="http://xml.agoda.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Bookings>
        <Booking>
            <BookingID> xxxxxxxxx </BookingID>
            <Tag>00000000-0000-0000-0000-000000000000</Tag>
            <Status>BookingTest</Status>
            <Country>Zimbabwe</Country>
            <City>Not Specified</City>
            <Hotel>X Test Hotel &amp; Spa - Do Not Book</Hotel>
            <CheckInDate>2013-01-14</CheckInDate>
            <CheckOutDate>2013-01-15</CheckOutDate>
            <Room>
                <RoomType>Magic Room</RoomType>
                <RoomsBooked>1</RoomsBooked>
            </Room>
            <TotalRateUSD rateplan="Retail" ratetype="Sell" exclusive="0.00" tax="0.00" fees="0"
           inclusive="0.00" />
            <Payment>
                <PaymentRateInclusive currency="USD">0.00</PaymentRateInclusive>
            </Payment>
            <GuestDetails>
                <GuestDetail Primary="true">
                    <Title>Mr.</Title>
                    <FirstName>Sample</FirstName>
                    <LastName>Sample</LastName>
                    <CountryOfPassport>Netherlands</CountryOfPassport>
                </GuestDetail>
            </GuestDetails>
            <SelfServiceURL>http://www.agoda.com/mybooking/customer-
           request.aspx?MjU2NDgwNzAequAL-VwWEzMMjS2gequAL</SelfServiceURL>
        </Booking>
    </Bookings>
</BookingDetailsResponseV2>
```

2.5 Customer Service

Agoda will provide several kinds of XML API for customer service. Partners can cancel, or send special request to properties by using XML requests by referring to Booking ID. Customer Services XML consists of the following: Cancellation, Confirm Cancellation, and Send Special Request.

Important

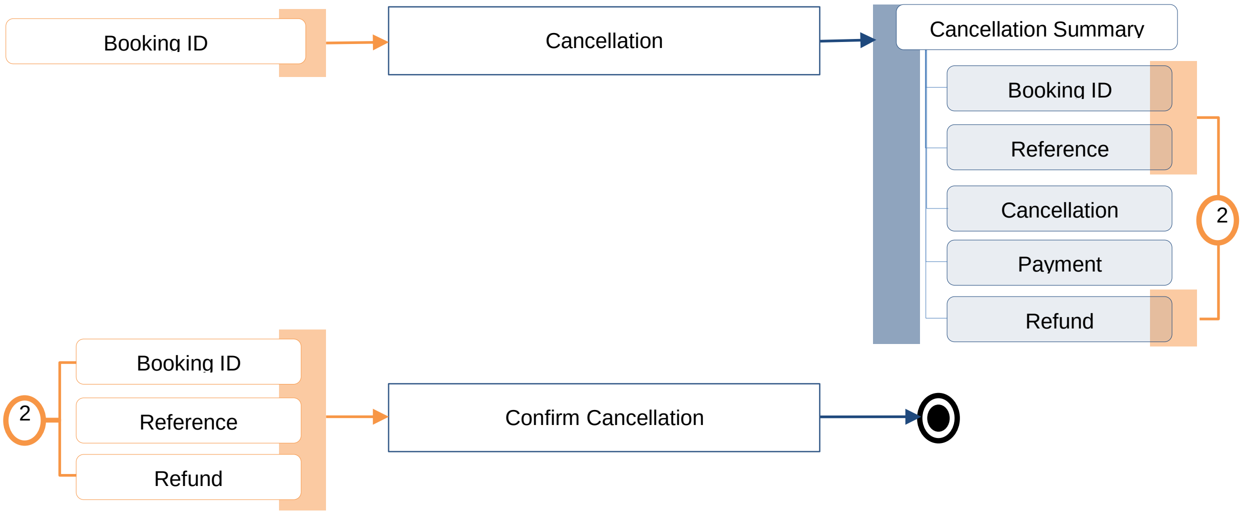
Because customer contact information such as e-mail, billing information, and correct names, are in the Partner's side and not Agoda, the Partner is required to also integrate our Customer Service APIs (i.e., cancellation, confirm cancellation, and special requests).

2.5.1 Cancellation

Partner can cancel their booking by sending cancellation request. Cancellation has two steps:

1. Cancellation Request
2. Confirm Cancellation Request

To cancel the booking, partner send request with booking id. When partner requests Cancellation, partner can recognize the cancellation policy. Based on the policy of cancellation, it is possible that the hotel price can incur either full/partial refund amount. After checking cancellation policy, partner can send requests with booking id, reference id, and refund amount.



Example of Cancellation (Request)

```
<?xml version="1.0" encoding="utf-8" ?>
<CancellationRequestV2 siteid="xxxxxx" apikey="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
xmlns="http://xml.agoda.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BookingID>3107586</BookingID>
</CancellationRequestV2>
```

Example of Cancellation (Response)

```
<?xml version="1.0" encoding="utf-8" ?>
<CancellationResponseV2 status="200" xmlns="http://xml.agoda.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <CancellationSummary>
            <BookingID>3107586</BookingID>
            <Reference>45886</Reference>
           <Cancellation>
                <PolicyText language="en-us">
                     Any cancellation received within 2 day/s prior to arrival will incur the first night/s …
                </PolicyText>
                …
           </Cancellation>
           <Payment> … </Payment>
           <Refund> … </Refund>
      </CancellationSummary>
</CancellationResponse>
```

2.5.2 Confirm Cancellation

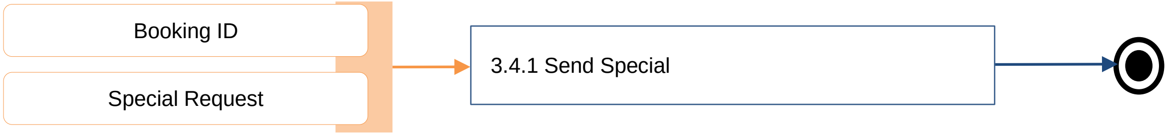
Example of Confirm Cancellation (Request)

```
<?xml version="1.0" encoding="utf-8" ?>
<ConfirmCancellationRequestV2 siteid="xxxxxx" apikey="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
xmlns="http://xml.agoda.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BookingID>3107586</BookingID>
  <Reference>45886</Reference>
  <CancelReason>1</CancelReason>
  <Refund>
    <RefundRateInclusive currency="THB">1100.00</RefundRateInclusive>
  </Refund>
</ConfirmCancellationRequestV2>
```

Example of Confirm Cancellation (Response)

```
<?xml version="1.0" encoding="utf-8" ?>
<ConfirmCancellationResponseV2 status="200" xmlns="http://xml.agoda.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</ConfirmCancellationResponseV2>
```

2.5.3 Send Special Request



Through “Send Special Request” API, partners can send special requests such as “late check in”, “smoking-room”, “twin-beds”, “high floor”, and etc. The key values consist of booking id & special request text.

Important

Note that special request is not guaranteed and is subjected to the property’s service-ability.

Example of Send Special (Request)

```
<?xml version="1.0" encoding="utf-8" ?>
<SendSpecialRequest siteid="xxxxxx" apikey="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
xmlns="http://xml.agoda.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BookingID>3107586</BookingID>
  <SpecialRequest>Special request text</SpecialRequest>
</SendSpecialRequest>
```

Example of Send Special (Response)

```
<?xml version="1.0" encoding="UTF-8" ?>
<SendSpecialResponse status="200" xmlns="http://xml.agoda.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

3. Customer Vouchers

To avoid confusion during check-in, please ensure your customer vouchers follow the policies as mentioned here below. Voucher format will require approval by Agoda prior to launch.

1. At top of voucher in English make clear that booking is **Pre-paid**.
2. Include Agoda's name and Booking ID. Booking ID must be clearly stated as an 'Agoda' Booking ID.
3. agoda.com's logo is mandatory. Please use "agoda.com" logo from https://mediaroom.agoda.com/materials-download/
4. Following to be included in English –
    - Hotel name & address
    - Check-in/out dates
    - Room Type
    - Room Quantity
    - Number of PAX in booking (adults, children, extra beds)
    - Meal plan
    - Cancellation policy
    - Special request. Need to stress that special request is "subject to hotel confirmation"
5. Cancellation policy to be in English as well as in preferred language.
6. First name and last name should be easily distinguishable.
7. Date format should either be 04 Mar 2012, or indicate what format it is eg. dd/mm/yy to avoid confusion between month and day.
8. Add your customer care contact details. This has to come with clear wording that your customer service is the default customer service for customers.

Important
Customer vouchers will be approved by Agoda only if they follow the policies here above.

4. Testing on Sandbox Environment

4.1 Testing Policy

Agoda provides a sandbox environment which can be used for creating test bookings. Partners are allowed to test only on the sandbox environment before attempting to test on the live environment. No testing on the live environment is allowed before explicit approval by your Agoda Account Manager. Please note that unauthorized test bookings on the live environment may incur a non-refundable cancellation fee.

4.2 Sandbox Endpoint URLs

To test integrations on the sandbox, the API request URL needs to be set to the following Endpoint URLs listed below. Once on the sandbox all the functionality needed to test XML integrations is provided.

Partners are allowed unlimited test bookings on Agoda's sandbox environment.

Endpoint URL Lists for XML API Requests

Request	URL
XML Availability	http://sandbox-affiliateapi.agoda.com/xmlpartner/xmlservice/search_lrv3
XML Booking	https://sandbox-affiliateapisecure.agoda.com/xmlpartner/xmlbookservice/book_v3
XML Booking List	https://sandbox-affiliateapisecure.agoda.com/xmlpartner/xmlbookservice/booklist_v2
XML Booking Details	https://sandbox-affiliateapisecure.agoda.com/xmlpartner/xmlbookservice/bookdetails_v2
XML Cancellation	https://sandbox-affiliateapisecure.agoda.com/xmlpartner/xmlbookservice/cancel_service
XML Confirm Cancellation	https://sandbox-affiliateapisecure.agoda.com/xmlpartner/xmlbookservice/confirmcancel_service
XML Send Special Request	https://sandbox-affiliateapisecure.agoda.com/xmlpartner/xmlbookservice/sendspecialrequest_service

5. AGODA CERTIFICATION

Once your site has been fully tested with Agoda's Sandbox environment and the confirmation vouchers have been approved by Agoda, your site will then need to be verified by Agoda Account Manager before it can go live. This ensures proper integration between Partner and Agoda.

Please submit test site URL as well as login credentials for testing. We will then conduct a test based on the scenarios below.

PASS/FAIL

1.	**Hotel Database**	
1.1	Is hotel and room level information correctly displayed?	
1.2	Is room-level cancellation policy being displayed correctly?	
1.3	Is room-level benefits being mentioned clearly?	
1.4	Is child age policy being mentioned clearly?	
1.5	Hotel information is presented with hotel remark?	
1.6	Is the hotel content refreshed regularly as suggested in 'Hotel Content Update best-practice'?	
2.	**Searching**	
2.1	Location search returning correct results using XML hotel list search (30 hotels/request)	
2.2	Are mandatory surcharges (where applicable) clearly mentioned that this amount will be charged as part of total booking?	
2.3	Are excluded surcharges (where applicable) clearly mentioned that they are extra charges which will be collected at hotel directly?	
2.4	Are hotel cancellation policies clearly displayed?	
3.	**Booking**	
3.1	Is total price correctly calculated and displayed?	
3.2	Are mandatory surcharges (where applicable) included in total price?	
3.3	Booking confirmation flow has been implemented	
3.4	Lead guest information mandatory	
4.	**Customer Information**	
4.1	Customer information (i.e. name, email, and phone number) are submitted as part of reservation process	
4.2	Newsletter "opt-in/opt-out" flag & checkbox are implemented in accordance with the guideline	
5.	**Customer Services**	
5.1	Customer service/self-service can retrieve booking list	
5.2	Customer service/self-service can retrieve booking details	
5.3	Customer service/self-service can see inclusive price	
5.4	Customer service/self-service can see exclusive price	
5.5	Customer service/self-service can see surcharge details	
5.6	Customer service/self-service can send special request	

5.7	Cancellation API has been implemented	
5.8	Cancellation with refundable amount	
5.9	Cancellation reason is sent through Cancellation API	
5.10	Customer service can retrieve cancellation policy	
5.11	Customer service can retrieve hotel remark	
6.	**Customer Voucher**	
6.1	Make clear that booking is Pre-paid	
6.2	Include Agoda's name and Booking ID. Booking ID must be clearly stated as an 'Agoda' Booking ID)	
6.3	Agoda's logo is mandatory	
6.4	Followings included in English – - Hotel name & address - Check-in/out dates - Room Type - Room Quantity - Number of PAX in booking (adults, children, extra beds) - Meal plan - Room conditions - Cancellation policy - Special request. Need to stress that special request is "subject to hotel confirmation"	
6.5	Cancellation policy in English as well as in preferred language	
6.6	First name and last name should be easily distinguishable	
6.7	Date format should either be 04 Mar 2012, or indicate what format it is eg. dd/mm/yy to avoid confusion between month and day	
6.8	Add your customer care contact details. This has to come with clear wording that your customer service is the default customer service for customers	
7.	**Reporting**	
7.1	Booking List and Booking Details APIs are implemented and are retrieve at least once a day to ensure all booking status is in sync with Agoda environment.	