Hotwire QuickConnect™
API User Documentation
Document Version 1.2
December 17, 2012

Contents

1. Overview

Hotwire QuickConnect is a set of APIs that allow a client system to interact with the information available in the Hotwire Hotel Extranet (https://extranet.hotwire.com). In turn, that information is used by Hotwire.com and its affiliates to display and sell hotel rooms to travelers.

Currently the Hotwire QuickConnect API can be used to create and update a hotel's availability, rates, and inventory (ARI), and retrieve recently confirmed and/or cancelled bookings. The ARI functionality includes the ability to set the number of available rooms, per-day rate amounts and extra person charges, and several restrictions. Retrieved bookings contain guest and Hotwire single use credit card information.

1.1. Relationship to Expedia QuickConnect®

Hotwire QuickConnect uses a message set similar to Expedia QuickConnect®. However Hotwire QuickConnect is a separate service. Updates made using Hotwire QuickConnect will not impact Expedia QuickConnect® or vice versa.

1.2. Registration

In order to use Hotwire QuickConnect, a prospective user must be registered with an Expedia QuickConnect® account. Users may register at http://www.expediaquickconnect.com.

A new Hotwire QuickConnect user must also register for an account at http://developer.hotwire.com and apply for an API key. Please contact Hotwire at hqc@hotwire.com to activate the API key. Once active, the API key must be provided with every request made to the API.

1.3. Endpoints

Production SOAP WSDL location: https://api.hotwire.com**/xnetApi/v1.1/**XnetHotelService?wsdl
Non-prod SOAP WSDL location: https://api.preprod.hotwire.com**/xnetApi/v1.1/**XnetHotelService?wsdl

1.4.Communication Protocol

A client may communicate with the Hotwire QuickConnect system using an HTTP request with an embedded SOAP based message. All request/response transactions are synchronous. All messages must be sent as HTTP posts.

The Hotwire QuickConnect API currently accepts requests wrapped in a SOAP Envelope, and returns responses wrapped in a SOAP Envelope.

Here is the structure of an example request or response message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.xnet.hotwire/">
  <soapenv:Header/>
  <soapenv:Body>
    <!-- API message payload here -->
  </soapenv:Body>
</soapenv:Envelope>
```

1.5.Authentication

1.5.1. API Key Authentication

Every request sent to Hotwire QuickConnect must contain a valid API Key. The API Key identifies the system sending the request. A single API key may be used to send requests on behalf of many hotels.

A client application should append the API key as a parameter to the request URL.

1.5.2. Extranet User Authentication

Every request sent to Hotwire QuickConnect must contain the username and password of a valid Hotwire Hotel Extranet user who is:

- linked to the Hotel whose information is being accessed
- authorized to perform the requested function

For instance there is a ‘view-only’ user role in the Hotwire Extranet. If a user is configured with the ‘view only’ role at a hotel, then an API request sent with their credentials that changes rates will fail.

An example HTTP request with both the API Key and Extranet user authentication looks like this:

```
POST xnetApi/v1.1/XnetHotelService?apikey=%KEY%
Host:  https://api.hotwire.com
Authorization: Basic YWNjb3VudEBmYWtlaG90ZWwuY29tOmY0azNwdzk=
```

Where the ‘Authorization’ string is the base64 encoding of the string ‘username:password’ and ‘%KEY%’ is the client’s registered API key.

1.6.Namespaces

The Hotwire QuickConnect system uses two namespaces:

- SOAP Namespace= http://schemas.xmlsoap.org/soap/envelope/
- API namespace = http://api.xnet.hotwire/

1.7.Versioning

API version information is provided in the path to the service. The current version is v1.1.

1.8.Message Sequencing

The Hotwire QuickConnect API will process messages in the order in which they are received. Please take care when sending updates that may affect the same pieces of hotel inventory. If such messages are sent close together in time, Hotwire cannot guarantee that the client's desired order of processing will be preserved.

1.9.Rate Limiting

The Hotwire QuickConnect API will apply rate limiting to user requests. Rate limits will be set at the API key level. The default rate limit in production is 5000 requests per day and 5 requests/second. Rate limits can be modified by contacting Hotwire.

The API will respond to a request which exceeds a rate limit with a 403 error with an error message such as "Over Queries Per Second Limit", "Over rate limit", or "Service Over Queries-per-Second-Limit".

2. Error Messaging

2.1.System Error Messaging

System level errors will be returned to client systems using standard HTTP/1.1 error codes.

2.2.Authentication Errors

If a request message contains an invalid username and password, the API will respond with an 'HTTP/1.1 401 Unauthorized' message.

The API will respond with the same 'HTTP/1.1 401 Unauthorized' message if the user is not allowed to perform the requested operation in the Hotwire Extranet web application. For instance, a user with 'read only' privileges is not authorized to update rates.

2.3.Schema Validation Errors

The API system performs XSD schema validation on the data contained in each request message. If there is a XSD validation error the API will return an error wrapped in a SOAP <Fault> element.

Here is an example error returned when a non-integer value is passed in a field that should be of type integer:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>Unmarshalling Error: cvc-datatype-valid.1.2.1: '34323az' is not a valid value for
'integer'.</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Here is an example error where the request passed in an incomplete date value

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>Unmarshalling Error: cvc-datatype-valid.1.2.1: '2012-05-' is not a valid value
for 'date'.</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

2.4.Logical Error Messaging

The Hotwire QuickConnect API will return a list of error messages when a request message raises an error condition. The system will wrap any errors in a SOAP <Fault> element.

In some cases the API may return multiple error messages per request if there are multiple error conditions that pertain to the same part of the request message. For example, if the start date and end date of a date range are each in the past.

2.5.Error Message Format

Level	Data Element	Data Type	Required	Business Rules and Description
0	Errors		Yes	Container for <Error> elements
0	@echoToken	string	No	Returned if the request message contained an echoToken attribute
0	@timestamp	date time	Yes	Timestamp will be in ISO 8601 combined date and time format e.g. “2012-04-02T11:11:11.302-07:00” The time will always be returned with the time zone offset for California, USA.
1	Error	string	Yes	The text element child is the error message
1	@code	string	Yes	3-character error code

3. Update Inventory

The Hotwire QuickConnect update inventory request and response allows suppliers to create and update hotel availability, rates, and inventory (ARI).

The ARI API provides the ability to:

- Create new inventory for a specified range of stay dates
- Update existing inventory for a specified range of stay dates
- Set or modify available room counts
- Set or modify per day rates and extra person charges in the hotel's preferred currency
- Set or modify availability restrictions

3.1.Notes on default values and inventory creation

An API user can create new inventory with an <updateInventory> request message. For instance, if a hotel has not previously loaded any inventory for a given stay date range and room type either through the Extranet web application or API, it can do so with this message.

Most parts of the <updateInventory> request message are optional. When a request creates a piece of inventory that does not yet exist, and that request omits optional parts of the message, the unspecified parts of the inventory will be created with system default values as follows:

- total inventory available = 0
- rate amount per night = 0 (Hotwire.com uses a rate floor greater than zero, so this rate will not sell)
- extra person charge per night = 0
- closed = false
- minimum length of stay = 1
- maximum length of stay = the maximum number of days searchable on Hotwire.com, currently 30
- closed to arrival = false
- closed to departure = false
- maximum days to arrival = the maximum number of days in the future searchable on Hotwire.com, currently 330 days

3.2.Notes on new restrictions and the Hotwire Extranet web application

The Hotwire QuickConnect API includes three new restrictions. These restrictions are not directly viewable or modifiable in the Hotwire Extranet web application. They can only be set via the API. You can view and validate changes to the restrictions in the "review changes" section of the Hotwire Extranet web application.

The new restrictions are:

- closed to departure
- maximum length of stay
- maximum days to arrival

3.3. Update Inventory Request Message format:

Level	Data Element	Data Type	Required	Business Rules and Description
0	updateInventory		Yes	
1	AvailRateUpdateRQ		Yes	
1	@echoToken	string	No	The token will be returned in the response to the message. It can be up to 12 characters long.
2	Hotel		Yes	
2	@id	long integer	Yes	The hotel's unique Hotwire hotel id.
2	AvailRateUpdate		Yes	
3	DateRange		Yes	
3	@from	date	Yes	Use ISO date format yyyy-mm-dd. The start date of the range. The date must be equal to or greater than the current date.
3	@to	date	Yes	The last date of the range. It is included in the updates. The date must not be further than 15 months from the current date.
3	@sun	boolean	No	Apply the changes to all Sundays that fall within the date range. The value defaults to true if the attribute is not present.
3	@mon	boolean	No	Apply the changes to all Mondays that fall within the date range. The value defaults to true if the attribute is not present.
3	@tue	boolean	No	Apply the changes to all Tuesdays that fall within the date range. The value defaults to true if the attribute is not present.
3	@wed	boolean	No	Apply the changes to all Wednesdays that fall within the date range. The value defaults to true if the attribute is not present.
3	@thu	boolean	No	Apply the changes to all Thursdays that fall within the date range. The value defaults to true if the attribute is not present.
3	@fri	boolean	No	Apply the changes to all Fridays that fall within the date range. The value defaults to true if the attribute is not present.

3	@sat	boolean	No	Apply the changes to all Saturdays that fall within the date range. The value defaults to true if the attribute is not present.
3	RoomType		No	If present, cannot be empty
3	@id	string	Yes	The name of the room type, as defined in the Hotwire Extranet. This value is case sensitive.
3	@closed	boolean	No	The master open/close restriction on the room
4	Inventory		No	
4	@totalInventoryAvailable	integer	Yes	The total number of available rooms
4	RatePlan		No	If present, cannot be empty
4	@id	string	Yes	The rate plan id. Please send 'XHW' in all requests.
5	Rate		No	
5	@currency	string	Yes	The currency code of the rates
5	@perDay	float	No	The per day base rate amount
5	@extraPerson	float	No	The per day extra person amount
5	Restrictions		No	
5	@minLOS	integer	No	The minimum length of stay. The restriction is arrival-date based.
5	@maxLOS	integer	No	The maximum length of stay. The restriction is arrival-date based.
5	@closedToArrival	boolean	No	Closed to arrival restriction
5	@closedToDeparture	boolean	No	Closed to departure restriction
5	@maxDaysToArrival	integer	No	Maximum number of days to arrival restriction. Used for rates that a hotel wishes to open only in last minute circumstances.

3.4.Additional Message Validation

The API will validate that each <updateInventory> request message meets the following business rule checks not expressed by the API schema and WSDL.

- The echoToken attribute must be between 1 and 12 characters
- If there are multiple AvailRateUpdate elements in a request, then the stay dates and room types they update must not overlap. Doing so will fail the entire update. An example would be a request with multiple <AvailRateUpdate> elements, each of which updates the same RoomType on one or more overlapping stay dates.
- The hotel Id must be a valid Hotwire hotelID
- The hotel Id must be a number between 1 and 999,999,999
- totalInventoryAvailable must be a number between 0 and 4,999
- The 'from' date in a range must not be in the past; it can be the current date or greater

- The ‘to’ date in a range must not be in the past; it can be the current date or greater
- The ‘from’ date in a range must not be after the ‘to’ date
- The ‘to’ date must be within 15 months of the current date
- A date range cannot cover more than 60 days
- The room type ID must be between 1 and 12 characters long
- Inactive room types cannot be updated, they must be manually activated using the Hotwire Extranet web application prior to inventory management via API
- The Room type must exist for a hotel; new room types cannot be created with the API
- If present, the RoomType element cannot be empty
- If present, the RatePlan container element cannot be empty
- The currency code must be 3 characters long
- The currency code must match the currency used by the hotel
- Per day rate must be between 0 and 999,999 if present
- Extra person charge must be between 0 and 999,999 if present
- Minimum length of stay must be between 0 and 30
- Maximum length of stay must be between 1 and 30
- Maximum days to arrival must be between 0 and 330

3.5.Update Inventory Response Message Format

The system will return an <updateInventoryResponse> message if the request was successfully processed. A request that raises one or more errors will return a SOAP <Fault> element.

3.5.1. Successful Response Message Format

Level	Data Element	Data Type	Required	Business Rules and Description
0	updateInventoryResponse		Yes	
1	AvailRateUpdateResponse		No	
1	@version	string	Yes	Version of the Hotwire QuickConnect API
1	@timeStamp	date time	Yes	Timestamp will be in ISO 8601 combined date and time format e.g. “2012-04-02T11:11:11.302-07:00” The time will always be returned with the time zone offset for California, USA.
1	@echoToken	string	No	If submitted, the echoToken from the request
2	Success		No	Empty element indicating the requested operation succeeded

3.5.2. Error Response Message Format

If an <updateInventory> request raises a logical error condition then the system will return a SOAP <Fault> element. The fault will contain a collection of <Error> elements as detailed above in the Error Message specification.

3.6.Update Inventory Error List

Here is a list of the possible logical errors that can be returned in response to an <updateInventory> request.

Code	Message
100	EchoToken must be between 1 and 12 characters long
101	Dates and room types must not overlap
200	Unknown hotel id
201	Hotel id must be between 1 and 999999999
301	Number of total inventory available must be between 0 and 4999
400	Start date must not be in the past
401	End date must not be in the past
402	Start date must not be after end date
403	End date must be within 15 months in the future
404	End date must be within 60 days of the start date
500	Room type must not be more than 12 characters long
501	Inactive room type
502	Unknown room type
503	Room type must not be empty
600	Rate plan id must not be more than 12 characters long
601	Unknown rate plan
602	RatePlan must not be empty
700	Currency must be 3 characters long
701	Unknown currency
702	Per day rate must be between 0 and 999999
703	Extra person rate must be between 0 and 999999
800	Minimum length of stay must be between 0 and 30
801	Maximum length of stay must be between 1 and 30
802	Maximum days to arrival must be between 0 and 330

3.7.Update Inventory Request XML Examples

Example 1 sets the total inventory, rate, extra person charges, closed to arrival, and closed to departure for the STANDARD room for the date range August 1-31 2012

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://api.xnet.hotwire/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:updateInventory>
      <AvailRateUpdateRQ>
        <Hotel id="3546"/>
        <AvailRateUpdate>
          <DateRange from="2012-08-01" to="2012-08-31"/>
          <RoomType id="STANDARD">
            <Inventory totalInventoryAvailable="15"/>
            <RatePlan id="XHW">
              <Rate currency="usd" perDay="55.15" extraPerson="10.00"/>
              <Restrictions closedToArrival="false" closedToDeparture="false"/>
            </RatePlan>
          </RoomType>
        </AvailRateUpdate>
      </AvailRateUpdateRQ>
    </ns2:updateInventory>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 2 sets the rate amount for Fridays and Saturdays during the month of June 2012 for two room types:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://api.xnet.hotwire/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:updateInventory>
      <AvailRateUpdateRQ echoToken="09992391">
        <Hotel id="1109"/>
        <AvailRateUpdate>
          <DateRange from="2012-06-01" to="2012-06-30" fri="true" sat="true" sun="false"
mon="false" tue="false" wed="false" thu="false"/>
          <RoomType id="STANDARD">
            <RatePlan id="XHW">
              <Rate currency="usd" perDay="88"/>
            </RatePlan>
          </RoomType>
          <RoomType id="Courtyard">
            <RatePlan id="XHW">
              <Rate currency="usd" perDay="95"/>
```

```
            </RatePlan>
          </RoomType>
        </AvailRateUpdate>
      </AvailRateUpdateRQ>
    </ns2:updateInventory>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 3 closes out inventory for a single stay date:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://api.xnet.hotwire/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:updateInventory>
      <AvailRateUpdateRQ echoToken="90845544">
        <Hotel id="31145"/>
        <AvailRateUpdate>
          <DateRange from="2012-05-28" to="2012-05-28"/>
          <RoomType id="STANDARD" closed="true">
            <Inventory totalInventoryAvailable="1"/>
          </RoomType>
        </AvailRateUpdate>
      </AvailRateUpdateRQ>
    </ns2:updateInventory>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 4 sets weekend and weekday inventory and rates over the same date range for the same room type:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://api.xnet.hotwire/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:updateInventory>
      <AvailRateUpdateRQ>
        <Hotel id="5022"/>
        <AvailRateUpdate>
          <DateRange from="2012-05-01" to="2012-05-31" mon="false" tue="false" wed="false"
thu="false" fri="true" sat="true" sun="false"/>
          <RoomType id="STANDARD">
            <Inventory totalInventoryAvailable="6"/>
```

```
            <RatePlan id="XHW">
                <Rate currency="USD" perDay="55"/>
            </RatePlan>
          </RoomType>
        </AvailRateUpdate>
        <AvailRateUpdate>
            <DateRange from="2012-05-01" to="2012-05-31" mon="true" tue="true"
wed="true" thu="true" fri="false" sat="false" sun="true"/>
            <RoomType id="STANDARD">
              <Inventory totalInventoryAvailable="10"/>
              <RatePlan id="XHW">
                <Rate currency="USD" perDay="45"/>
              </RatePlan>
            </RoomType>
        </AvailRateUpdate>
      </AvailRateUpdateRQ>
    </ns2:updateInventory>
  </soapenv:Body>
</soapenv:Envelope>
```

3.8.Update Inventory Response XML examples

Example 1 shows a successful response:

```
<soap:Envelope xmlns:soap=" http://schemas.xmlsoap.org/soap/envelope/ ">
  <soap:Body>
    <ns2:updateInventoryResponse xmlns:ns2="http://api.xnet.hotwire">
      <AvailRateUpdateRS version="1" timeStamp="2012-03-29T09:10:11.796-07:00"
echoToken="1">
        <Success/>
      </AvailRateUpdateRS>
    </ns2:updateInventoryResponse>
  </soap:Body>
</soap:Envelope>
```

Example 2 shows an error response to a request that attempted to update inventory for an inactive room type:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Error submitting request</faultstring>
```

```
      <detail>
        <ns2:Errors echoToken="90845540" timestamp="2012-04-03T11:26:39.999-07:00"
xmlns:ns2="http://api.xnet.hotwire/">
          <Error code="501">Inactive room type</Error>
        </ns2:Errors>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

4. Booking Retrieval

The Booking Retrieval API provides the ability to:

- Retrieve information for confirmed and/or cancelled hotel bookings based on hotel id and activity date
- Securely obtain Hotwire single use credit card information

4.1.Polling

It is Hotwire's expectation that users of the Hotwire QuickConnect API will periodically poll the booking retrieval API to obtain recently booked and/or cancelled bookings. The Booking Retrieval API allows a calling system to specify a "MinsInPast" value in booking retrieval requests. To request information on bookings made or cancelled in the past 30 minutes, a calling system will send a request with MinsInPast = 30. The "MinsInPast" value may be a number between 30 and 360.

It is up to the implementor to determine the frequency of polling for confirmed bookings. The Hotwire Quick Connect API enforces rate limits on all implementor applications. Please contact Hotwire at hqc@hotwire.com to determine and set your application's rate limit.

4.2. Duplicate Bookings

Developers should take care not to create duplicated hotel bookings when polling.

- Every booking returned by the API will have a unique "id" attribute.
- The same booking id can be returned with either an active(@status ="Book") or cancelled (@status="Cancel") status.

To avoid sending duplicated bookings to hotel systems, an HQC user application should keep track of booking ids and statuses. The user application can use the combined id and status to identify already retrieved bookings before sending new bookings to a hotel system.

4.3.Security

The Booking Response message contains sensitive information:

- Hotwire single use credit card numbers
- Guest name information

All communication with the Hotwire Quick Connect API must be conducted over a secure HTTPS connection.

4.4.Booking Retrieval Request Message Format

Level	Data Element	Data Type	Required	Business Rules and Description
0	retrieveBooking		Yes	
1	BookingRetrievalRQ		No	
1	@echoToken	string	No	The token will be returned in the response to the message. It can be up to 12 characters long.

2	Hotel		No	Used in combination with <MinsInPast > element to retrieve bookings and cancellations at a particular hotel.
2	@id	long integer	Yes	The hotel's unique Hotwire hotel id.
2	Booking		No	Used to retrieve one specific booking (not yet available).
2	@id	string	Yes	The Hotwire itinerary ID of the requested booking.
2	MinsInPast	Integer	No	Used in combination with <Hotel> element to retrieve bookings and cancellations at a particular hotel.

4.5. Additional Message Validation

The Booking Retrieval API will validate that each <retrieveBooking> request message meets the following business rule checks not expressed by the API schema and WSDL.

4.6.Booking Retrieval Response Message Format

4.6.1. Successful Response Message Format

Level	Data Element	Data Type	Required	Business Rules and Description
0	retrieveBookingResponse		Yes	
1	BookingRetrievalRS		No	
1	@version	string	Yes	Version of the Hotwire QuickConnect API
1	@timestamp	dateTime	Yes	Timestamp will be in ISO 8601 combined date and time format e.g. "2012-04-02T11:11:11.302-07:00" The time will always be returned with the time zone offset for California, USA.
1	@echoToken	string		If submitted, the echoToken from the request.
2	Booking		No	
2	@id	string	Yes	Hotwire's itinerary ID.
2	@type	string	Yes	Whether the booking is active or cancelled. The value is "Book" for an active booking and "Cancel" for a cancelled reservation.
2	@createDateTime	datetime	Yes	For active bookings, the time the booking was confirmed on Hotwire.com. For cancelled bookings, the time that it was

				cancelled on Hotwire.com.
2	@source	string	Yes	The source of the booking. “HW” for Hotwire.com.
3	Hotel		Yes	
3	@id	long integer	Yes	The hotel’s unique Hotwire hotel id.
3	RoomStay			Complex container element for details of the room stay.
3	@roomTypeID	string	Yes	The name of the room type, as defined in the Hotwire Extranet. This value is case sensitive.
3	@ratePlanID	string	Yes	The rate plan id.
3	@numberOfRooms	integer	Yes	The number of rooms booked.
4	StayDate		Yes	
4	@arrivalDate	date	Yes	The check-in date. Format will be ISO YYYY-MM-DD.
4	@departureDate	date	Yes	The check-out date. Format will be ISO YYYY-MM-DD.
4	GuestCount		Yes	
4	@adult	integer	Yes	The total number of adults being booked into the reserved rooms.
4	@child	integer	Yes	The total number of children being booked into the reserved rooms.
4	PerDayRates		Yes	
4	@currency	string	Yes	The currency code of the rates.
5	PerDayRate		Yes	
5	@stayDate	date	Yes	The stay date. Format will be ISO YYYY-MM-DD.
5	@baseRate	float	Yes	The rate for the stay date.
4	TotalAmount			
4	@amountAfterTaxes	float	Yes	The total amount of the reservation – total base rate plus total taxes. The total amount to charge to the Hotwire card.
4	@amountOfTaxes	float	Yes	The total amount of taxes.
4	@currency	string	Yes	The currency code of the @amountAfterTaxes and @amountOfTaxes attributes.
4	PaymentCard		Yes	Details of the Hotwire single use credit card attached to the booking.
4	@cardHolderName	string	Yes	The name on the Hotwire card.
4	@cardNumber	string	Yes	The credit card number.
4	@cardCode	string	Yes	All Hotwire single use cards are CA –

				MasterCard.
4	@seriesCode	string	Yes	The card validation number.
4	@expireDate	string	Yes	The card expiration date. The format will be MMYYYY.
3	Guest		Yes	Container element for guest information.
4	PrimaryGuest		Yes	
4	@givenName	string	Yes	
4	@middleName	string	No	
4	@lastName	string	Yes	
3	SpecialRequest	string	No	Special request text. For instance, request for confirmed Bed Type.

4.6.2. Error Response Message Format

If a <retrieveBooking> request raises a logical error condition then the system will return a SOAP <Fault> element. The fault will contain a collection of <Error> elements as detailed above in the Error Message specification.

4.7. Booking Retrieval Error List

Code	Message
900	`Minutes in past must be between 30 and 360`
1000	`The current API version only supports booking retrieval by hotel id AND minutes in past.`

4.8. Booking Retrieval Simulator

The API endpoint in pre-production exposes a simulator that allows a user to retrieve example booking responses. This enables developers to test basic integration with the booking retrieval API without having to create test customer bookings in the pre-production environment.

To invoke the simulator:

- Provide your usual apiKey
- Use the authorization credentials 'hqcsim@hotwire.com:hotwire' (aHFjc2ltQGhvdHdpcmUuY29tOmhvdHdpcmU=)
- Send the request to the pre-production endpoint https://api.preprod.hotwire.com**/xnetApi/v1.1/**XnetHotelService?wsdl
- Send hotel id **17515**

The contents of the response will vary depending on the value sent in the MinsInPast parameter of the request

- If MinsInPast is less than or equal to 60, the response will be empty

- If MinsInPast is greater than 60, the response will contain several <Booking> elements with various guest, room, and length of stay combinations. One booking will be cancelled.

4.9.Booking Retrieval Request XML Examples

Example 1 shows a request for bookings made or cancelled at a hotel within the past 300 minutes.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://api.xnet.hotwire/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:retrieveBooking>
      <BookingRetrievalRQ echoToken="1111111">
        <Hotel id="34323"/>
        <MinsInPast>300</MinsInPast>
      </BookingRetrievalRQ>
    </ns2:retrieveBooking>
  </soapenv:Body>
</soapenv:Envelope>
```

4.10. Booking Retrieval Response XML Examples

Example 1 shows an empty response – there were no confirmed or cancelled bookings made within the requested number of minutes. The <BookingRetrievalResponse> element contains no child <Booking> elements.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:retrieveBookingResponse xmlns:ns2="http://api.xnet.hotwire/">
      <BookingRetrievalRS version="1" timeStamp="2012-10-15T10:57:00.777-07:00"
echoToken="1111111"/>
    </ns2:retrieveBookingResponse>
  </soap:Body>
</soap:Envelope>
```

Example 2 shows a response with multiple active and cancelled bookings.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:retrieveBookingResponse xmlns:ns2="http://api.xnet.hotwire/">
      <BookingRetrievalRS version="1" timeStamp="2012-10-19T10:49:55.543-07:00">
        <Booking id="58652752" type="Book" createDateTime="2012-10-19T10:46:51-07:00"
source="HW">
          <Hotel id="34323"/>
          <RoomStay roomTypeID="STANDARD" ratePlanID="XHW" numberOfRooms="1">
            <StayDate arrivalDate="2012-10-23" departureDate="2012-10-24"/>
            <GuestCount adult="2" child="0"/>
```

```
            <PerDayRates currency="AUD">
              <PerDayRate stayDate="2012-10-23" baseRate="33.3"/>
            </PerDayRates>
            <TotalAmount amountAfterTaxes="37.0" amountOfTaxes="3.7" currency="AUD"/>
            <PaymentCard cardHolderName="Test Booking" cardNumber="5555555555554444"
cardCode="CA" seriesCode="925" expireDate="122017"/>
          </RoomStay>
          <Guest>
            <PrimaryGuest givenName="Firstname " middleName="m" lastName="lastname "/>
          </Guest>
        </Booking>
        <Booking id=" 58652753" type="Book" createDateTime="2012-10-19T10:47:42-07:00"
source="HW">
          <Hotel id="34323"/>
          <RoomStay roomTypeID="STANDARD" ratePlanID="XHW" numberOfRooms="2">
            <StayDate arrivalDate="2012-10-23" departureDate="2012-10-25"/>
            <GuestCount adult="2" child="0"/>
            <PerDayRates currency="AUD">
              <PerDayRate stayDate="2012-10-23" baseRate="22.5"/>
              <PerDayRate stayDate="2012-10-24" baseRate="22.5"/>
            </PerDayRates>
            <TotalAmount amountAfterTaxes="100.0" amountOfTaxes="10.0" currency="AUD"/>
            <PaymentCard cardHolderName="Test Booking" cardNumber="5555555555554444"
cardCode="CA" seriesCode="925" expireDate="122017"/>
          </RoomStay>
          <Guest>
            <PrimaryGuest givenName="Firstname firstname" lastName="lastname last name"/>
          </Guest>
        </Booking>
        <Booking id="58652754" type="Cancel" createDateTime="2012-10-19T10:49:39-07:00"
source="HW">
          <Hotel id="34323"/>
          <RoomStay roomTypeID="STANDARD" ratePlanID="XHW" numberOfRooms="1">
            <StayDate arrivalDate="2012-10-23" departureDate="2012-10-24"/>
            <GuestCount adult="2" child="0"/>
            <PerDayRates currency="AUD">
              <PerDayRate stayDate="2012-10-23" baseRate="33.3"/>
            </PerDayRates>
            <TotalAmount amountAfterTaxes="37.0" amountOfTaxes="3.7" currency="AUD"/>
            <PaymentCard cardHolderName="test booking" cardNumber="5555555555554444"
cardCode="CA" seriesCode="925" expireDate="122017"/>
          </RoomStay>
```

```
            <Guest>
              <PrimaryGuest givenName="Firstname firstname" middleName="m m m"
lastName="lastname last name"/>
            </Guest>
          </Booking>
        </BookingRetrievalRS>
      </ns2:retrieveBookingResponse>
    </soap:Body>
</soap:Envelope>
```