

RAZOR WEB SERVICES

The Razor property management system offers a comprehensive set of web services to provide business partners with up to date property information, pricing and availability, and to enable on-line reservations to be made. The diagram illustrates various scenarios by which this is achieved.

SOAP Services

Razor has SOAP (Simple Object Access Protocol) web services with documents that conform to OTA (Open Travel Alliance) XML schema. SOAP/OTA may be used to communicate with legacy property management systems and central reservations systems. The only benefit of this approach is standardization.

REST Services

Razor also has REST (Representational State Transfer) web services for both OTA and a set of simple XML documents that directly reflect Razor information. A simple approach is recommended to link Razor to dynamic web servers which typically have their own databases. Web servers that integrate data from multiple sources may require this approach.

Razor Applications

Razor client applications communicate directly with the server via fast and efficient remote procedure calls. The applications can be inserted into a web page either as a page division or as a button. Clicking the button opens the application page in situ – the page has a close button to restore the button view. This option might be used by property managers and agencies to embed the application into their Internet and/or intranet web sites.

Widget Services

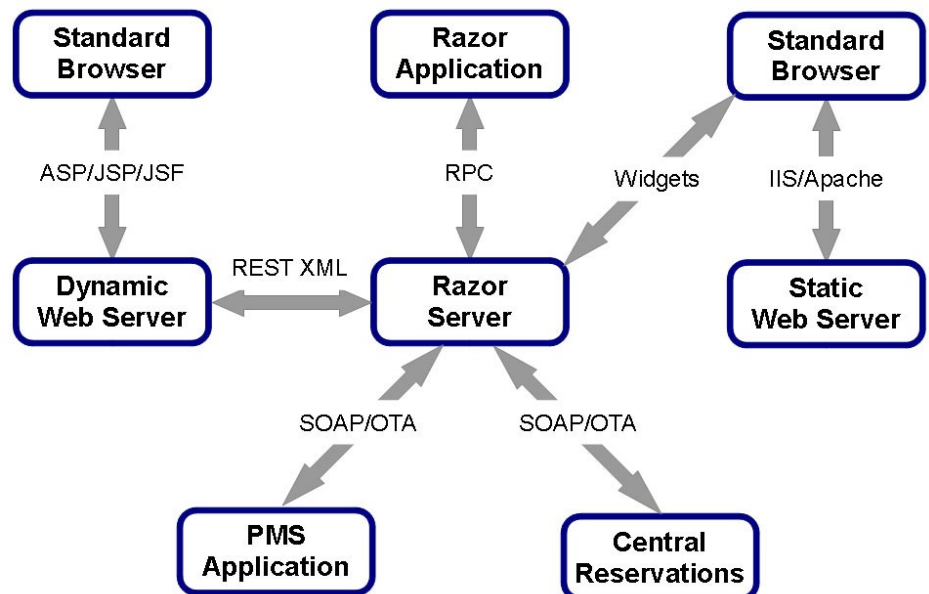
The most simple option, however, is to use Razor widgets. A widget is a modular software component that exposes dynamic content in static web pages. Widgets are embedded in a standard HTML web page using a few lines of code to configure them – a task easily performed without use of ASP, JSP, JSF, plug-ins or other technologies. Each widget has direct and secure access to the appropriate Razor service and updates itself to reflect parameters in the host web page. For simplicity, use embedded widgets and applications, for flexibility use REST services, and for legacy integration use SOAP services.

Access Identification and Control

A unique web service code is allocated to each business partner, which is used to set an auto-login parameter in the host web page. This identifies the partner to the system and enables reservations, orders and commissions to be allocated accordingly. Once installed and configured widgets require no maintenance or administration – software upgrades are installed with no user intervention – and data is refreshed automatically.

Razor applications are more specialized but are also very simple to embed in existing web pages. Applications are typically used by business partners which require integrated functions. Simple applications group related functions into a tabbed panels. Complex applications group many more functions into tabbed panels according to the roles and permissions that the user has been allocated.

REST services should be used where there is no corporate or legacy imperative to use SOAP services. The decision



between the simple XML document formats and the Open Travel Alliance (OTA) XML schema depends on one's attitude to simplicity versus standardization. Initiatives by the OTA, European Committee for Standardization (CEN), Hotel Technology - Next Generation (HTNG), Expedia Quick Connect (EQC) etc indicate that the benefits of such standards will take some time to accrue. Other major players such as Amadeus, TripAdvisor, LastMinute, HomeAway etc do not use OTA but prefer simple schema that meet their specific needs.

The REST services described below are designed to provide property descriptions, images, contacts, locations, rates and availability in the simplest possible XML format. The documents are largely self describing and provide detailed and timely information using simple URLs. There is no need for complex web service setup, and in many cases there is no need to parse the XML response. Razor allows the service requester to specify an XSL file that can transform the XML into HTML or other format without programming. For example, a drop down list of property names or a table showing availability can be generated from the raw XML with suitable XSL files.

REST services may also be used to effect the OTA request/response (RQ/RS) cycle. A URL is provided to create and return an OTA request (RQ) document using parameters specified in the URL. This URL does not perform any operation but is intended to show the request format expected by the SOAP service. However, if one substitutes RS for RQ in the URL, Razor creates a request from the URL parameters, passes it to the appropriate service, and returns the OTA response (RS) document. The REST response is used in exactly the same way as would be a SOAP response.

SOAP services require that the OTA request document is created and sent to the appropriate SOAP port and operation, which returns the matching response document. Examples of the Java source code and libraries to create and marshal request documents and to un-marshal response documents are included. It is recommended that the JAXB class library and object factory provided by Razor be used unless you are committed to another platform. This facilitates deployment via REST and SOAP, and makes best use of modern open source technology such as CXF.

Razor provide these web services as is and without charge in the hope that they will be useful in promoting property reservations via the Razor Property Management applications, widgets and XML services. The use of these services for any other purpose is expressly prohibited. The services and the information contained in this document are provided without liability and are subject to change without notice. Feedback and constructive suggestions and requests are welcome and should be directed to info@razorpms.com.

RAZOR SOAP SERVICES

OpenTravel (OTA) is a not-for-profit trade association, founded in 1999 by travel companies, with a primary focus on the creation of electronic message structures to facilitate communication between the disparate systems in the global travel industry. Members actually do the work of identifying what messages are needed, prioritizing the work and creating the messages. The OpenTravel Implementation Guide provides invaluable information to both analysts and implementers of the OpenTravel specification on how to more easily understand and build software systems that are interoperable with other travel systems using the OpenTravel schema.

The OTA XML message schema are defined in OTA2008B_Publication.zip and are described with use cases in the OTA_MessageUsersGuide2008BV1.0.pdf. Razor SOAP services conform to the recommendations contained in OTA_ImplementationGuide_WSDL_20060410.pdf and use the OTA code lists that are contained in the spreadsheet OTA_CodeTable20081205.xls. These and other documents may be downloaded from the OTA website at <http://www.opentravel.org/>. Advice may be obtained at <http://groups.google.com/group/OTA-Impl-Forum/>.

OTA web services are implemented to respond to requests using request/response (RQ/RS) pairs that conform to the XML schema listed above. While most request documents are designed to request information from a web service, some are used to send information to the service. Such a document typically has the word Notif as part of its name.

Razor implements a set of REST and SOAP web services that consume and produce XML documents that comply with the OTA schema. The documents are those in most common use, but may be extended to meet the specific needs of business partners. The REST services create OTA RQ documents from sets of parameters that are set via simple URLs. The RQ documents are then used to invoke the appropriate service which responds with an RS document to indicate if the request has succeeded, and if not, list the error(s) that caused it to fail.

The sections below explain for each document the format of the REST URL, the parameters required, the errors that may occur, the codes used, and an example URL with the resultant OTA XML document. An archive containing the classes and object factory to marshal and unmarshal these and other OTA2008B XML documents using the Java Architecture for XML Binding (JAXB) is available at <http://razorpms.com/service/opentravel-2008b-jaxb.jar> and is available for use under the Apache license. Similar XML binding libraries are available from other sources for XMLBeans and .NET3.5.

Razor supports multiple property managers and each URL has an optional query to define the id (list) of the property manager(s) to which the message is to be sent. The default property manager is used if there is no query. Multiple property managers are indicated by a comma delimited list of their ids. Each request also has an encoded point of sale code (*pos_code*), which is the last parameter of its URL, for authentication. Point of sale codes are allocated by the property manager's system administrator to those authorized to use the web services. The following errors caused by a missing or improper point of sale code will deny access to the requested service:

<i>POS_Missing</i>	Point of sale code is absent, null or blank
<i>POS_Invalid</i>	Point of sale code is invalid
<i>POS_Context</i>	Point of sale context is not valid (must be "RZ")

Razor REST service have URLs, parameters, return codes and codes to create each of these request messages. Each also has a complementary URL to create the request message, invoke the appropriate service, and return a response message. Alternatively, the SOAP service described in the WSDL may be used to invoke the services. The messages currently supported by Razor adapt the OTA schema, which are designed for hotel accommodation, so as to suit vacation rental properties. XML elements and attributes have been mapped to minimize any confusion that may arise as a consequence, but this is far from ideal. A core assumption, for example, is that a vacation rental property is let only to one party at a time, which has an impact on descriptive information, rate plans, availability, and reservations.

<u>Message Types</u>	<u>Description</u>
OTA_CancelRQ/RS	Cancel an existing reservation
OTA_HotelAvailRQ/RS	Respond to a request for available accommodation
OTA_HotelDescriptiveInfoRQ/RS	Respond to a request for descriptive information
OTA_HotelInvCountNotifRQ	Notify another system of available accommodation
OTA_HotelResNotifRQ/RS	Request a new reservation
OTA_HotelSearchRQ/RS	Search for accommodation using criteria
OTA_PingRQ/RS	Test the connection to a service
OTA_ProfileCreateRQ/RS	Notify another system of a customer or company profile
OTA_ReadRQ/OTA_ResRetrieveRS	Respond to a request for an existing reservation

OTA_CancelRQ

This document is to cancel a reservation via a URL that returns an OTA_CancelRQ document created with the specified parameters. It does not perform any operation, but illustrates the document required by the SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/CancelRQ/reservationid/reason?pos=pos_code`

Parameters

<code>reservationid</code>	Identity of reservation to be canceled
<code>reason</code>	Description of reason for the cancellation
<code>pos_code</code>	Point of sale code requesting the cancellation

Example REST URL

`http://razorpms.com/service/xml/ota/CancelRQ/100602/This is a test cancellation.?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_CancelRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="1.010" Target="Production"
  Timestamp="2010-07-20T10:54:02" EchoToken="OTA_CancelRQ 5, 100602, This is a test.,
  8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <UniqueID ID="100602" ID_Context="RZ" Type="14" />
  <Reasons>
    <Reason Language="en">This is a test.</Reason>
  </Reasons>
</OTA_CancelRQ>
```

OTA_CancelRS

This document is to cancel a reservation by creating an OTA_CancelRQ and returning an OTA_CancelRS document to indicate success or list any errors that may have prevented the cancellation.

REST URL

`http://razorpms.com/service/xml/ota/CancelRS/reservationid/reason?pos=pos_code`

Parameters

<code>reservationid</code>	Identity of reservation to be canceled
<code>reason</code>	Description of reason for the cancellation
<code>pos_code</code>	Point of sale code requesting the cancellation

Errors

<code>UniqueID</code>	UniqueID is absent, null or blank (must be <code>reservationid</code>)
<code>UniqueID_Context</code>	UniqueID context is not valid (must be "RZ")
<code>UniqueID_Invalid</code>	UniqueID is not valid (<code>reservationid</code> must be valid)
<code>UniqueID_Type</code>	UniqueID type is not valid (must be "14")
<code>State_Invalid</code>	Reservation state is Cancelled, Arrived, Departed or Archived

Example REST URL

`http://razorpms.com/service/xml/ota/CancelRS/100602/reason/This is a test cancellation.?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_CancelRS xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="2.006" Target="Production"
  Timestamp="2010-07-20T10:54:02" EchoToken="OTA_CancelRQ 5, 100602, This is a test.,
8508a7e6ce43e091">
  <Errors>
    <Error Code="Departed" ShortText="Invalid reservation status"
      Type="394" Language="en" />
  </Errors>
</OTA_CancelRS>
```

OTA_HotelAvailRQ

This document is to request available accommodation via a URL that creates an OTA_HotelAvailRQ document with the specified parameters. It does not perform any operation, but illustrates the request document required by the SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/HotelAvailRQ/propertyid/start/end?pos=pos_code`

Parameters

<code>propertyid</code>	Property(s) for which availability is requested
<code>start</code>	Date from which availability is required in yyyy-MM-dd format
<code>end</code>	Date to which availability is required in yyyy-MM-dd format
<code>pos_code</code>	Point of sale code requesting the information

Example REST URL

`http://razorpms.com/service/xml/ota/HotelAvailRQ/143,144,146,155/2010-12-19/2010-12-25?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  EchoToken="OTA_HotelAvailRQ 5, 143, 144, 146, 155, 2010-12-19, 2010-12-25,
  8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <AvailRequestSegments>
    <AvailRequestSegment>
      <HotelSearchCriteria>
        <Criterion>
          <HotelRef HotelCodeContext="RZ" HotelCode="143" />
          <HotelRef HotelCodeContext="RZ" HotelCode=" 144" />
          <HotelRef HotelCodeContext="RZ" HotelCode=" 146" />
          <HotelRef HotelCodeContext="RZ" HotelCode=" 155" />
          <StayDateRange End="2010-12-25" Start="2010-12-19" />
        </Criterion>
      </HotelSearchCriteria>
    </AvailRequestSegment>
  </AvailRequestSegments>
</OTA_HotelAvailRQ>
```

OTA_HotelAvailRS

This document is to request available accommodation via a URL that creates an OTA_HotelAvailRQ and returns an OTA_HotelAvailRS document to indicate success and list the available accommodation and rates, or list any errors that may have prevented the request being fulfilled.

REST URL

`http://razorpms.com/service/xml/ota/HotelAvailRQ/propertyid/start/end?pos=pos_code`

Parameters

<code>propertyid</code>	Property(s) for which availability is requested
<code>start</code>	Date from which availability is required in yyyy-MM-dd format
<code>end</code>	Date to which availability is required in yyyy-MM-dd format
<code>pos_code</code>	Point of sale code requesting the information

Errors

<code>HotelCode</code>	Property id is absent, null or blank (must be <code>propertyid</code>)
<code>HotelCode_Context</code>	Context is not valid (must be "RZ")
<code>HotelCode_Invalid</code>	Property id is not valid (<code>propertyid</code> must be a valid property)

Example REST URL

`http://razorpms.com/service/xml/ota/HotelAvailRQ/143,144,146,155/2010-12-19/2010-12-25?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="5.001" Target="Production"
  Timestamp="2010-07-20T10:54:04"
  EchoToken="OTA_HotelAvailRQ 5, 143, 144, 146, 155, 2010-12-19, 2010-12-25,
8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Success />
  <RoomStays>
    <RoomStay AvailableIndicator="true" PromotionVendorCode="1"
      WarningRPH="">
      <RoomTypes>
        <RoomType PromotionVendorCode="" BedTypeCode="5" RoomType="2 Bedrooms" />
      </RoomTypes>
      <RoomRates>
        <RoomRate EffectiveDate="2010-12-19" PromotionVendorCode="1"
          RatePlanType="13">
          <Rates>
            <Rate>
              <Base CurrencyCode="ZAR" AmountAfterTax="4500" />
            </Rate>
          </Rates>
        </RoomRate>
      </RoomRates>
      <CancelPenalties CancelPolicyIndicator="true">
        <CancelPenalty>
          <Deadline OffsetUnitMultiplier="30" OffsetTimeUnit="Day" />
          <AmountPercent Percent="100" />
          <PenaltyDescription Language="en"
            CreateDateTime="2010-07-20T10:54:04">
            <Text Formatted="false">Refunds and waiver of cancellation
              policies are at the discretion of management.
              Refunds are dependent on re-letting of the room.
              Guests are advised to take out cancellation insurance.</Text>
          </PenaltyDescription>
        </CancelPenalty>
      </CancelPenalties>
    </RoomStay>
  </RoomStays>
</OTA_HotelAvailRS>
```

```

    </CancelPenalty>
    <CancelPenalty>
      <Deadline OffsetUnitMultiplier="60" OffsetTimeUnit="Day" />
      <AmountPercent Percent="50" />
    </CancelPenalty>
    <CancelPenalty>
      <Deadline OffsetUnitMultiplier="90" OffsetTimeUnit="Day" />
      <AmountPercent Percent="25" />
    </CancelPenalty>
  </CancelPenalties>
  <BasicPropertyInfo AreaID="58777" BrandName="Nox Tourism cc"
    HotelName="La Plantacion" HotelCode="155">
    <Position AltitudeUnitOfMeasureCode="MTR" Altitude="0.0"
      Longitude="18.3817859349754" Latitude="-33.963552777778" />
    <Address>
      <CityName>Camps Bay</CityName>
      <PostalCode>8040</PostalCode>
      <StateProv StateCode="WC">Western Cape</StateProv>
      <CountryName Code="ZA">South Africa</CountryName>
    </Address>
    <ContactNumbers>
      <ContactNumber PhoneNumber="+27 21 424 3353"
        PhoneUseType="3" PhoneTechType="1" />
      <ContactNumber PhoneNumber="+27 21 424 3355"
        PhoneUseType="3" PhoneTechType="3" />
      <ContactNumber PhoneNumber="+27 84 804 9902"
        PhoneUseType="5" PhoneTechType="5" />
    </ContactNumbers>
  </BasicPropertyInfo>
</RoomStay>
</RoomStays>
</OTA_HotelAvailRS>

```

OTA_HotelDescriptiveInfoRQ

This document is to request detailed property information via a URL that creates an OTA_HotelDescriptiveInfoRQ document with the specified parameters. It does not perform any operation, but illustrates the request document required by the SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/HotelDescriptiveInfoRQ/propertyid?pos=pos_code`

Parameters

<code>propertyid</code>	Property(s) for which information is requested
<code>pos_code</code>	Point of sale code requesting the information

Example REST URL

`http://razorpms.com/service/xml/ota/HotelAvailRQ/143?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelDescriptiveInfoRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="1.009" Target="Production"
  Timestamp="2010-07-20T10:54:04" EchoToken="OTA_HotelDescriptiveInfoRQ 5, 143,
8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <HotelDescriptiveInfos>
    <HotelDescriptiveInfo BrandCodeList=""
      CountryCodeList="" StateCodeList="" HotelCodeContext="RZ" HotelCode="143"
      ChainCode="5" />
  </HotelDescriptiveInfos>
</OTA_HotelDescriptiveInfoRQ>
```

OTA_HotelDescriptiveInfoRS

This document is to request detailed property information via a URL that creates an OTA_HotelDescriptiveInfoRQ and returns an OTA_HotelDescriptiveInfoRS document with the requested information. It also indicates success, or lists any errors that may have prevented the request being fulfilled. Much of the information is encoded using OTA code lists, the meaning of which are described in the OTA_CodeTable20081205.xls spreadsheet. The codes actually used by Razor are listed by the URL <http://razorpms.com/service/ota/list/XXX> where XXX is the three character OTA list identifier.

REST URL

http://razorpms.com/service/xml/ota/HotelDescriptiveInfoRS/propertyid?pos=pos_code

Parameters

propertyid Property(s) for which information is requested
pos_code Point of sale code requesting the information

Errors

HotelCode Property id is absent, null or blank (must be *propertyid*)
HotelCode_Context Context is not valid (must be "RZ")
HotelCode_Invalid Property id is not valid (*propertyid* must be a valid property)

OTA Codes:

OTA Code Categories	OTA code groups at http://razorpms.com/service/xml/ota/list
AccessibleCode	OTA PHY list at http://razorpms.com/service/xml/ota/list/PHY
AddressType	OTA AUT list at http://razorpms.com/service/xml/ota/list/AUT
ArchitectureStyle	OTA ARC list at http://razorpms.com/service/xml/ota/list/ARC
BusinessServiceCode	OTA BUS list at http://razorpms.com/service/xml/ota/list/BUS
CardCode	Card code list at http://razorpms.com/service/xml/ota/list/X17
CardType	OTA CDT list at http://razorpms.com/service/xml/ota/list/CDT
EmailType	OTA EAT list at http://razorpms.com/service/xml/ota/list/EAT
GuestRoomInfo	OTA GRI list at http://razorpms.com/service/xml/ota/list/GRI
HotelCategory	OTA ACC list at http://razorpms.com/service/xml/ota/list/ACC
HotelInfoCode	OTA HIC list at http://razorpms.com/service/xml/ota/list/HIC
LocationCategory	OTA LOC list at http://razorpms.com/service/xml/ota/list/LOC
MealPlan	OTA MPT list at http://razorpms.com/service/xml/ota/list/MPT
MeetingRoomCode	OTA MRC list at http://razorpms.com/service/xml/ota/list/MRC
PetsAllowedCode	OTA PET list at http://razorpms.com/service/xml/ota/list/PET
PhoneLocationType	OTA PLT list at http://razorpms.com/service/xml/ota/list/PLT
PhoneTechType	OTA PTT list at http://razorpms.com/service/xml/ota/list/PTT
PhoneUseType	OTA PUT list at http://razorpms.com/service/xml/ota/list/PUT
ProximityCode	OTA LOC list at http://razorpms.com/service/xml/ota/list/LOC
RoomAmenityCode	OTA RMA list at http://razorpms.com/service/xml/ota/list/RMA
RoomLocation	OTA RLT list at http://razorpms.com/service/xml/ota/list/RLT
SecurityFeature	OTA SEC list at http://razorpms.com/service/xml/ota/list/SEC
SegmentCategory	OTA SEG list at http://razorpms.com/service/xml/ota/list/SEG

Example REST URL

<http://razorpms.com/service/xml/ota/HotelDescriptiveInfoRS/143?pos=8508a7e6ce43e091>

Example OTA Document

```
<OTA_HotelDescriptiveInfoRS xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNbr="1"
  TransactionIdentifier="1" Version="6.004" Target="Production"
  Timestamp="2010-07-24T09:01:20" EchoToken="OTA_HotelDescriptiveInfoRQ 5, 143,
8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Success />
  <HotelDescriptiveContents AreaID="58777"
    BrandName="Tourinfo cc" ChainName="Nox Tourism cc" HotelCodeContext="RZ"
    HotelName="CBT Lodge" HotelCityCode="CPT" HotelCode="143" BrandCode="890"
    ChainCode="5" Overwrite="true">
    <HotelDescriptiveContent HotelCityCode="CPT"
      Overwrite="true" CurrencyCode="ZAR" LanguageCode="en">
```

```

<HotelInfo PMSSystem="RZ">
  <HotelName HotelShortName="CBT Lodge">CBT Lodge</HotelName>
  <CategoryCodes>
    <LocationCategory Code="2" />
    <SegmentCategory Code="7" />
    <HotelCategory Code="4" />
    <HotelCategory Code="72" />
    <HotelCategory Code="88" />
    <ArchitecturalStyle Code="10" />
    <ArchitecturalStyle Code="4" />
    <GuestRoomInfo Code="4" />
    <GuestRoomInfo Code="72" />
    <GuestRoomInfo Code="88" />
  </CategoryCodes>
  <HotelInfoCodes>
    <HotelInfoCode Code="4" />
  </HotelInfoCodes>
  <Position AltitudeUnitOfMeasureCode="3" Altitude="0.0"
    Longitude="18.3772945404053" Latitude="-33.9571902537236" />
  <!-- Google Maps LatLng format, altitude unit metres -->
  <Services>
    <Service BusinessServiceCode="4" />
    <Service BusinessServiceCode="72" />
    <Service BusinessServiceCode="88" />
    <Service MeetingRoomCode="110" />
    <Service MeetingRoomCode="124" />
    <Service MeetingRoomCode="129" />
    <Service MeetingRoomCode="130" />
    <Service MeetingRoomCode="150" />
    <Service MeetingRoomCode="60" />
    <Service MeetingRoomCode="67" />
    <Service MeetingRoomCode="68" />
    <Service ProximityCode="2" />
  </Services>
</HotelInfo>
<FacilityInfo>
  <GuestRooms MaxOccupancy="6">
    <GuestRoom MaxOccupancy="6" RoomTypeName="3 Bedrooms">
      <Amenities>
        <Amenity RoomAmenityCode="4" />
        <Amenity RoomAmenityCode="72" />
        <Amenity RoomAmenityCode="88" />
      </Amenities>
      <Features>
        <Feature AccessibleCode="1" />
        <Feature AccessibleCode="104" />
        <Feature AccessibleCode="2" />
        <Feature AccessibleCode="53" />
        <Feature AccessibleCode="68" />
        <Feature ProximityCode="3" />
        <Feature AccessibleCode="4" />
        <Feature AccessibleCode="72" />
        <Feature AccessibleCode="88" />
      </Features>
    </GuestRoom>
  </GuestRooms>
</FacilityInfo>
<Policies>
  <Policy Removal="false" LastUpdated="2010-07-24T09:01:22">
    <CancelPolicy CancelPolicyIndicator="true">
      <CancelPenalty>
        <Deadline OffsetUnitMultiplier="30" OffsetTimeUnit="Day" />
        <AmountPercent Percent="100" />
        <PenaltyDescription Language="en"
          CreateDateTime="2010-07-24T09:01:22">
          <Text Formatted="false">Refunds and waiver of
cancellation policies are at the discretion of management. Refunds are dependent on re-

```

```

letting of the room.  Guests are advised to take out cancellation insurance.</Text>
    </PenaltyDescription>
  </CancelPenalty>
  <CancelPenalty>
    <Deadline OffsetUnitMultiplier="60" OffsetTimeUnit="Day" />
    <AmountPercent Percent="50" />
  </CancelPenalty>
  <CancelPenalty>
    <Deadline OffsetUnitMultiplier="90" OffsetTimeUnit="Day" />
    <AmountPercent Percent="25" />
  </CancelPenalty>
</CancelPolicy>
</Policy>
<Policy Removal="false" LastUpdated="2010-07-24T09:01:22">
  <GuaranteePaymentPolicy>
    <GuaranteePayment NoCardHolderInfoReqInd="false"
      Type="31">
      <AcceptedPayments>
        <AcceptedPayment>
<Cash CashIndicator="true" />
        </AcceptedPayment>
        <AcceptedPayment>
<Cash CashIndicator="true" />
        </AcceptedPayment>
        <AcceptedPayment>
<PaymentCard CardCode="VI" CardType="5" />
        </AcceptedPayment>
        <AcceptedPayment>
<PaymentCard CardCode="VI" CardType="6" />
        </AcceptedPayment>
      </AcceptedPayments>
    </GuaranteePayment>
  </GuaranteePaymentPolicy>
</Policy>
<Policy Removal="false" LastUpdated="2010-07-24T09:01:22">
  <PetsPolicies PetsAllowedCode="8" />
</Policy>
<Policy Removal="false" LastUpdated="2010-07-24T09:01:22">
  <PolicyInfo InternetGuaranteeRequiredInd="false"
    MaxChildAge="0" KidsStayFree="false" TotalGuestCount="0"
    UsualStayFreeChildPerAdult="0" UsualStayFreeCutoffAge="0"
    MinRecommendedGuestAge="0" MinGuestAge="0" />
</Policy>
<Policy Removal="false" LastUpdated="2010-07-24T09:01:22">
  <StayRequirements>
    <StayRequirement StayContext="RZ" MaxLOS="30"
      MinLOS="1" />
  </StayRequirements>
</Policy>
</Policies>
<MultimediaDescriptions>
  <MultimediaDescription>
    <TextItems>
      <TextItem CopyrightOwner="Tourinfo cc"
        CopyrightNotice="License:
http://razorpms.com/service/License.html
        Title="Property Notes">
      <Description Language="EN">Enjoy the old world charm of this
lovingly refurbished colonial home! The Lodge is a lovely three-bedroom apartment, ideal
for a group of friends or a family. The three-bedroom Lodge Apartment evokes days gone by
with its colonial ambience and comfort and has spectacular views of the Twelve Apostles
and the Atlantic Ocean. Built over a century ago as a beach front retreat, the Lodge has
been lovingly restored to its former state - rich brown mahogany sash windows and
shutters, golden Oregon pine floors, rustic red clay tiles - all adding up to a
delightful place in which to relax and enjoy Cape hospitality. Furnished with lovingly
collected antique instruments and Africana the Lodge has a very special magic. The
stunning lounge is in colonial style with leather and rich fabric furniture and fittings.

```

Plate glass sash windows look down on the exclusive Bakoven beach - a sheltered haven for Cape Town's most selective sun seekers. The two comfortable wing chairs with Ottoman are the ideal spot for soaking up the afternoon sun, and the large leather sofa is ideal for curling up in front of the satellite TV or sitting with a book. Old-fashioned double doors open onto a paved deck with a solar-heated swimming pool and barbecue. The sun deck has eight sun loungers, and it's the perfect place to watch the sun set into the Atlantic, with a chilled glass of wine in your hand. The three bedrooms are accessed from the lounge area and have views of mountain and sea, each with it's own en-suite bathroom. Two rooms have queen-sized beds, and one can be made up as either two single beds of one king-size bed. If you'd prefer to eat out, the Lodge Apartment is a short stroll from the Camps Bay beach and promenade, with it's array of restaurants and bars. The apartment is fully serviced every day except Sundays and Public Holidays. We will also do everything we can to ensure that you have as relaxing a holiday as possible, by helping where we can; anything from baby-sitter bookings through to restaurant recommendations. Come and put your feet up in new Camps Bay's most elegant old home. We look forward to welcoming you!

</TextItem>

<TextItem CopyrightOwner="Tourinfo cc"
CopyrightNotice="License:

<http://razorpms.com/service/License.html>

Title="Included Items">

<Description Language="EN">Daily Servicing: Monday

to Saturday except Public Holidays. Concierge Service. Compact Bouquet DSTV. All taxes. All Linen and Towels.</Description>

</TextItem>

<TextItem CopyrightOwner="Tourinfo cc"
CopyrightNotice="License:

<http://razorpms.com/service/License.html>

Title="Optional Items">

<Description Language="EN">Laundry on request at

additional expense. Baby sitting on request at additional expense. Catering on request at additional expense. Transfers/Shuttles at additional expense. Grocery stocking at additional expense. DVD Collection on request. Breakfast on request at additional expense. Dinner Reservations</Description>

</TextItem>

<TextItem CopyrightOwner="Tourinfo cc"
CopyrightNotice="License:

<http://razorpms.com/service/License.html>

Title="Manager Notes">

<Description Language="EN">Nox Tourism was founded in

September 2003 by Richard Marshall. We are market leaders on Cape Town's Atlantic Seaboard, and continue to innovate in terms of our service to both the travel trade and to the guests that stay in our villas and apartments. All of our properties are in the four or five star market, and options range from one bedroom apartments to seven bedroom villas, from penthouse apartments to beach cottages. However different they may be, each property has what we call tourism potential and has something unique and exceptional about it, whether that be views, location, luxury or a combination of these and many other factors. We have an obsessive focus on the guest experience, and every guest's needs are catered to by our dedicated team of guest relations consultants, property managers and housekeepers.

</Description>

</TextItem>

</TextItems>

</MultimediaDescription>

<MultimediaDescription>

<ImageItems>

<ImageItem Category="16">

<ImageFormat CopyrightOwner="Tourinfo cc"
CopyrightNotice="License:

<http://razorpms.com/service/License.html>

Title="Thumbnail Image" FileName="Product143-000Thumb.jpg">

<URL>

<http://razorpms.com/pictures/Product143-000Thumb.jpg>

</URL>

</ImageFormat>

<Description />

```

                </ImageItem>
..... for all images
            </ImageItems>
        </MultimediaDescription>
    </MultimediaDescriptions>
    <ContactInfos>
        <ContactInfo Removal="false" ContactProfileType="12"
            ContactProfileID="5">
            <Addresses>
                <Address Type="6">
                    <PostalCode>8040</PostalCode>
                </Address>
            </Addresses>
            <Phones>
                <Phone PhoneNumber="+27 21 424 3353"
                    PhoneUseType="3" PhoneTechType="1" PhoneLocationType="9" />
                <Phone PhoneNumber="+27 21 424 3355"
                    PhoneUseType="3" PhoneTechType="3" PhoneLocationType="9" />
                <Phone PhoneNumber="+27 84 804 9902"
                    PhoneUseType="5" PhoneTechType="5" PhoneLocationType="9" />
            </Phones>
            <Emails>
                <Email DefaultInd="true" EmailType="8">info@noxrentals.co.za
            </Email>
            </Emails>
            <URLs>
                <URL Type="1">www.noxrentals.co.za</URL>
            </URLs>
        </ContactInfo>
    </ContactInfos>
    </HotelDescriptiveContent>
</HotelDescriptiveContents>
</OTA_HotelDescriptiveInfoRS>

```

OTA_HotelInvCountNotifRQ

This document is to notify available accommodation via a URL that returns an OTA_HotelInvCountNotifRQ document created with the specified parameters. It notifies another system of available inventory, typically via a SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/HotelInvCountNotifRQ/propertyid/start/end?
pos=pos_code`

Parameters

<code>propertyid</code>	Property(s) for which availability is requested
<code>start</code>	Date from which availability is required in yyyy-MM-dd format
<code>end</code>	Date to which availability is required in yyyy-MM-dd format
<code>pos_code</code>	Point of sale code requesting the information

Example REST URL

`http://razorpms.com/service/xml/ota/HotelInvCountNotifRQ/155/2010-07-19/2010-07-25?
pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelInvCountNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="1.009" Target="Production"
  Timestamp="2010-07-20T10:54:05" EchoToken="OTA_HotelInvCountNotifRQ 5, 155, 2010-
07-19, 2010-07-25, 8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Inventories HotelCodeContext="RZ" HotelCode="155"
  ChainCode="5">
    <Inventory>
      <StatusApplicationControl InvTypeCode="2 Bedrooms"
      End="2010-07-25" Start="2010-07-19" PromotionVendorCode="1" />
      <InvCounts>
        <InvCount Count="1" CountType="1" />
      </InvCounts>
    </Inventory>
  </Inventories>
</OTA_HotelInvCountNotifRQ>
```

OTA_HotelResNotifRQ

This document is to reserve accommodation via a URL that creates an OTA_HotelResNotifRQ document with the specified parameters. It does not perform any operation, but illustrates the request document required by the SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/HotelResNotifRQ/propertyid/start/end/email/name?pos=pos_code`

Parameters

<code>propertyid</code>	Property(s) for which a reservation is requested
<code>start</code>	Date of arrival in yyyy-MM-dd format
<code>end</code>	Date of departure in yyyy-MM-dd format
<code>email</code>	Email address of the guest which must be in a valid format
<code>name</code>	Name of the guest in <i>surname,givename</i> format
<code>pos_code</code>	Point of sale code requesting the information

Example REST URL

`http://razorpms.com/service/xml/ota/HotelResNotifRQ/155/2010-07-19/2010-07-25/chris@razor.com/ Mars,Christopher?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelResNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="4.001" Target="Production"
  Timestamp="2010-07-20T10:54:05"
  EchoToken="OTA_HotelResNotifRQ 5, 155, 2010-07-19, 2010-07-25, chris@razor.com,
  Mars, Christopher, 8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <HotelReservations>
    <HotelReservation>
      <RoomStays>
        <RoomStay PromotionVendorCode="" WarningRPH="">
          <TimeSpan End="2010-07-25" Start="2010-07-19" />
          <BasicPropertyInfo HotelCodeContext="RZ"
            HotelCode="155" ChainCode="5" />
        </RoomStay>
      </RoomStays>
      <ResGuests>
        <ResGuest>
          <Profiles>
            <ProfileInfo>
              <Profile ProfileType="1" StatusCode="">
                <Customer>
                  <PersonName>
                    <GivenName> Christopher</GivenName>
                    <Surname>Mars</Surname>
                  </PersonName>
                  <Email EmailType="1">chris@razor.com</Email>
                </Customer>
              </Profile>
            </ProfileInfo>
          </Profiles>
        </ResGuest>
      </ResGuests>
    </HotelReservation>
  </HotelReservations>
</OTA_HotelResNotifRQ>
```

OTA_HotelResNotifRS

This document is to reserve accommodation via a URL that creates an OTA_HotelResNotifRQ with the specified parameters and returns an OTA_HotelResNotifRS document. It requests Razor to accept a reservation for the specified property(s), dates and guest. The guest email address should be valid as it is used to request feedback after the visit. If a record of a person having the specified email address exists, Razor assumes that the guest is that person. If successful, the response document details RACK and STO rates and currency, cancellation policies, and reservation confirmation code (*ResID_Value*). If not, it lists the error(s) that caused it to fail.

REST URL

`http://razorpms.com/service/xml/ota/HotelResNotifRS/propertyid/start/end/email/name?pos=pos_code`

Parameters

<i>propertyid</i>	Property(list) for which a reservation is requested
<i>start</i>	Date of arrival in yyyy-MM-dd format
<i>end</i>	Date of departure in yyyy-MM-dd format
<i>email</i>	Email address of the guest which must be in a valid format
<i>name</i>	Name of the guest in <i>surname,givename</i> format
<i>pos_code</i>	Point of sale code requesting the information

Errors

<i>ChainCode</i>	Property manager id is absent, null or blank
<i>ChainCode_Invalid</i>	Property manager id is not valid
<i>CreditCard</i>	The credit card number is absent, null or blank
<i>CreditCard_Invalid</i>	The credit card number is not in a valid format
<i>Customer</i>	Customer is absent, null or blank (must be <i>name</i>)
<i>Customer_Surname</i>	Customer surname is absent, null or blank (must be <i>surname</i>)
<i>Emailaddress</i>	Email address is absent, null or blank (must be <i>email</i>)
<i>Emailaddress_Invalid</i>	Email address is not in a properly formed format
<i>EndDate</i>	End (departure) date is absent, null or not in yyyy-MM-dd format
<i>Guarantee_Invalid</i>	Requestor is not an agent and there are no credit card details
<i>HotelCode</i>	Property id is absent, null or blank (must be <i>propertyid</i>)
<i>HotelCode_Contxt</i>	Context is not valid (must be "RZ")
<i>HotelCode_Invalid</i>	Property id is not valid (<i>propertyid</i> must be a valid property)
<i>Reservation_Collides</i>	Reservation collides with another and could not be created
<i>Reservation_Invalid</i>	Reservation could not be created
<i>StartDate</i>	Start (arrival) date is absent, null or not in yyyy-MM-dd format

Example REST URL

`http://razorpms.com/service/xml/ota/HotelResNotifRS/155/2010-07-19/2010-07-25/chris@razor.com/ Mars,Chris?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelResNotifRS xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="4.001" Target="Production"
  Timestamp="2010-07-25T14:52:47"
  EchoToken="OTA_HotelResNotifRQ 5, 144, 2010-09-19, 2010-09-29, chris@razor.com,
Mars, Christopher, 8508a7e6ce43e091">
  <Success />
  <HotelReservations>
    <HotelReservation>
      <RoomStays>
        <RoomStay PromotionVendorCode="1" WarningRPH="1">
          <RoomRates>
            <RoomRate PromotionVendorCode=""1">
              <Rates>
                <Rate ExpireDate="2010-09-29" EffectiveDate="2010-09-19">
                  <Base CurrencyCode="ZAR" AmountAfterTax="16000.0" />
                  <RateDescription Language="en" CreateDateTime="2010-
07-25T14:52:47">
                    <Text Formatted="false">Total RACK per
stay</Text>
                  </RateDescription>
```

```

        </Rate>
        <Rate ExpireDate="2010-09-29" EffectiveDate="2010-09-19">
          <Base CurrencyCode="ZAR" AmountAfterTax="12000.0" />
          <RateDescription Language="en" CreateDateTime="2010-
07-25T14:52:47">
            <Text Formatted="false">Total STO per stay</Text>
          </RateDescription>
        </Rate>
      </Rates>
    </RoomRate>
  </RoomRates>
  <TimeSpan End="2010-09-29" Start="2010-09-19" />
  <BasicPropertyInfo HotelCodeContext="RZ"
    HotelCode="144" ChainCode="5" />
  <BookingRules>
    <BookingRule>
      <CancelPenalties CancelPolicyIndicator="true">
        <CancelPenalty>
          <Deadline OffsetUnitMultiplier="30"
OffsetTimeUnit="Day" />
          <AmountPercent Percent="100" />
          <PenaltyDescription Language="en"
            CreateDateTime="2010-07-25T14:52:47">
            <Text Formatted="false">Refunds and waiver of
cancellation policies are at the discretion of management. Refunds are dependent on re-
letting of the room. Guests are advised to take out cancellation insurance.</Text>
          </PenaltyDescription>
        </CancelPenalty>
        <CancelPenalty>
          <Deadline OffsetUnitMultiplier="60"
OffsetTimeUnit="Day" />
          <AmountPercent Percent="50" />
        </CancelPenalty>
        <CancelPenalty>
          <Deadline OffsetUnitMultiplier="90"
OffsetTimeUnit="Day" />
          <AmountPercent Percent="25" />
        </CancelPenalty>
      </CancelPenalties>
      <RequiredPaymts>
        <GuaranteePayment NonRefundableIndicator="true"
          NoCardHolderInfoReqInd="true" Type="31">
          <AcceptedPayments>
            <AcceptedPayment>
              <Cash CashIndicator="true" />
            </AcceptedPayment>
            <AcceptedPayment>
              <PaymentCard CardCode="VI" CardType="5" />
            </AcceptedPayment>
            <AcceptedPayment Remark="Deposit" />
          </AcceptedPayments>
        </GuaranteePayment>
        <GuaranteePayment NonRefundableIndicator="true"
          Type="8">
          <AmountPercent CurrencyCode="ZAR" Percent="50.0"
            NmbrOfNights="10" FeesInclusive="true"
TaxInclusive="true" />
          <Deadline AbsoluteDeadline="2010-07-25" />
        </GuaranteePayment>
      </RequiredPaymts>
    </BookingRule>
  </BookingRules>
</RoomStay>
</RoomStays>
<ResGlobalInfo>
  <HotelReservationIDs>
    <HotelReservationID ResID_SourceContext="RZ"

```

```
                ResID_Source="5" ResID_Value="102974" ResID_Type="14" />
            </HotelReservationIDs>
        </ResGlobalInfo>
    </HotelReservation>
</HotelReservations>
<POS>
    <Source>
        <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
</POS>
</OTA_HotelResNotifRS>
```

OTA_HotelSearchRQ

This document is to search for accommodation via a URL that returns an OTA_HotelSearchRQ document with the specified parameters. It does not perform any operation, but illustrates the request documents required by the SOAP service. A search criterion is specified by a key value pair in the format *key=value*. Multiple criteria are specified in a comma delimited list of these key value pairs. Criteria are added to refine the search and to shrink the response list. Criteria that are OTA list codes may substitute the criterion key with the OTA list name (eg: *PHY* may be used in place of *Accessibility*). Only properties that meet all criteria are included in the response, with the following exceptions:

AreaID	Includes properties in any of the specified locations
ArrivalDate	Includes only properties available until the <i>DepartureDate</i>
BedType	Includes properties with any of the specified bed types
CardCode	Includes properties that accept any of the listed credit cards
Latitude/Longitude	Includes properties within <i>Radius</i> kilometers of the position
PaymentType	Includes properties that accept any of the listed payment types

REST URL

http://razorpms.com/service/xml/ota/HotelSearchRQ/criteria?pos=pos_code

Parameters

<i>criteria</i>	One or more criteria by which properties are to be selected
<i>pos_code</i>	Point of sale code requesting the information

Criteria

<i>Accessibility</i>	Physical accessibility of the property from OTA PHY code list
<i>Adults</i>	Number of adults to be accommodated
<i>Amenity</i>	Amenities at or near the property from OTA HAC code list
<i>ArchitectureStyle</i>	Architectural style of the property from OTA ARC code list
<i>AreaID</i>	Identity of the location of the property
<i>ArrivalDate</i>	Date of arrival
<i>BedType</i>	Bed type required from OTA BED code list
<i>BrandCode</i>	Identity of the owner of the property
<i>BrandName</i>	Name of the owner of the property
<i>CardCode</i>	Credit cards accepted from CARD_CODE list
<i>Category</i>	Quality category of the property from OTA SEG code list
<i>ChainCode</i>	Identity of the manager of the property
<i>ChainName</i>	Name of the manager of the property
<i>Children</i>	Number of children to be accommodated
<i>CityCode</i>	Three character ISO code of the property location
<i>Currency</i>	Currency in which rate range is specified
<i>DepartureDate</i>	Date of departure
<i>Duration</i>	Duration of stay in days, as alternative to <i>DepartureDate</i>
<i>HotelCode</i>	Identity of the property
<i>HotelName</i>	Name of the property
<i>Infants</i>	Number of infants to be accommodated
<i>Latitude</i>	Latitude of the property in Google Maps format
<i>LocationCategory</i>	Location category of the property from OTA LOC code list
<i>Longitude</i>	Longitude of the property in Google Maps format
<i>MaxRate</i>	Maximum rate per day of the property (now if no <i>ArrivalDate</i>)
<i>MaxResponses</i>	Maximum number of properties to be included in the response
<i>MeetingFacility</i>	Size of meeting facility required at the property
<i>MinRate</i>	Minimum rate per day of the property (now if no <i>ArrivalDate</i>)
<i>PaymentType</i>	Accepted payment types from OTA PMT code list
<i>PropertyClass</i>	Classification of the property from OTA PCT code list
<i>Radius</i>	Maximum distance in kilometers from the specified position
<i>Recreation</i>	Recreation at or near the property from OTA REC code list
<i>RoomAmenity</i>	Amenities at or near the property from OTA RMA code list
<i>RoomLocation</i>	Location of the property rooms from OTA RLT code list
<i>Security</i>	Security of the property from OTA SEC code list
<i>View</i>	View from the property from OTA RVT code list

Results

<i>RateRange</i>	<i>FixedRate</i> is the quoted rate <i>MaxRate</i> is the rack rate <i>MinRate</i> is the Standard Travel Operator (STO) rate
------------------	-------------------------------------------------------------------------------------------------------------------------------------

This example returns the two properties specified by the identities 143 and 144.

Example REST URL

http://razorpms.com/service/xml/ota/HotelSearchRQ/HotelCode=143,HotelCode=144?
pos=8508a7e6ce43e091

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="2.004" Target="Production"
  TimeStamp="2010-07-21T15:21:12" EchoToken="OTA_HotelSearchRQ 5,
  HotelCode=143,HotelCode=144, 8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Criteria>
    <Criterion>
      <HotelRef HotelCodeContext="RZ" HotelCode="143" />
    </Criterion>
    <Criterion>
      <HotelRef HotelCodeContext="RZ" HotelCode="144" />
    </Criterion>
  </Criteria>
</OTA_HotelSearchRQ>
```

This example returns properties that accept MC (Master Card) credit cards.

Example REST URL

http://razorpms.com/service/xml/ota/HotelSearchRQ/CardCode=MC?pos=8508a7e6ce43e091

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="2.004" Target="Production"
  TimeStamp="2010-07-23T13:14:29" EchoToken="OTA_HotelSearchRQ 5, CardCode=MC,
  8508a7e6ce43e091">
  <POS><Source><RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" /></Source></POS>
  <Criteria>
    <Criterion>
      <AcceptedPayments>
        <AcceptedPayment>
          <PaymentCard CardCode="MC" CardType="1" />
        </AcceptedPayment>
      </AcceptedPayments>
    </Criterion>
  </Criteria>
</OTA_HotelSearchRQ>
```

This example returns properties that have HotelAmenity or HAC code of 115 (220 AC).

Example REST URL

<http://razorpms.com/service/xml/ota/HotelSearchRQ/HAC=115?pos=8508a7e6ce43e091>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="2.004" Target="Production"
  TimeStamp="2010-07-22T13:28:06" EchoToken="OTA_HotelSearchRQ 5, HAC=115,
8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Criteria>
    <Criterion>
      <HotelAmenity Code="115" />
    </Criterion>
  </Criteria>
</OTA_HotelSearchRQ>
```

This example returns a maximum for ten properties that are within three kilometers of S33.95°, E18.37°.

Example REST URL

<http://razorpms.com/service/xml/ota/HotelSearchRQ/Latitude=33.95,Longitude=18.37,Radius=3,MaxResponses=10?pos=8508a7e6ce43e091>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="2.004" Target="Production"
  TimeStamp="2010-07-21T21:41:52"
  EchoToken="OTA_HotelSearchRQ 5, Latitude=-33.95,Longitude=18.37,Radius=3,
MaxResponses=10, 8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Criteria>
    <Criterion>
      <Position Latitude="-33.95" />
    </Criterion>
    <Criterion>
      <Position Longitude="18.37" />
    </Criterion>
    <Criterion>
      <Radius UnitOfMeasureCode="2" Distance="1" />
    </Criterion>
  </Criteria>
</OTA_HotelSearchRQ>
```

OTA_HotelSearchRS

This document is to search for accommodation via a URL that creates an OTA_HotelSearchRQ and returns an OTA_HotelSearchRS document to indicate success and detail the properties that meet the specified criteria, or list any errors that may have prevented the request being fulfilled.

REST URL

http://razorpms.com/service/xml/ota/HotelSearchRS/criteria?pos=pos_code

Parameters

<code>criteria</code>	One or more criteria by which properties are to be selected
<code>pos_code</code>	Point of sale code requesting the information

Errors

<code>Criteria</code>	Criteria are absent, null or blank (must be key=value(s))
<code>No_Results</code>	There are no properties that fit the specified criteria
<code>Not_Available</code>	The property is not available for the specified period

Example REST URL

<http://razorpms.com/service/xml/ota/HotelSearchRS/Latitude=33.95,Longitude=18.37,Radius=3,MaxResponses=10?pos=8508a7e6ce43e091>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_HotelSearchRS PrimaryLangID="en"
  RetransmissionIndicator="false" SequenceNmbr="1" TransactionIdentifier="1"
  Version="2.004" Target="Production" TimeStamp="2010-07-21T21:41:52"
  EchoToken="OTA_HotelSearchRQ 5, Latitude=-33.95,Longitude=18.37,Radius=1,
8508a7e6ce43e091"
  xmlns="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <Success />
  <Properties>
    <Property BrandName="100 Geneva Drive Partnership"
      HotelCodeContext="RZ" HotelName="100 Geneva Drive" HotelCityCode="CPT"
      HotelCode="121" BrandCode="6729" ChainCode="5" MaxGroupRoomQuantity="0">
      <Position AltitudeUnitOfMeasureCode="MTR" Altitude="0.0"
        Longitude="18.3884769102589" Latitude="-33.9487332161371" />
      <Address UseType="3">
        <StreetNmbr>P.O.Box 32452
          Camps Bay
          Cape Town</StreetNmbr>
        <PostalCode>8040</PostalCode>
      </Address>
      <ContactNumbers>
        <ContactNumber PhoneNumber="+27 21 424 3353"
          PhoneUseType="3" PhoneTechType="1" />
        <ContactNumber PhoneNumber="+27 21 424 3355"
          PhoneUseType="3" PhoneTechType="3" />
        <ContactNumber PhoneNumber="+27 84 804 9902"
          PhoneUseType="5" PhoneTechType="5" />
      </ContactNumbers>
      <RateRange CurrencyCode="ZAR" RateTimeUnit="Day" FixedRate="3500.0" />
    </Property>
```

.....repeated for all properties that meet the criteria

```
<Property BrandName="Cape Luxury Rentals" HotelCodeContext="RZ"
  HotelName="Henley Manor" HotelCityCode="CPT" HotelCode="257"
  BrandCode="8007" ChainCode="8007" MaxGroupRoomQuantity="0">
  <Position AltitudeUnitOfMeasureCode="MTR" Altitude="null"
    Longitude="18.4011769294739" Latitude="-33.9014102979927" />
  <Address UseType="3">
```

```
        <PostalCode></PostalCode>
    </Address>
    <ContactNumbers>
        <ContactNumber PhoneNumber="021 433 0142"
            PhoneUseType="3" PhoneTechType="1" />
    </ContactNumbers>
    <RateRange CurrencyCode="ZAR" RateTimeUnit="Day" />
</Property>
</Properties>
<Criteria>
    <Criterion>
        <Position Latitude="-33.95" />
    </Criterion>
    <Criterion>
        <Position Longitude="18.37" />
    </Criterion>
    <Criterion>
        <Position MaxResponses="10" />
    </Criterion>
    <Criterion>
        <Radius UnitOfMeasureCode="2" Distance="3" />
    </Criterion>
</Criteria>
</OTA_HotelSearchRS>
```

OTA_PingRQ

This document is to test the connection to a web service via a URL that returns an OTA_PingRQ document. It does not perform any operation, but illustrates the request document required by the SOAP service.

REST URL

http://razorpms.com/service/xml/ota/PingRQ?pos=pos_code

Parameters

`pos_code` Point of sale code requesting the ping

Example REST URL

<http://razorpms.com/service/xml/ota/PingRS?pos=8508a7e6ce43e091>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_PingRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="1.005" Target="Production"
  Timestamp="2010-07-21T12:43:07" EchoToken="OTA_PingRQ 5, 8508a7e6ce43e091" />
```

OTA_PingRS

This document is to test the connection via a URL that creates an OTA_PingRQ and returns an OTA_PingRS document if the connection exists with the service.

REST URL

http://razorpms.com/service/xml/ota/PingRQ?pos=pos_code

Parameters

`pos_code` Point of sale code requesting the ping

Example REST URL

<http://razorpms.com/service/xml/ota/PingRS/8508a7e6ce43e091?id=5>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_PingRS xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="1.007" Target="Production"
  Timestamp="2010-07-21T12:43:07" EchoToken="OTA_PingRQ 5, 8508a7e6ce43e091" />
```

OTA_ProfileCreateRQ

This document is to notify the profile of a customer or a company via a URL that returns an OTA_ProfileCreateRQ document created with the specified parameters. It does not perform any operation, but illustrates the request document required by the SOAP service.

REST URL

http://razorpms.com/service/xml/ota/ProfileCreateRQ/partyid?pos=pos_code

Parameters

<i>partyid</i>	Customer(s) for which a profile is requested
<i>pos_code</i>	Point of sale code requesting the information

Example REST URL

<http://razorpms.com/service/xml/ota/ProfileCreateRQ/5?pos=8508a7e6ce43e091>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_ProfileCreateRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="2.004" Target="Production"
  TimeStamp="2010-07-21T13:06:20" EchoToken="OTA_ProfileCreateRQ 5, 5,
8508a7e6ce43e091">
  <UniqueID ID="5" ID_Context="RZ" Type="1" URL="www.noxrentals.co.za" />
  <Profile CreateDateTime="2010-07-21T13:06:20" StatusCode="">
    <Customer BirthDate="2002-02-07" CurrencyCode="ZAR"
      Text="Nox Tourism was founded in September 2003 by Richard Marshall. We
are market leaders on Cape Town's Atlantic Seaboard, and continue to innovate in terms
of our service to both the travel trade and to the guests that stay in our villas and
apartments. All of our properties are in the four or five star market, and options range
from one bedroom apartments to seven bedroom villas, from penthouse apartments to beach
cottages. However different they may be, each property has what we call tourism
potential and has something unique and exceptional about it, whether that be views,
location, luxury or a combination of these and many other factors. We have an obsessive
focus on the guest experience, and every guest's needs are catered to by our dedicated
team of guest relations consultants, property managers and housekeepers."
      VIP_Indicator="false">
      <PersonName NameType="2">
        <GivenName>Richard</GivenName>
        <Surname>Marshall</Surname>
      </PersonName>
      <Email DefaultInd="true" EmailType="1">info@noxrentals.co.za
      </Email>
    </Customer>
  </Profile>
</OTA_ProfileCreateRQ>
```

OTA_ProfileCreateRS

This document is to notify the profile of a customer or a company via a URL that creates an OTA_ProfileCreateRQ with the specified parameters and returns an OTA_ProfileCreateRS document. This is typically used to send information to another system via a SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/ProfileCreateRS/partyid?pos=pos_code`

Parameters

<code>partyid</code>	Customer(s) for which profile is requested
<code>pos_code</code>	Point of sale code requesting the information

Errors

<code>Profile</code>	Profile is absent, null or blank
----------------------	----------------------------------

Example REST URL

`http://razorpms.com/service/xml/ota/ProfileCreateRQ/5?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_ProfileCreateRS xmlns="http://www.opentravel.org/OTA/2003/05"
  RetransmissionIndicator="false" SequenceNmbr="1" TransactionIdentifier="1"
  Version="1.008" Target="Production" TimeStamp="2010-07-21T13:06:20"
  EchoToken="OTA_ProfileCreateRQ 5, 5, 8508a7e6ce43e091">
  <Success />
</OTA_ProfileCreateRS>
```

OTA_ReadRQ

This document is to request an existing reservation via a URL that returns an OTA_ReadRQ document with the specified parameters. It does not perform any operation, but illustrates the request document required by the SOAP service.

REST URL

`http://razorpms.com/service/xml/ota/ReadRQ/reservationid?pos=pos_code`

Parameters

<code>reservationid</code>	Identity of reservation for which a information is requested
<code>pos_code</code>	Point of sale code requesting the information

Example REST URL

`http://razorpms.com/service/xml/ota/ReadRQ/102503/7171?pos=8508a7e6ce43e091`

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_ReadRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNbr="1"
  TransactionIdentifier="1" Version="1.011" Target="Production"
  Timestamp="2010-07-28T12:57:14" EchoToken="OTA_ReadRQ 5, 102503, 8508a7e6ce43e091">
  <POS>
    <Source>
      <RequestorID ID="8508a7e6ce43e091" ID_Context="RZ" />
    </Source>
  </POS>
  <ReadRequests>
    <ReadRequest>
      <UniqueID ID="102503" ID_Context="RZ" Type="14" />
    </ReadRequest>
  </ReadRequests>
</OTA_ReadRQ>
```

OTA_ResRetrieveRS

This document is to request an existing reservation via a URL that creates an OTA_ReadRQ document with the specified parameters and returns an OTA_ResRetrieveRS document with the requested information, or list any errors that may have prevented the request being fulfilled.

REST URL

http://razorpms.com/service/xml/ota/ResRetrieveRS/reservationid?pos=pos_code

Parameters

<i>reservationid</i>	Identity of reservation for which a information is requested
<i>pos_code</i>	Point of sale code requesting the information

Errors

<i>ChainCode_Invalid</i>	Manager id is not valid
<i>Customer_Invalid</i>	Reservation customer (guest) not valid
<i>HotelCode_Invalid</i>	Reservation property(s) not valid
<i>UniqueID</i>	UniqueID is absent, null or blank (must be <i>reservationid</i>)
<i>UniqueID_Context</i>	UniqueID context is not valid (must be "RZ")
<i>UniqueID_Invalid</i>	UniqueID is not valid (<i>reservationid</i> must be a valid reservation)
<i>UniqueID_Type</i>	UniqueID type is not valid (must be "14")

Example REST URL

<http://razorpms.com/service/xml/ota/ResRetrieveRS/102503/7171?pos=8508a7e6ce43e091>

Example OTA Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OTA_ResRetrieveRS xmlns="http://www.opentravel.org/OTA/2003/05"
  PrimaryLangID="en" RetransmissionIndicator="false" SequenceNmbr="1"
  TransactionIdentifier="1" Version="5.001" Target="Production"
  Timestamp="2010-07-28T12:57:17" EchoToken="OTA_ReadRQ 5, 102503, 7171">
  <Success />
  <ReservationsList>
    <HotelReservation CreatorID="7171" CreateDateTime="2010-05-29T00:00:00"
      ResStatus="FullyPaid">
      <ResGuests>
        <ResGuest CodeContext="RZ" DepartureTime="10:00:00"
          ArrivalTime="14:00:00">
          <Profiles>
            <ProfileInfo />
            <ProfileInfo>
              <Profile ProfileType="1" StatusCode="">
                <Customer>
                  <PersonName>
                    <GivenName>Family</GivenName>
                    <Surname>Rechain</Surname>
                  </PersonName>
                  <Email DefaultInd="true" EmailType="1">
                    jeanmarc@rechain.com
                  </Email>
                </Customer>
              </Profile>
            </ProfileInfo>
          </Profiles>
        </ResGuest>
      </ResGuests>
      <ResGlobalInfo>
        <TimeSpan End="2010-08-15" Duration="4.0" Start="2010-08-11" />
        <SpecialRequests>
          <SpecialRequest Language="en">
            <Text Formatted="false">FOUNTAIN 1105
              arrival - 12h00-12h30
              departure - 05h30
              JE135 arrives at 10h40 from JNB
              2 adults and 2 children age 13 and 9
              Names and ages of all guests.
            </Text>
          </SpecialRequest>
        </SpecialRequests>
      </ResGlobalInfo>
    </HotelReservation>
  </ReservationsList>
</OTA_ResRetrieveRS>
```

```

    Mr Jean-Marc Rechain
    Mrs Sylvie Chataignier
    Mr Antoine Rechain - 13
    Miss Alice Rechain - 9</Text>
  </SpecialRequest>
</SpecialRequests>
<DepositPayments>
  <GuaranteePayment NoCardHolderInfoReqInd="false"
    Type="31">
    <AcceptedPayments />
  </GuaranteePayment>
</DepositPayments>
<CancelPenalties CancelPolicyIndicator="true">
  <CancelPenalty>
    <Deadline OffsetUnitMultiplier="30" OffsetTimeUnit="Day" />
    <AmountPercent Percent="100" />
    <PenaltyDescription Language="en"
      CreateDateTime="2010-07-28T12:57:17">
      <Text Formatted="false">Refunds and waiver of cancellation
policies are at the discretion of management. Refunds are dependent on re-letting of the
room. Guests are advised to take out cancellation insurance.</Text>
    </PenaltyDescription>
  </CancelPenalty>
  <CancelPenalty>
    <Deadline OffsetUnitMultiplier="60" OffsetTimeUnit="Day" />
    <AmountPercent Percent="50" />
  </CancelPenalty>
  <CancelPenalty>
    <Deadline OffsetUnitMultiplier="90" OffsetTimeUnit="Day" />
    <AmountPercent Percent="25" />
  </CancelPenalty>
</CancelPenalties>
<Total CurrencyCode="ZAR" AmountAfterTax="3600.0" />
<HotelReservationIDs>
  <HotelReservationID ResID_Date="2010-05-29T00:00:00"
    ResID_Value="102503" ResID_Type="14" />
</HotelReservationIDs>
</ResGlobalInfo>
</HotelReservation>
</ReservationsList>
</OTA_ResRetrieveRS>

```

SOURCE CODE EXAMPLES

Razor uses the Apache CXF Open Source Service Framework for both SOAP and REST services, and the JAXB Java Architecture for Data Binding. Developers wishing to use Razor web services may wish to familiarize themselves with these technologies:

- Unofficial JAXB Guide <https://jaxb.dev.java.net/guide/>
- JAXB Home Page <http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>
- Apache CXF <http://cxf.apache.org/>
- CXF User's Guide <http://cwiki.apache.org/CXF20DOC/>

The services have been developed using the contract first approach in which Java code has been generated from the following WSDL files. Note that the RazorService.wsdl implementation file imports the RazorInterface.wsdl via an import statement, and the RazorInterface.wsdl imports OTA XML schema via import:xs statements. The OTA2008B xsd files are therefore also required, as detailed in the OTA_ImplementationGuide_WSDL_20060410.pdf.

The JAXB class libraries and object factory are available at <http://razorpms.com/service/opentravel-2008b-jaxb.jar>, and the example code listed below is at <http://razorpms.com/service/Examples.java>, so it is not necessary to generate code if they are used. The interface and binding WSDL files are at <http://razorpms.com/service/RazorInterface.wsdl> and at <http://razorpms.com/service/RazorService.wsdl>. Examples of how the libraries may be used to formulate SOAP request documents are listed in the OTA Requests with JAXB section.

RazorInterface.wsdl Interface Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ota="http://www.opentravel.org/OTA/2003/05"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.opentravel.org/OTA/2003/05" name="RazorService">

  <!-- Define data types (import OTA schemas) -->
  <wsdl:types>

    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
        schemaLocation="http://www.opentravel.org/2008B/OTA_CancelRQ.xsd" />
    </xs:schema>
    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
        schemaLocation="http://www.opentravel.org/2008B/OTA_CancelRS.xsd" />
    </xs:schema>

    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
        schemaLocation="http://www.opentravel.org/2008B/OTA_HotelAvailRQ.xsd" />
    </xs:schema>
    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
        schemaLocation="http://www.opentravel.org/2008B/OTA_HotelAvailRS.xsd" />
    </xs:schema>

    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
        schemaLocation="http://www.opentravel.org/2008B/OTA_HotelDescriptiveInfoRQ.xsd"
      />
    </xs:schema>
    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
        schemaLocation="http://www.opentravel.org/2008B/OTA_HotelDescriptiveInfoRS.xsd"
      />
    </xs:schema>

    <xs:schema>
      <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
```

```

        schemaLocation="http://www.opentravel.org/2008B/OTA_HotelRatePlanRQ.xsd" />
</xs:schema>
<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_HotelRatePlanRS.xsd" />
</xs:schema>

<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_HotelResNotifRQ.xsd" />
</xs:schema>
<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_HotelResNotifRS.xsd" />
</xs:schema>

<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_PingRQ.xsd" />
</xs:schema>
<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_PingRS.xsd" />
</xs:schema>

<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_ProfileCreatorRQ.xsd" />
</xs:schema>
<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_ProfileCreatorRS.xsd" />
</xs:schema>

<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_HotelSearchRQ.xsd" />
</xs:schema>
<xs:schema>
  <xs:import namespace="http://www.opentravel.org/OTA/2003/05"
    schemaLocation="http://www.opentravel.org/2008B/OTA_HotelSearchRS.xsd" />
</xs:schema>

</wsdl:types>

<!-- Define request and response messages -->

<wsdl:message name="CancelRQ">
  <wsdl:part name="OTA_CancelRQ" element="ota:OTA_CancelRQ" />
</wsdl:message>
<wsdl:message name="CancelRS">
  <wsdl:part name="OTA_CancelRS" element="ota:OTA_CancelRS" />
</wsdl:message>

<wsdl:message name="HotelAvailRQ">
  <wsdl:part name="OTA_HotelAvailRQ" element="ota:OTA_HotelAvailRQ" />
</wsdl:message>
<wsdl:message name="HotelAvailRS">
  <wsdl:part name="OTA_HotelAvailRS" element="ota:OTA_HotelAvailRS" />
</wsdl:message>

<wsdl:message name="HotelDescriptiveInfoRQ">
  <wsdl:part name="OTA_HotelDescriptiveInfoRQ"
    element="ota:OTA_HotelDescriptiveInfoRQ" />
</wsdl:message>
<wsdl:message name="HotelDescriptiveInfoRS">
  <wsdl:part name="OTA_HotelDescriptiveInfoRS"

```

```

        element="ota:OTA_HotelDescriptiveInfoRS" />
</wsdl:message>

<wsdl:message name="HotelRatePlanRQ">
  <wsdl:part name="OTA_HotelRatePlanRQ" element="ota:OTA_HotelRatePlanRQ" />
</wsdl:message>
<wsdl:message name="HotelRatePlanRS">
  <wsdl:part name="OTA_HotelRatePlanRS" element="ota:OTA_HotelRatePlanRS" />
</wsdl:message>

<wsdl:message name="HotelResNotifRQ">
  <wsdl:part name="OTA_HotelResNotifRQ" element="ota:OTA_HotelResNotifRQ" />
</wsdl:message>
<wsdl:message name="HotelResNotifRS">
  <wsdl:part name="OTA_HotelResNotifRS" element="ota:OTA_HotelResNotifRS" />
</wsdl:message>

<wsdl:message name="HotelSearchRQ">
  <wsdl:part name="OTA_HotelSearchRQ" element="ota:OTA_HotelSearchRQ" />
</wsdl:message>
<wsdl:message name="HotelSearchRS">
  <wsdl:part name="OTA_HotelSearchRS" element="ota:OTA_HotelSearchRS" />
</wsdl:message>

<wsdl:message name="PingRQ">
  <wsdl:part name="OTA_PingRQ" element="ota:OTA_PingRQ" />
</wsdl:message>
<wsdl:message name="PingRS">
  <wsdl:part name="OTA_PingRS" element="ota:OTA_PingRS" />
</wsdl:message>

<wsdl:message name="ProfileCreateRQ">
  <wsdl:part name="OTA_ProfileCreateRQ" element="ota:OTA_ProfileCreateRQ" />
</wsdl:message>
<wsdl:message name="ProfileCreateRS">
  <wsdl:part name="OTA_ProfileCreateRS" element="ota:OTA_ProfileCreateRS" />
</wsdl:message>

<!-- Define operation and reference messages-->
<wsdl:portType name="RazorPortType">

  <wsdl:operation name="CancelReservation">
    <wsdl:input message="ota:CancelRQ" />
    <wsdl:output message="ota:CancelRS" />
  </wsdl:operation>

  <wsdl:operation name="GetPropertyAvailability">
    <wsdl:input message="ota:HotelAvailRQ" />
    <wsdl:output message="ota:HotelAvailRS" />
  </wsdl:operation>

  <wsdl:operation name="GetPropertyDetails">
    <wsdl:input message="ota:HotelDescriptiveInfoRQ" />
    <wsdl:output message="ota:HotelDescriptiveInfoRS" />
  </wsdl:operation>

  <wsdl:operation name="ReserveProperty">
    <wsdl:input message="ota:HotelResNotifRQ" />
    <wsdl:output message="ota:HotelResNotifRS" />
  </wsdl:operation>

  <wsdl:operation name="GetPropertyPrices">
    <wsdl:input message="ota:HotelRatePlanRQ" />
    <wsdl:output message="ota:HotelRatePlanRS" />
  </wsdl:operation>

  <wsdl:operation name="FindProperty">

```

```
    <wsdl:input message="ota:HotelSearchRQ" />
    <wsdl:output message="ota:HotelSearchRS" />
</wsdl:operation>

<wsdl:operation name="Ping">
  <wsdl:input message="ota:PingRQ" />
  <wsdl:output message="ota:PingRS" />
</wsdl:operation>

<wsdl:operation name="CreateCustomerProfile">
  <wsdl:input message="ota:ProfileCreateRQ" />
  <wsdl:output message="ota:ProfileCreateRS" />
</wsdl:operation>

</wsdl:portType>
</wsdl:definitions>
```

RazorService.wsdl Binding Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ota="http://www.opentravel.org/OTA/2003/05"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.opentravel.org/OTA/2003/05">

  <!-- Import Interface Definition WSDL-->
  <import namespace="http://www.opentravel.org/OTA/2003/05"
location="RazorInterface.wsdl"/>

  <!-- Define SOAP binding-->
  <binding name="RazorBinding" type="ota:RazorPortType">
    <!-- Use document style and not rpc-->
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"
  />

  <operation name="CancelReservation">
    <soap:operation soapAction="CancelReservation" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>

  <operation name="GetPropertyAvailability">
    <!-- Use 'literal' to include OTA XML as-is-->
    <soap:operation soapAction="GetPropertyAvailability" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>

  <operation name="GetPropertyDetails">
    <soap:operation soapAction="GetPropertyDetails" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>

  <operation name="GetPropertyPrices">
    <soap:operation soapAction="GetPropertyPrices" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>

  <operation name="FindProperty">
    <soap:operation soapAction="SearchForProperties" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
  </operation>
</definitions>
```

```

        <output>
            <soap:body use="literal" />
        </output>
    </operation>

    <operation name="Ping">
        <soap:operation soapAction="Ping" style="document" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>

    <operation name="CreateCustomerProfile">
        <soap:operation soapAction="CreateCustomerProfile" style="document" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>

</binding>

<!-- Define SOAP interface with previously declared binding-->
<service name="RazorService">
    <port name="RazorPort" binding="ota:RazorBinding">
        <!-- Replace "http://mydomain/myservicename" with actual service endpoint -->
        <soap:address location="http://razorpms.com/service/soap/ota" />
    </port>
</service>
</definitions>

```

OTA Requests with JAXB

```
/*
 * @copyright      CBT Limited, London, UK
 * @author        CJ Marshall
 * @version       0.91
 * @see          License at http://www.fliteplan.co.uk/License.html
 *
 * Demonstration code for creating SOAP request documents that conform to the OTA2008B
 * XML schema and that uses the Java Architecture for XML Binding (JAXB) class library
 * at http://razorpms.com/service/opentravel-2008b-jaxb.jar or similar resources.
 */

import java.math.BigDecimal;
import java.math.BigInteger;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Map;

import javax.xml.datatype.DatatypeConfigurationException;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.datatype.XMLGregorianCalendar;

import org.opentravel.ota2008b.BasicPropertyInfoType;
import org.opentravel.ota2008b.CustomerType;
import org.opentravel.ota2008b.DateTimespanType;
import org.opentravel.ota2008b.FreeTextType;
import org.opentravel.ota2008b.GuestCountType;
import org.opentravel.ota2008b.HotelReservationType;
import org.opentravel.ota2008b.HotelReservationsType;
import org.opentravel.ota2008b.HotelSearchCriteriaType;
import org.opentravel.ota2008b.HotelSearchCriterionType;
import org.opentravel.ota2008b.OTA_CancelRQ;
import org.opentravel.ota2008b.OTA_HotelAvailRQ;
import org.opentravel.ota2008b.OTA_HotelDescriptiveInfoRQ;
import org.opentravel.ota2008b.OTA_HotelResNotifRQ;
import org.opentravel.ota2008b.OTA_HotelSearchRQ;
import org.opentravel.ota2008b.OTA_PingRQ;
import org.opentravel.ota2008b.OTA_ProfileCreateRQ;
import org.opentravel.ota2008b.OTA_ReadRQ;
import org.opentravel.ota2008b.ObjectFactory;
import org.opentravel.ota2008b.POS_Type;
import org.opentravel.ota2008b.PaymentCardType;
import org.opentravel.ota2008b.PaymentFormType;
import org.opentravel.ota2008b.PersonNameType;
import org.opentravel.ota2008b.ProfileType;
import org.opentravel.ota2008b.ProfilesType;
import org.opentravel.ota2008b.ResGuestsType;
import org.opentravel.ota2008b.RoomAmenityPrefType;
import org.opentravel.ota2008b.RoomStayCandidateType;
import org.opentravel.ota2008b.RoomStaysType;
import org.opentravel.ota2008b.SourceType;
import org.opentravel.ota2008b.UniqueID_Type;
import org.opentravel.ota2008b.AvailRequestSegmentsType.AvailRequestSegment;
import org.opentravel.ota2008b.CustomerType.Email;
import org.opentravel.ota2008b.GuestCountType.GuestCount;
import org.opentravel.ota2008b.HotelSearchCriteriaType.Criterion;
import org.opentravel.ota2008b.HotelSearchCriterionType.HotelAmenity;
import org.opentravel.ota2008b.HotelSearchCriterionType.HotelFeature;
import org.opentravel.ota2008b.HotelSearchCriterionType.RoomStayCandidates;
import org.opentravel.ota2008b.ItemSearchCriterionType.HotelRef;
import org.opentravel.ota2008b.ItemSearchCriterionType.Radius;
import org.opentravel.ota2008b.OTA_CancelRQ.Reasons;
import org.opentravel.ota2008b.OTA_CancelRQ.UniqueID;
```

```

import org.opentravel.ota2008b.OTA_HotelAvailRQ.AvailRequestSegments;
import org.opentravel.ota2008b.OTA_HotelDescriptiveInfoRQ.HotelDescriptiveInfos;
import org.opentravel.ota2008b.OTA_ReadRQ.ReadRequests;
import org.opentravel.ota2008b.OTA_ReadRQ.ReadRequests.ReadRequest;
import org.opentravel.ota2008b.ProfilesType.ProfileInfo;
import org.opentravel.ota2008b.ResGuestsType.ResGuest;
import org.opentravel.ota2008b.SourceType.RequestorID;

```

```

public class Examples {

```

```

    private final static BigDecimal VER_1005 = new BigDecimal("1.005");
    private final static BigDecimal VER_1009 = new BigDecimal("1.009");
    private final static BigDecimal VER_1010 = new BigDecimal("1.010");
    private final static BigDecimal VER_1011 = new BigDecimal("1.011");
    private final static BigDecimal VER_2004 = new BigDecimal("2.004");
    private final static BigDecimal VER_4001 = new BigDecimal("4.001");
    private final static BigInteger SEQUENCE = new BigInteger("1");

    private final static String ARC = "ARC";//architecture style
    private final static String BED = "BED";//bed type
    private final static String HAC = "HAC";//hotel amenity code
    private final static String LOC = "LOC";//location category
    private final static String PCT = "PCT";//property class type
    private final static String PHY = "PHY";//physically disabled accessibility
    private final static String PMT = "PMT";//payment type
    private final static String REC = "REC";//recreation service detail code
    private final static String RLT = "RLT";//room location
    private final static String RMA = "RMA";//room amenity type
    private final static String RVT = "RVT";//room view
    private final static String SEC = "SEC";//security feature
    private final static String SEG = "SEG";//segment category code

    private final static String AQC_INFANT = "7";
    private final static String AQC_CHILD = "8";
    private final static String AQC_ADULT = "10";
    private final static String CDT_CREDIT = "1";
    private final static String EAT_PERSONAL = "1";
    private final static String PRT_CUSTOMER = "1";
    private final static String UIT_CUSTOMER = "1";
    private final static String UIT_RESERVATION = "14";
    private final static String UOM_KILOMETERS = "2";
    private final static String UOM_MILES = "1";
    private final static String UOM_METERS = "3";
    private final static String UOM_YARDS = "6";

    private final static String CONTEXT = "RZ";
    private final static String CRITERION_ADULT_COUNT = "Adults";
    private final static String CRITERION_ARC = "ArchitectureStyle";
    private final static String CRITERION_ARRIVAL_DATE = "ArrivalDate";
    private final static String CRITERION_BED = "BedType";
    private final static String CRITERION_CARD_CODE = "CardCode";
    private final static String CRITERION_CHILD_COUNT = "Children";
    private final static String CRITERION_DEPARTURE_DATE = "DepartureDate";
    private final static String CRITERION_DURATION = "Duration";
    private final static String CRITERION_HAC = "Amenity";
    private final static String CRITERION_HOTEL_AREA_ID = "AreaID";
    private final static String CRITERION_HOTEL_BRAND_CODE = "BrandCode";
    private final static String CRITERION_HOTEL_BRAND_NAME = "BrandName";
    private final static String CRITERION_HOTEL_CHAIN_CODE = "ChainCode";
    private final static String CRITERION_HOTEL_CHAIN_NAME = "ChainName";
    private final static String CRITERION_HOTEL_CITY_CODE = "CityCode";
    private final static String CRITERION_HOTEL_CODE = "HotelCode";
    private final static String CRITERION_HOTEL_NAME = "HotelName";
    private final static String CRITERION_INFANT_COUNT = "Infants";
    private final static String CRITERION_LATITUDE = "Latitude";
    private final static String CRITERION_LOC = "LocationCategory";
    private final static String CRITERION_LONGITUDE = "Longitude";

```

```
private final static String CRITERION_MAX_RESPONSES = "MaxResponses";
private final static String CRITERION_PCT = "PropertyClass";
private final static String CRITERION_PHY = "Accessibility";
private final static String CRITERION_PMT = "PaymentType";
private final static String CRITERION_RADIUS = "Radius";
private final static String CRITERION_RATE_CURRENCY = "Currency";
private final static String CRITERION_RATE_MAXIMUM = "MaxRate";
private final static String CRITERION_RATE_MINIMUM = "MinRate";
private final static String CRITERION_REC = "Recreation";
private final static String CRITERION_RLT = "RoomLocation";
private final static String CRITERION_RMA = "RoomAmenity";
private final static String CRITERION_ROOM_TYPE = "Accommodation";
private final static String CRITERION_RVT = "View";
private final static String CRITERION_SEC = "Security";
private final static String CRITERION_SEG = "Category";
private final static String DEFAULT_CURRENCY_CODE = "ZAR";
private final static String DEFAULT_LANGUAGE_CODE = "en";
private final static String PROMOTION_VENDOR_CODE = "1";
private final static String TARGET = "Production";
private final static String TRANSACTION_IDENTIFIER = "1";

private static final ObjectFactory of = new ObjectFactory();
```

```

/*****
 * @param id of reservation to be cancelled
 * @param reason for cancellation
 * @param pos_code in CONTEXT (RZ for Razor) of the request originator
 * @return OTA_CancelRS request to cancel reservation
 * @see OTA_MessageUsersGuide2008BV1.0.pdf Section 3.2
 *****/
public static OTA_CancelRQ cancelRQ(
    String id,
    String reason,
    String pos_code
) throws Exception {

    OTA_CancelRQ rq = of.createOTA_CancelRQ();
    rq.setEchoToken("OTA_CancelRQ " + id + ", " + reason + ", " + pos_code);
    rq.setPOS(getPOS(pos_code, CONTEXT));
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNbr(SEQUENCE);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_1010);

    Reasons reasons = of.createOTA_CancelRQReasons();
    rq.setReasons(reasons);
    reasons.getReason().add(getFreeText(reason, DEFAULT_LANGUAGE_CODE));

    UniqueID uid = of.createOTA_CancelRQUniqueID();
    rq.getUniqueID().add(uid);
    uid.setID(id);
    uid.setID_Context(CONTEXT);
    uid.setType(UIT_RESERVATION);
    return rq;
}

```

```

/*****
 * @param ids list of property identities for which availability is requested
 * @param fromdate date from which the availability is requested
 * @param todate date to which the availability is requested
 * @param pos_code in context of CONTEXT (RZ for Razor) making the request
 * @return OTA_HotelAvailRQ hotel availability request
 * @see OTA_MessageUsersGuide2008BV1.0.pdf Section 10.1
 *****/
public static OTA_HotelAvailRQ hotelAvailRQ(
    List<String> ids,
    String fromdate,
    String todate,
    String pos_code
) {

    OTA_HotelAvailRQ rq = of.createOTA_HotelAvailRQ();
    rq.setEchoToken("OTA_HotelAvailRQ " + ids + ", " + fromdate
        + ", " + todate + ", " + pos_code);
    rq.setPOS(getPOS(pos_code, CONTEXT));

    AvailRequestSegments availRequestSegments =
        of.createOTA_HotelAvailRQAvailRequestSegments();
    rq.setAvailRequestSegments(availRequestSegments);
    AvailRequestSegment availRequestSegment =
        of.createAvailRequestSegmentsTypeAvailRequestSegment();
    availRequestSegments.getAvailRequestSegment().add(availRequestSegment);
    HotelSearchCriteriaType hotelSearchCriteria = of.createHotelSearchCriteriaType();
    availRequestSegment.setHotelSearchCriteria(hotelSearchCriteria);
    Criterion criterion = of.createHotelSearchCriteriaTypeCriterion();
    hotelSearchCriteria.getCriterion().add(criterion);

    for (String id : ids) {
        HotelRef hotelRef = of.createItemSearchCriterionTypeHotelRef();
        criterion.getHotelRef().add(hotelRef);
        hotelRef.setHotelCode(id);
        hotelRef.setHotelCodeContext(CONTEXT);
        DateTimeSpanType stayDateRange = of.createDateTimeSpanType();
        criterion.setStayDateRange(stayDateRange);
        criterion.getStayDateRange().setStart(fromdate);
        criterion.getStayDateRange().setEnd(todate);
    }

    return rq;
}

```

```

/*****
 * @param ids list of properties for which descriptive information is requested
 * @param pos_code in context of CONTEXT (RZ for Razor) making the request
 * @return OTA_HotelDescriptiveInfoRQ hotel descriptive info request
 * @see OTA_MessageUsersGuide2008BV1.0.pdf Section 10.7
 *****/
public static OTA_HotelDescriptiveInfoRQ hotelDescriptiveInfoRQ(
    List<String> ids,
    String pos_code
) {
    OTA_HotelDescriptiveInfoRQ rq = of.createOTA_HotelDescriptiveInfoRQ();
    rq.setEchoToken("OTA_HotelDescriptiveInfoRQ " + ids + ", " + pos_code);
    rq.setPOS(getPOS(pos_code, CONTEXT));
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNmbr(SEQUENCE);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_1009);

    HotelDescriptiveInfos hotelDescriptiveInfos =
        of.createOTA_HotelDescriptiveInfoRQHotelDescriptiveInfos();
    rq.setHotelDescriptiveInfos(hotelDescriptiveInfos);

    for (String propertyid : ids) {
        HotelDescriptiveInfo hotelDescriptiveInfo =
            of.createOTA_HotelDescriptiveInfoRQHotelDescriptiveInfosHotelDescriptiveInfo();
        hotelDescriptiveInfo.setHotelCode(propertyid);
        hotelDescriptiveInfo.setHotelCodeContext(CONTEXT);
        rq.getHotelDescriptiveInfos().getHotelDescriptiveInfo()
            .add(hotelDescriptiveInfo);
    }

    return rq;
}

```

```

/*****
* @param ids list of identities of properties to be reserved
* @param arrivaldate date of the reservation in yyyy-MM-dd format
* @param departuredate date of the reservation in yyyy-MM-dd format
* @param emailaddress of the guest for which the reservation is made
* @param surname or family name of the guest
* @param givenname or first name of the guest
* @param pos_code code in CONTEXT (RZ for Razor) of the request originator
* @return OTA_HotelResNotifRQ request to make a reservation
* @see OTA_MessageUsersGuide2008BV1.0.pdf Section 10.22
*****/
public static OTA_HotelResNotifRQ hotelResNotifRQ(
    List<String> ids,
    String arrivaldate,
    String departuredate,
    String emailaddress,
    String surname,
    String givenname,
    String pos_code
) {

    OTA_HotelResNotifRQ rq = of.createOTA_HotelResNotifRQ();
    rq.setEchoToken("OTA_HotelResNotifRQ " + ids + ", " + arrivaldate + ", "
        + departuredate + ", " + emailaddress + ", " + surname + ", "
        + givenname + ", " + pos_code);
    rq.setPOS(getPOS(pos_code, CONTEXT));
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNmbr(SEQUENCE);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_4001);

    HotelReservationsType hotelReservations = of.createHotelReservationsType();
    rq.setHotelReservations(hotelReservations);
    HotelReservationType hotelReservation = of.createHotelReservationType();
    hotelReservations.getHotelReservation().add(hotelReservation);
    RoomStaysType roomStays = of.createRoomStaysType();
    hotelReservation.setRoomStays(roomStays);

    for (String id : ids) {
        RoomStaysType.RoomStay roomStay = of.createRoomStaysTypeRoomStay();
        roomStays.getRoomStay().add(roomStay);
        roomStay.getPromotionVendorCode().add(PROMOTION_VENDOR_CODE);

        BasicPropertyInfoType basicPropertyInfo = of.createBasicPropertyInfoType();
        roomStay.setBasicPropertyInfo(basicPropertyInfo);
        basicPropertyInfo.setHotelCode(id);
        basicPropertyInfo.setHotelCodeContext(CONTEXT);

        DateTimeSpanType dateTimeSpan = of.createDateTimeSpanType();
        roomStay.setTimeSpan(dateTimeSpan);
        dateTimeSpan.setStart(arrivaldate);
        dateTimeSpan.setEnd(departuredate);
    }

    ResGuestsType resGuests = of.createResGuestsType();
    hotelReservation.setResGuests(resGuests);
    ResGuest resGuest = of.createResGuestsTypeResGuest();
    resGuests.getResGuest().add(resGuest);
    ProfilesType profiles = of.createProfilesType();
    resGuest.setProfiles(profiles);
    ProfileInfo profileInfo = of.createProfilesTypeProfileInfo();
    profiles.getProfileInfo().add(profileInfo);
    ProfileType profile = of.createProfileType();
    profileInfo.setProfile(profile);
}

```

```
profile.setProfileType(PRT_CUSTOMER);
CustomerType customer = of.createCustomerType();
profile.setCustomer(customer);
PersonNameType personName = of.createPersonNameType();
customer.getPersonName().add(personName);
personName.setSurname(surname);
personName.getGivenName().add(givename);
customer.getEmail().add(getCustomerEmail(emailaddress, EAT_PERSONAL));

return rq;
}
```

```

/*****
 * @param kvs as key value pairs that map the criterion name to its value
 * @param pos_code in CONTEXT (RZ for Razor) of the request originator
 * @return OTA_HotelSearchRQ request to search for properties that meet the criteria
 * @see OTA_MessageUsersGuide2008BV1.0.pdf Section 10.29
 *****/
public static OTA_HotelSearchRQ hotelSearchRQ(
    Map<String, String> kvs,
    String pos_code
) throws Exception {

    OTA_HotelSearchRQ rq = of.createOTA_HotelSearchRQ();
    rq.setEchoToken("OTA_HotelSearchRQ " + kvs + ", " + pos_code);
    rq.setPOS(getPOS(pos_code, CONTEXT));
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNmbr(SEQUENCE);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_2004);

    HotelSearchCriteriaType hotelSearchCriteria = of.createHotelSearchCriteriaType();
    rq.setCriteria(hotelSearchCriteria);

    Criterion criterion = of.createHotelSearchCriteriaTypeCriterion();
    hotelSearchCriteria.getCriterion().add(criterion);

    for (Map.Entry<String, String> kv : kvs.entrySet()) {
        String key = kv.getKey();
        String value = kv.getValue();

        if (CRITERION_CARD_CODE.equalsIgnoreCase(key)) {
            //card code from AX, MC, VI etc
            addCardCode(criterion, value, CDT_CREDIT);
        }
        else if (CRITERION_LATITUDE.equalsIgnoreCase(key)) {
            //latitude in Google Maps format
            if (criterion.getPosition() == null) {
                criterion.setPosition(of.createItemSearchCriterionTypePosition());
            }
            criterion.getPosition().setLatitude(value);
        }
        else if (CRITERION_LONGITUDE.equalsIgnoreCase(key)) {
            //longitude in Google Maps format
            if (criterion.getPosition() == null) {
                criterion.setPosition(of.createItemSearchCriterionTypePosition());
            }
            criterion.getPosition().setLongitude(value);
        }
        else if (CRITERION_RADIUS.equalsIgnoreCase(key)) {
            //radius from above postion to be searched
            Radius radius = of.createItemSearchCriterionTypeRadius();
            radius.setDistance(value);
            radius.setUnitOfMeasureCode(UOM_KILOMETERS);
            //radius unit in UOM_KILOMETERS, UOM_MILES, UOM_METERS or UOM_YARDS
            criterion.setRadius(radius);
        }
        else if (CRITERION_HOTEL_AREA_ID.equalsIgnoreCase(key)) {
            //id of property location - see Error: Reference source not found above
            addHotelRef(criterion).setAreaID(value);
        }
        else if (CRITERION_HOTEL_BRAND_CODE.equalsIgnoreCase(key)) {
            //id of property owner - see Error: Reference source not found above for Owner party type
            addHotelRef(criterion).setBrandCode(value);
        }
        else if (CRITERION_HOTEL_BRAND_NAME.equalsIgnoreCase(key)) {

```

```

//name of property owner
    addHotelRef(criterion).setBrandName(value);
}
else if (CRITERION_HOTEL_CHAIN_CODE.equalsIgnoreCase(key)) {
//id of property manager - Error: Reference source not found above for Supplier party
type
    addHotelRef(criterion).setChainCode(value);
}
else if (CRITERION_HOTEL_CHAIN_NAME.equalsIgnoreCase(key)) {
//name of property manager
    addHotelRef(criterion).setChainName(value);
}
else if (CRITERION_HOTEL_CITY_CODE.equalsIgnoreCase(key)) {
//property city code - see Error: Reference source not found above
    addHotelRef(criterion).setHotelCityCode(value);
}
else if (CRITERION_HOTEL_CODE.equalsIgnoreCase(key)) {
//id of property - see Error: Reference source not found above for product type
Accommodation
    Criterion.HotelRef hotelRef = addHotelRef(criterion);
    hotelRef.setHotelCode(value);
    hotelRef.setHotelCodeContext(CONTEXT);
}
else if (CRITERION_HOTEL_NAME.equalsIgnoreCase(key)) {
//name of property
    addHotelRef(criterion).setHotelName(value);
}
else if (CRITERION_ARRIVAL_DATE.equalsIgnoreCase(key)) {
//arrival date for available properties in yyyy-MM-dd format
    if (criterion.getStayDateRange() == null) {
        criterion.setStayDateRange(of.createDateTimeSpanType());
    }
    criterion.getStayDateRange().setStart(value);
}
else if (CRITERION_DEPARTURE_DATE.equalsIgnoreCase(key)) {
//departure date for available properties in yyyy-MM-dd format
    if (criterion.getStayDateRange() == null) {
        criterion.setStayDateRange(of.createDateTimeSpanType());
    }
    criterion.getStayDateRange().setEnd(value);
}
else if (CRITERION_DURATION.equalsIgnoreCase(key)) {
//duration in days from arrival date for availability
    if (criterion.getStayDateRange() == null) {
        criterion.setStayDateRange(of.createDateTimeSpanType());
    }
    criterion.getStayDateRange().setDuration(value);
}
else if (CRITERION_ADULT_COUNT.equalsIgnoreCase(key)) {
//number of adults to be accommodated
    addGuestCount(criterion, new Integer(value), AQC_ADULT);
}
else if (CRITERION_CHILD_COUNT.equalsIgnoreCase(key)) {
//number of children to be accommodated
    addGuestCount(criterion, new Integer(value), AQC_CHILD);
}
else if (CRITERION_INFANT_COUNT.equalsIgnoreCase(key)) {
//number of infants to be accommodated
    addGuestCount(criterion, new Integer(value), AQC_INFANT);
}
else if (CRITERION_ROOM_TYPE.equalsIgnoreCase(key)) {
//accommodation size - see Error: Reference source not found above for product attributes
    getRoomStayCandidate(criterion).setRoomType(value);
}
else if (CRITERION_RATE_CURRENCY.equalsIgnoreCase(key)) {
//ISO three character currency code for rate range
    getRateRange(criterion).setCurrencyCode(value);
}

```

```

    }
    else if (CRITERION_RATE_MAXIMUM.equalsIgnoreCase(key)) {
//maximum daily rate in above currency
        getRateRange(criterion).setMaxRate(getBigDecimal(value));
    }
    else if (CRITERION_RATE_MINIMUM.equalsIgnoreCase(key)) {
//optional minimum daily rate in above currency
        getRateRange(criterion).setMinRate(getBigDecimal(value));
    }
    else if (CRITERION_MAX_RESPONSES.equalsIgnoreCase(key)) {
//maximum number of properties to be included in response
        rq.setMaxResponses(getBigInteger(value));
    }
    else if (CRITERION_ARC.equalsIgnoreCase(key) || ARC.equalsIgnoreCase(key)) {
//ARC architecture style
        addHotelRef(criterion).setArchitecturalStyleCode(value);
    }
    else if (CRITERION_BED.equalsIgnoreCase(key) || BED.equalsIgnoreCase(key)) {
//BED bed type
        getRoomStayCandidate(criterion).getBedTypeCode().add(value);
    }
    else if (CRITERION_HAC.equalsIgnoreCase(key) || HAC.equalsIgnoreCase(key)) {
//HAC hotel amenity
        addHotelAmenity(criterion, value);
    }
    else if (CRITERION_LOC.equalsIgnoreCase(key) || LOC.equalsIgnoreCase(key)) {
//LOC location category
        addHotelRef(criterion).setHotelName(value);
    }
    else if (CRITERION_PCT.equalsIgnoreCase(key) || PCT.equalsIgnoreCase(key)) {
//PCT property class
        addHotelRef(criterion).setPropertyClassCode(value);
    }
    else if (CRITERION_PMT.equalsIgnoreCase(key) || PMT.equalsIgnoreCase(key)) {
//PMT payment types accepted
        addAcceptedPayment(criterion, value);
    }
    else if (CRITERION_PHY.equalsIgnoreCase(key) || PHY.equalsIgnoreCase(key)) {
//PHY physical accessibility feature
        addAccessibilityCode(criterion, value);
    }
    else if (CRITERION_REC.equalsIgnoreCase(key) || REC.equalsIgnoreCase(key)) {
//REC recreation service
        addRecreation(criterion, value);
    }
    else if (CRITERION_RLT.equalsIgnoreCase(key) || RLT.equalsIgnoreCase(key)) {
//RLT room location
        getRoomStayCandidate(criterion).setRoomLocationCode(value);
    }
    else if (CRITERION_RMA.equalsIgnoreCase(key) || RMA.equalsIgnoreCase(key)) {
//RMA room amenity
        addRoomAmenity(criterion, value);
    }
    else if (CRITERION_RVT.equalsIgnoreCase(key) || RVT.equalsIgnoreCase(key)) {
//RVT room view
        getRoomStayCandidate(criterion).setRoomViewCode(value);
    }
    else if (CRITERION_SEC.equalsIgnoreCase(key) || SEC.equalsIgnoreCase(key)) {
//SEC security feature
        addSecurityFeatureCode(criterion, value);
    }
    else if (CRITERION_SEG.equalsIgnoreCase(key) || SEG.equalsIgnoreCase(key)) {
//SEG segment category
        addHotelRef(criterion).setSegmentCategoryCode(value);
    }
}
return rq;

```

}

```

/*****
* @param pos_code in context of CONTEXT (RZ for Razor) making the request
* @return OTA_PingRQ request to test application connectivity
* @see OTA_MessageUsersGuide2008BV1.0.pdf Section 3.6
*****/
public static OTA_PingRQ pingRQ(String pos_code) throws Exception {
    OTA_PingRQ rq = of.createOTA_PingRQ();
    rq.setEchoToken("OTA_PingRQ " + pos_code);
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNbr(SEQUENCE);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_1005);
    return rq;
}

```

```

/*****
* @param id of customer to be uploaded
* @param surname or family name of the customer
* @param givenname or first name of the customer
* @param emailaddress of the customer
* @param URL of the customer's web site
* @param text description or comment relating to the customer
* @param birthdate date of birth of the customer in yyyy-MM-dd format
* @param pos_code in context of CONTEXT (RZ for Razor) making the request
* @return OTA_ProfileCreateRQ request to transfer informatin about a customer
* @see OTA_MessageUsersGuide2008BV1.0.pdf Section 17.1
*****/
public static OTA_ProfileCreateRQ profileCreateRQ(
    String id,
    String surname,
    String givenname,
    String emailaddress,
    String URL,
    String text,
    Date birthdate,
    boolean isVIP,
    String agentid
) throws Exception {

    OTA_ProfileCreateRQ rq = of.createOTA_ProfileCreateRQ();
    rq.setEchoToken("OTA_ProfileCreateRQ " + id + ", " + agentid);
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNmbr(SEQUENCE);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_4001);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setVersion(VER_2004);

    UniqueID_Type uniqueID = of.createUniqueID_Type();
    rq.getUniqueID().add(uniqueID);
    uniqueID.setID(id);
    uniqueID.setID_Context(CONTEXT);
    uniqueID.setType(UIT_CUSTOMER);
    uniqueID.setURL(URL);

    ProfileType profile = of.createProfileType();
    rq.setProfile(profile);
    profile.setCreateDateTime(getXGC());
    CustomerType customer = of.createCustomerType();
    profile.setCustomer(customer);
    customer.setBirthDate(getXGC(birthdate));
    customer.setCurrencyCode(DEFAULT_CURRENCY_CODE);
    customer.setVIP_Indicator(isVIP);
    customer.setText(text);
    PersonNameType personName = of.createPersonNameType();
    customer.getPersonName().add(personName);
    personName.setNameType(PRT_CUSTOMER);
    personName.setSurname(surname);
    personName.getGivenName().add(givenname);
    customer.getEmail().add(getCustomerEmail(emailaddress, EAT_PERSONAL));

    return rq;
}
/*****
* @param id of the reservation
* @param pos_code in context of CONTEXT (RZ for Razor) making the request
* @return OTA_ReadRQ request to retrieve an existing reservation record
* @see OTA_MessageUsersGuide2008BV1.0.pdf Section 3.8
*****/

```

```

public static OTA_ReadRQ otaReadRQ(
    String id,
    String pos_code
) {
    OTA_ReadRQ rq = of.createOTA_ReadRQ();
    rq.setEchoToken("OTA_ReadRQ " + id + ", " + pos_code);
    rq.setPOS(getPOS(pos_code, CONTEXT));
    rq.setPrimaryLangID(DEFAULT_LANGUAGE_CODE);
    rq.setRetransmissionIndicator(false);
    rq.setSequenceNmbr(SEQUENCE);
    rq.setTarget(TARGET);
    rq.setTimeStamp(getXGC());
    rq.setTransactionIdentifier(TRANSACTION_IDENTIFIER);
    rq.setVersion(VER_1011);

    ReadRequests readRequests = of.createOTA_ReadRQReadRequests();
    rq.setReadRequests(readRequests);

    ReadRequest readRequest = of.createOTA_ReadRQReadRequestsReadRequest();
    readRequests.getReadRequest().add(readRequest);
    UniqueID_Type uniqueID = of.createUniqueID_Type();
    readRequest.setUniqueID(uniqueID);
    uniqueID.setID(id);
    uniqueID.setID_Context(CONTEXT);
    uniqueID.setType(UIT_RESERVATION);

    return rq;
}

```

```

/*****
 * General helper functions
 * @param value as String
 * @return BigInteger value
 *****/
private static BigInteger getBigInteger(String value) {
    if (value == null) {return null;}
    else {return new BigInteger(value);}
}

/*****
 * @param value as Integer
 * @return BigInteger value
 *****/
private static BigInteger getBigInteger(Integer value) {
    if (value == null) {return null;}
    else {return new BigInteger(String.valueOf(value));}
}

/*****
 * @param value as BigInteger
 * @return Integer value
 *****/
private static Integer getInteger(BigInteger value) {
    if (value == null) {return null;}
    else {return Integer.valueOf(value.toString());}
}

/*****
 * @param value as String
 * @return BigDecimal value
 *****/
private static BigDecimal getBigDecimal(String value) {
    if (value == null) {return null;}
    else {return new BigDecimal(value);}
}

/*****
 * @param value as Double
 * @return BigDecimal value
 *****/
private static BigDecimal getBigDecimal(Double value) {
    if (value == null) {return null;}
    else {return new BigDecimal(String.valueOf(value));}
}

/*****
 * @param emailaddress as String
 * @param eat from OTA EAT code list
 * @return customer email object
 *****/
private static Email getCustomerEmail(String emailaddress, String eat){
    Email email = of.createCustomerTypeEmail();
    email.setDefaultInd(true);
    email.setEmailType(eat);
    email.setRemark(null);
    email.setRPH(null);
    email.setValue(emailaddress);
    return email;
}

/*****
 * @param text as String
 * @param language code from ISO two character code list
 * @return free text object
 *****/
private static FreeTextType getFreeText(String text, String language) {

```

```

FreeTextType freeText = of.createFreeTextType();
freeText.setLanguage(language);
freeText.setValue(text);
return freeText;
}

/*****
 * @param pos_code in context of CONTEXT (RZ for Razor) at the point of sale
 * @param context of pos_code
 * @return point of sale (POS) object
 *****/
private static POS_Type getPOS(
    String pos_code,
    String context
) {
    POS_Type POS = of.createPOS_Type();
    SourceType source = of.createSourceType();
    POS.getSource().add(source);
    RequestorID requestorID = of.createSourceTypeRequestorID();
    source.setRequestorID(requestorID);
    requestorID.setID(pos_code);
    requestorID.setID_Context(context);
    return POS;
}

/*****
 * @return current time as XML Gregorian calendar
 *****/
private static XMLGregorianCalendar getXGC () {
    return getXGC(new Date());
}

/*****
 * @param date as java.util.Date
 * @return date as XML Gregorian calendar
 *****/
private static XMLGregorianCalendar getXGC (Date date) {
    XMLGregorianCalendar xgc = null;
    try {
        xgc = DatatypeFactory.newInstance().newXMLGregorianCalendar();
        Calendar cal = Calendar.getInstance();
        cal.setTime(date);
        xgc.setDay(cal.get(Calendar.DAY_OF_MONTH));
        xgc.setMonth(cal.get(Calendar.MONTH)+1);
        xgc.setYear(cal.get(Calendar.YEAR));
        xgc.setTime(cal.get(Calendar.HOUR_OF_DAY),
            cal.get(Calendar.MINUTE),
            cal.get(Calendar.SECOND));
    } catch (DatatypeConfigurationException e) { e.printStackTrace(); }
    return xgc;
}

```

```

/*****
* Search helper functions
*****/
private static void addCardCode(Criterion criterion, String code, String cdt){
    if (criterion.getAcceptedPayments() == null){
        criterion.setAcceptedPayments(of.createAcceptedPaymentsType());
    }
    PaymentFormType paymentForm = of.createPaymentFormType();
    criterion.getAcceptedPayments().getAcceptedPayment().add(paymentForm);
    PaymentCardType paymentCard = of.createPaymentCardType();
    paymentForm.setPaymentCard(paymentCard);
    paymentCard.setCardCode(code);
    paymentCard.setCardType(cdt);
}

private static void addAcceptedPayment(Criterion criterion, String code ){
    if (criterion.getAcceptedPayments() == null){
        criterion.setAcceptedPayments(of.createAcceptedPaymentsType());
    }
    PaymentFormType paymentForm = of.createPaymentFormType();
    criterion.getAcceptedPayments().getAcceptedPayment().add(paymentForm);
    paymentForm.setRemark(code);
}

private static void addGuestCount(Criterion criterion, Integer value, String aqc) {
    GuestCountType guestCounts = getRoomStayCandidate(criterion).getGuestCounts();
    if (guestCounts == null) {
        guestCounts = of.createGuestCountType();
        getRoomStayCandidate(criterion).setGuestCounts(guestCounts);
    }
    guestCounts.setIsPerRoom(true);
    GuestCount guestCount = of.createGuestCountTypeGuestCount();
    guestCounts.getGuestCount().add(guestCount);
    guestCount.setCount(value);
    guestCount.setAgeQualifyingCode(aqc);
}

private static void addHotelAmenity(Criterion criterion, String code) {
    HotelAmenity hotelAmenity = of.createHotelSearchCriterionTypeHotelAmenity();
    criterion.getHotelAmenity().add(hotelAmenity);
    hotelAmenity.setCode(code);
}

private static HotelRef addHotelRef(Criterion criterion) {
    HotelRef hotelRef = of.createItemSearchCriterionTypeHotelRef();
    criterion.getHotelRef().add(hotelRef);
    return hotelRef;
}

private static void addRecreation(Criterion criterion, String code ){
    Criterion.Recreation recreation = of.createHotelSearchCriterionTypeRecreation();
    criterion.getRecreation().add(recreation);
    recreation.setCode(code);
}

private static void addRoomAmenity(Criterion criterion, String value ){
    RoomAmenityPrefType roomAmenityPref = of.createRoomAmenityPrefType();
    criterion.getRoomAmenity().add(roomAmenityPref);
    roomAmenityPref.setValue(value);
}

private static void addAccessibilityCode(Criterion criterion, String feature ){
    HotelFeature hotelFeature = of.createHotelSearchCriterionTypeHotelFeature();
    criterion.getHotelFeature().add(hotelFeature);
    hotelFeature.setAccessibilityCode(feature);
}

```

```

private static void addSecurityFeatureCode(Criterion criterion, String feature ){
    HotelFeature hotelFeature = of.createHotelSearchCriterionTypeHotelFeature();
    criterion.getHotelFeature().add(hotelFeature);
    hotelFeature.setSecurityFeatureCode(feature);
}

private static HotelSearchCriterionType.RateRange getRateRange(Criterion criterion) {
    if (criterion.getRateRange() == null
        || criterion.getRateRange().isEmpty()) {
        HotelSearchCriterionType.RateRange rateRange =
            of.createHotelSearchCriterionTypeRateRange();
        criterion.getRateRange().add(rateRange);
        rateRange.setCurrencyCode(DEFAULT_CURRENCY_CODE);
        rateRange.setMinRate(getBigDecimal(0.0));
    }
    return criterion.getRateRange().get(0);
}

private static RoomStayCandidateType getRoomStayCandidate(Criterion criterion) {
    RoomStayCandidates roomStayCandidates = criterion.getRoomStayCandidates();
    if (roomStayCandidates == null){
        roomStayCandidates = of.createHotelSearchCriterionTypeRoomStayCandidates();
        criterion.setRoomStayCandidates(roomStayCandidates);
        roomStayCandidates.getRoomStayCandidate()
            .add(of.createRoomStayCandidateType());
    }
    return roomStayCandidates.getRoomStayCandidate().get(0);
}
}
}

```