RemoteBookingClient Class

Supporting Files

The following files are required in the application folder:

RemoteBookingClient.config

QSLRemoteBookingInterfaceV5.dll

QSLRemoteBookingTypes.dll

Common Language Runtime

Requires CLR v2.0.50727

References

Add references to the following assemblies:

QSLRemoteBookingInterfaceV5.dll

QSLRemoteBookingTypes.dll

Test

```
Public Function Test(ByVal xmldata As String, ByVal hostname As
String, ByVal port As Integer) As String
Overloads Function Test(ByVal xmldata As String) As String
```

xmldata: xml string in the following format:

```
<testrequest>Any text</testrequest>
```

hostname: ip address or hostname of the remote computer running the remote booking service

port: port number the service is listening on - usually 48488

returns: (string)

```
<testresponse appdomain="{AppDomain.CurrentDomain.FriendlyName}">" &
xml record you sent & "</testresponse>
```

BookTable

```
Public Function BookTable(ByVal xmldata As String, ByVal hostname As
String, ByVal port As Integer) As String
```

xmldata: xml string in one of the following formats:

V5 mandatory parameters:

```
<request recordversion="5.0.0">
               <parameters locationprefix="RUL" atdate="2006-12-05"
attime="19:45" session="3" covers="4" surname="Test" title="Mr"
telephone="01628472999" productid="2" sourcesite="XXXXXXXX"
authcode="XXXXXXXX" />
</request>
```

V5 full record:

```
<request recordversion="5.0.0">
               <parameters locationprefix="RUL" atdate="2006-12-05"
attime="19:45" session="3" covers="4" surname="Test" title="Mr"
telephone="01628472999"  productid="2" sourcesite="XXXXXXXX"
authcode="XXXXXXXX" forename="" initial="" address="" address2=""
address3="" address4="" postcode="" countrycode="" email=""
creditcardno="" creditcardexp="" creditcardtype="" creditcardname=""
creditcardccv="" testmode="0" extra1="" extra2="" extra3=""
promotion="" message="" marketing="yes" email2="" fax=""
telephone2="" telephone3="" sendemail1="" sendemail2="" />
</request>
```

Note: The record must always contain the mandatory attributes, and the optionally any of the other attributes supported in the record. The parameters are explained individually at the end of the document.

hostname:	ip address or hostname of the remote computer running the remote booking service
port:	port number the service is listening on - usually 48488

returns: (string) xml in one of the following formats:

```
<response status="success" data="{Booking Reference}"/>
<response status="error" data="{Error message text}"/>
```

V5 Availability Enquiry

```
Public Function GetSessionAvailability(ByVal Prefix As String, ByVal
PlanDate As Date, ByVal Session As Integer, ByVal Covers As Integer,
ByVal UserAccessLevel As Integer, ByVal hostname As String, ByVal
port As Integer) As DataSet
```

Prefix:	3 character location prefix code (e.g. RES)
PlanDate:	Date for which availability is required
Session:	Session ID for which availability is required (1 = breakfast; 2= lunch; 3 = dinner)
Covers:	Number of covers for which availability is required
UserAccessLevel:	Access Level on which bookings will be accepted for your site (e.g. 1)

hostname:	ip address or hostname of the remote computer running the remote booking service
port:	port number the service is listening on - usually 48488

Returns: **(dataset)**

```
Table:  Availability
        ID              int             RecordID
        PlanTime        datetime        Timeslot
        Available       int             1 if available; 0 if not
        ReturnTime      datetime        Return time (null if none)
```

```
Table: SessionMessage (single row)
        ID              int             RecordID
        Message         string          Session message
```

V5 Web Config Enquiry

```
Public Function GetWebConfig(ByVal Prefix As String, ByVal hostname
As String, ByVal port As Integer) As DataSet
```

Prefix:	3 character location prefix code (e.g. RES)
hostname:	ip address or hostname of the remote computer running the remote booking service
port:	port number the service is listening on - usually 48488

Returns: **(dataset)**

```
Table:  WebConfiguration
        Prefix                  string  3 character location prefix
        MaxCovers               Int     maximum number of covers for
                                        which web bookings will be
                                        accepted
        AdvanceBookingMonths    int     number of months into the
                                        future for which web bookings
                                        will be accepted
```

XML Web Service

Test

Test that Server Component is listening

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.resv5.com/webservices/Test"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Test xmlns="http://www.resv5.com/webservices">
      <xmldata>string</xmldata>
    </Test>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TestResponse xmlns="http://www.resv5.com/webservices">
      <TestResult>string</TestResult>
    </TestResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Test xmlns="http://www.resv5.com/webservices">
      <xmldata>string</xmldata>
    </Test>
  </soap12:Body>
```

```
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <TestResponse xmlns="http://www.resv5.com/webservices">
      <TestResult>string</TestResult>
    </TestResponse>
  </soap12:Body>
</soap12:Envelope>
```

xmldata: xml string in the following format:

```
<testrequest>Any text</testrequest>
```

returns: (string)

```
<testresponse appdomain="{AppDomain.CurrentDomain.FriendlyName}">" &
xml record you sent & "</testresponse>
```

BookTable

Make a booking

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.resv5.com/webservices/BookTable"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <BookTable xmlns="http://www.resv5.com/webservices">
      <xmldata>string</xmldata>
    </BookTable>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <BookTableResponse xmlns="http://www.resv5.com/webservices">
      <BookTableResult>string</BookTableResult>
    </BookTableResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <BookTable xmlns="http://www.resv5.com/webservices">
      <xmldata>string</xmldata>
    </BookTable>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <BookTableResponse xmlns="http://www.resv5.com/webservices">
      <BookTableResult>string</BookTableResult>
    </BookTableResponse>
  </soap12:Body>
</soap12:Envelope>
```

xmldata: xml string the following format:

V5 mandatory parameters:

```
<request recordversion="5.0.0">
               <parameters locationprefix="RUL" atdate="2006-12-05"
attime="19:45" session="3" covers="4" surname="Test" title="Mr"
telephone="01628472999" productid="2" sourcesite="XXXXXXXX"
Authcode="XXXXXXXX" />
</request>
```

V5 full record:

```
<request recordversion="5.0.0">
               <parameters locationprefix="RUL" atdate="2006-12-05"
attime="19:45" session="3" covers="4" surname="Test" title="Mr"
telephone="01628472999"  productid="2" sourcesite="XXXXXXXX"
authcode="XXXXXXXX" forename="" initial="" address="" address2=""
address3="" address4="" postcode="" countrycode="" email=""
creditcardno="" creditcardexp="" creditcardtype="" creditcardname=""
creditcardccv="" testmode="0" extra1="" extra2="" extra3=""
promotion="" message="" marketing="yes" email2="" fax=""
telephone2="" telephone3="" sendemail1="" sendemail2="" />
</request>
```

Note: The record must always contain the mandatory attributes, and the optionally any of the other attributes supported in the record. The parameters are explained individually at the end of the document.

returns: (string) xml in one of the following formats:

```
<response status="success" data="{Booking Reference}"/>
<response status="error" data="{Error message text}"/>
```

GetSessionAvailability

v5 only: Gets availability for specified session

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.resv5.com/webservices/GetSessionAvailability"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetSessionAvailability xmlns="http://www.resv5.com/webservices">
      <Prefix>string</Prefix>
      <AtDate>dateTime</AtDate>
      <Session>int</Session>
      <Covers>int</Covers>
      <ProductID>int</ProductID>
    </GetSessionAvailability>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetSessionAvailabilityResponse
xmlns="http://www.resv5.com/webservices">
      <GetSessionAvailabilityResult>

<xsd:schema>schema</xsd:schema>xml</GetSessionAvailabilityResult>
    </GetSessionAvailabilityResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap12:Body>
    <GetSessionAvailability xmlns="http://www.resv5.com/webservices">
      <Prefix>string</Prefix>
      <AtDate>dateTime</AtDate>
      <Session>int</Session>
      <Covers>int</Covers>
      <ProductID>int</ProductID>
    </GetSessionAvailability>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetSessionAvailabilityResponse
xmlns="http://www.resv5.com/webservices">
      <GetSessionAvailabilityResult>

<xsd:schema>schema</xsd:schema>xml</GetSessionAvailabilityResult>
    </GetSessionAvailabilityResponse>
  </soap12:Body>
</soap12:Envelope>
```

Prefix:	3 character location prefix code (e.g. RES)
AtDate:	Date for which availability is required
Session:	Session ID for which availability is required (1 = breakfast; 2= lunch; 3 = dinner)
Covers:	Number of covers for which availability is required
ProductID:	ProductID for which bookings will be accepted for your site (e.g. 1)

Returns: **(dataset)**

```
Table:  Availability
        ID              int             RecordID
        PlanTime        datetime        Timeslot
        Available       int             1 if available; 0 if not
        ReturnTime      datetime        Return time (null if none)
```

```
Table: SessionMessage (single row)
        ID              int             RecordID
        Message         string          Session message
```

GetWebConfig

v5 only: Gets web booking configuration information

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.resv5.com/webservices/GetWebConfig"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetWebConfig xmlns="http://www.resv5.com/webservices">
      <Prefix>string</Prefix>
    </GetWebConfig>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetWebConfigResponse xmlns="http://www.resv5.com/webservices">
      <GetWebConfigResult>
        <xsd:schema>schema</xsd:schema>xml</GetWebConfigResult>
    </GetWebConfigResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /qslwebservice/QSLWebBooking.asmx HTTP/1.1
Host: qslwebsrv
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetWebConfig xmlns="http://www.resv5.com/webservices">
```

```
      <Prefix>string</Prefix>
    </GetWebConfig>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetWebConfigResponse xmlns="http://www.resv5.com/webservices">
      <GetWebConfigResult>
        <xsd:schema>schema</xsd:schema>xml</GetWebConfigResult>
    </GetWebConfigResponse>
  </soap12:Body>
</soap12:Envelope>
```

Prefix: 3 character location prefix code (e.g. RES)

Returns: (dataset)

```
Table:  WebConfiguration
        Prefix                  string  3 character location prefix
        MaxCovers               Int     maximum number of covers for
                                        which web bookings will be
                                        accepted
        AdvanceBookingMonths    int     number of months into the
                                        future for which web bookings
                                        will be accepted
```

Booking Request Parameters

locationprefix:	3 Character RES Prefix
atdate:	Booking Date in yyyy-MM-dd format
attime:	Booking Time in hh:mm format (24 hour)
session:	Session ID per RES session table
covers:	Number of covers for booking
surname:	Last Name
title:	Title
telephone:	Primary telephone number
productid:	Access Level for Web Bookings (1-99)
sourcesite:	Booking Source - automatically added to RES if it does not exist in Listing table
authcode:	Authcode for bookings (per QuadraNet)
forename:	First Name
initial:	Initial
address:	Address
address2:	Address
address3:	Address
address4:	Address
postcode:	Post/Zip Code
countrycode:	International ISO Country Code
email:	Primary email address
creditcardno:	Card Number
creditcardexp:	Expiry Date
creditcardtype:	Type per Card types table in RES e.g. VISA
creditcardname:	Card Holder Name
creditcardccv:	Security Code
testmode:	If testmode="1" is submitted booking is automatically deleted immediately after being made.
extra1:	Spare field
extra2:	Spare field
extra3:	Spare field
promotion:	Promotion Code - automatically added to RES if it does not exist in promotion table
message:	Booking message (text up to 4000 characters)
marketing:	if marketing="yes" is submitted then the opt-in correspondence flags are set for any contact fields included in the message
email2:	Additional email address
fax:	Fax number
telephone2:	Additional telephone number
telephone3:	Additional telephone number

sendemail1:	ID of Outbox document to send to email address in email parameter. This is the ID from the OutboxDocument table. The specified document is then sent via Outbox.
sendemail2:	ID of Outbox document to send to email address in email2 parameter. This is the ID from the OutboxDocument table. The specified document is then sent via Outbox