

InTour XML Availability

Common Information

Service provides remote programming interface to InTour Office system. It allows you to query InTour Office allotments and stop sales.

Protocols

Service can be accessed by either of two protocols:

1. SOAP over HTTP (standard web-service protocol);
2. Plain XML over HTTP.

To help you create wrapper classes for SOAP service provides its WSDL description.

When using plain XML you should send XML requests in HTTP POST body UTF-8 encoded. Response will also be UTF-8 encoded. Here is an example of plain XML communication method in C# language (contains error logging to console):

```
static string SendHttpRequest ( string xml ) {  
    try {  
        var rq = HttpRequest.Create ( "http://zelsoft.ru/InTourXml_v2/BookingService.ashx" );  
        rq.Method = "POST";  
        using ( var sw = new StreamWriter ( rq.GetRequestStream ( ), Encoding.UTF8 ) )  
            sw.Write ( xml );  
        using ( var sr = new StreamReader ( rq.GetResponse ( ).GetResponseStream ( ), Encoding.UTF8 ) )  
            return sr.ReadToEnd ( );  
    }  
    catch ( WebException ex ) {  
        using ( var sr = new StreamReader ( ex.Response.GetResponseStream ( ), Encoding.UTF8 ) ) {  
            Console.WriteLine ( sr.ReadToEnd ( ) );  
        }  
        throw;  
    }  
}
```

Error Handling

There are two error types:

- Handled by service. It depends on request you send and explained in response structures below.
- Unhandled by service. It follows protocols standard error handling mechanisms: HTTP error code plus error description in response body.

Data Types

Service uses following primitive data types:

- String – text string.
- Date – date. Format: yyyy-MM-dd (e.g. 2013-12-23).

- DateTime – date with time. Format: yyyy-MM-ddTHH:mm:ss(.f) (e.g. 2013-12-23T17:15:23.432).
- Int – integer.
- Decimal – decimal number (e.g. 1234.5).
- Boolean – “true” or “false”.

Square brackets “[]” used in this documentation denote array of items (e.g. Int[]).

Web Pages

Service web application root contains following pages:

- BookingService.asmx – SOAP handling page (e.g. http://zelsoft.ru/intourxml_v2/BookingService.asmx). It also provides WSDL description if queried with “?WSDL” parameter (e.g. http://zelsoft.ru/intourxml_v2/BookingService.asmx?WSDL).
- BookingService.ashx – plain XML handling page. (e.g. http://zelsoft.ru/intourxml_v2/BookingService.ashx).
- SendRequest.aspx – helper browser XML testing page (e.g. http://zelsoft.ru/intourxml_v2/SendRequest.aspx).

Requests

Requests and responses have the same structure unrelated to protocol you choose.

All requests must have Login and Password parameters specified. You should ask service owner for login\password pair.

Parameters not marked as mandatory can be omitted from XML request or left with their default values when using SOAP wrapper classes.

SOAP examples below imply you have service wrapper “svc” defined (something like “var svc = new BS.BookingService ();” in C#).

Quotas and Stop Sales

You can query quotas and stop sales with **GetQuota** method.

Request

Parameter	Data Type	Mandatory	Description
UpdatedFrom	DateTime	No	Minimal item last update time. All items will be returned If parameter is not specified.
ReturnQuotas	Boolean	No	Indicates service must return quotas
ReturnStopSales	Boolean	No	Indicates service must return stop sales

Response

Contains array of Hotel elements with following structure:

Field	Description
Id	Hotel identifier
Code	Hotel code
Name	Hotel name
Quotas StopSales	Array of hotel quotas and stop sales with following structure
RcId	Room category identifier (if specified)
RcCode	Room category code (if specified)
RcName	Room category name (if specified)
RtId	Room type identifier (if specified)
RtCode	Room type code (if specified)
RtName	Room type name (if specified)
ReleasePeriod	Quota release period (if specified)
QuotaValue	Quota value (if specified)
DateBegin	Item period begin
DateEnd	Item period end
UpdDate	Last update time
QuotaType	Quota type: Allotment or Commitment
State	State of stop sale ("Actual" or "Cancelled")

SOAP sample request:

```
var quotas = svc.GetQuota ( new BS.GetQuotaRq
    {
        Login = "Zelsoft",
        Password = "zell23",
        ReturnQuotas = true,
        ReturnStopSales = true
    } );
```

XML sample request:

```
<GetQuotaRq>
  <Login>Zelsoft</Login>
  <Password>zell23</Password>
  <UpdatedFrom>2013-10-28T19:19:01</UpdatedFrom>
  <ReturnQuotas>true</ReturnQuotas>
```

```
</ReturnStopSales>true</ReturnStopSales>
</GetQuotaRq>
```

XML sample response:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetQuotaRs>
<Hotels>
  <Hotel Id="1013" Code="ANAB" Name="Anabel">

    <Quotas>
      <QuotaItem ReleasePeriod="7" QuotaValue="12" DateBegin="2013-10-19T00:00:00" DateEnd="2013-11-30T00:00:00"
      UpdDate="2013-10-28T19:19:28.2" QuotaType="Allotment" />
      <QuotaItem RcId="19" RcCode="SUP" RcName="Superior" ReleasePeriod="7" QuotaValue="2" DateBegin="2013-11-
      01T00:00:00" DateEnd="2013-11-30T00:00:00" UpdDate="2013-10-28T19:19:28.2" QuotaType="Allotment" />
    </Quotas>

    <StopSales>
      <QuotaItem DateBegin="2013-11-05T00:00:00" DateEnd="2013-11-12T00:00:00" UpdDate="2013-10-28T19:19:28.187"
      QuotaType="Allotment" State="Actual" />
      <QuotaItem RcId="19" RcCode="SUP" RcName="Superior" DateBegin="2013-11-05T00:00:00" DateEnd="2013-11-
      12T00:00:00" UpdDate="2013-10-28T19:19:28.187" QuotaType="Allotment" State="Actual" />
    </StopSales>

  </Hotel>
</Hotels>
</GetQuotaRs>
```