



New Logic Booking API

Developer's guide

v.3.1 (30.12.2020)

Language – English

Copyright © 2012 - 2021 – New Logic

Annotation

This API makes it possible to carry out the following operations :

- getting static data (list of countries, regions, cities, hotels, rooms, hotel classes, room types, room categories, types of placement, types of food, excursions, transfer facilities, currencies, order statuses)
- search of hotels
- getting prices for early check-in and late check-out
- search of excursions
- search of transfer facilities
- ordering
- search of orders
- getting information about the order
- order modification
- order confirmation
- order cancellation

New Logic Booking API

Version	Date	Description	Pages
1.0	24.01.2012	First release of the guide, product's update.	36
1.1	07.02.2013	Added display of price of penalties for change and cancellation order when getting order information.	37
1.2	25.02.2013	Added price detailing for each day by room when search of hotels.	38
1.3	02.04.2013	Added display of rules of booking and cancellation, date of penalties, penalty amount in case of cancellation when search of content for the order. Added cancellation policy when getting information about the order.	38
1.4	04.04.2013	Added additional options when getting information about the types of placement. Added display of room types when search of hotels.	38
1.5	20.06.2013	Added display of hotel address and hotel coordinates when getting the list of hotels.	38
1.6	22.08.2013	Integrated special action tariffs. Changed the structure of search of hotels, adding and editing orders, getting order information.	41
1.7	11.09.2013	Added the information about the order status to the GetOrderInfo method.	42
1.8	11.10.2013	Added the possibility to search by hotel.	42
1.9	17.10.2013	Added the possibility to search rooms by availability status to the SearchHotel method.	42
2.0	13.01.2014	Added the possibility to search rooms without detailing prices for days to the method SearchHotel. It is possible to get only a recommended variant from the hotel.	42
2.1	18.02.2014	The following parameters were added to the response of method HotelList: class of the hotel, time of arrival and time of departure.	42
2.2	04.02.2014	Added description of the method to get the list of hotel classes HotelClassList.	43
2.3	05.08.2014	Added the possibility to get the number of available allotments in method SearchHotel. The request is the same, but the response is containing one more parameter.	45
2.4	13.05.2015	Added method for search orders by date of booking and the hotel SearchOrders. Updated guide: - formatted text - added information about the available languages - in response examples of method SearchHotelRS removed attributes «id», «href» in elements «Access», «BookingPolicyInfo», «CancellationPolicyInfo» (typo in the documentation) - in response example of method SearchExcursionRS changed date format in element «DateExcursion» (typo in the documentation)	35

		<ul style="list-style-type: none"> - in description of elements method RoomListRS changed name of element «food type» – «FoodType» (was «FoodTypes» – typo in the documentation) - in request example of method AddOrderRQ changed name of method (was «AddOrderRequest» – typo in the documentation) - in response example of method AddOrderRS changed name of method (was «AddOrderResponse» – typo in the documentation) - added description of new method SearchOrders - in description of elements method EditOrderRQ removed attributes «id», «name» for element «RoomActionTariff» (typo in the documentation) - in response example of method EditOrderRS changed name of method (was «AddOrderResponse» – typo in the documentation) 	
2.5	08.07.2015	Added method for order confirmation ConfirmOrder.	36
2.6	25.08.2015	Added method for getting prices for early check-in and late check-out GetOvertime. In methods AddOrderRQ and EditOrderRQ in element «Option» added attribute «TypeEarlyCheckIn».	39
2.7	14.11.2018	API renamed to «New Logic Booking API». Updated guide: - in request example of method TransferListRQ changed name of method (was «AddServiceListRQ» – typo in the documentation)	40
2.8	04.07.2019	Added method for getting hotel surcharges GetHotelSurcharges. In response of method GetOrderInfoRS added element «HotelSurcharge». Added method for simulating order modification SimulateEditOrder.	42
2.9	03.12.2019	Added method RoomCategoryList. In response of methods RoomListRS and SearchHotelRS added element «RoomCategory».	42
3.0	01.12.2020	<p>General changes:</p> <p>Changed the URL for connecting to the API. The «Auth» element has been replaced by the «Credentials» element. The «lang» attribute has been replaced by the «Language» element. Method PlacementList is replaced by method PlacementTypeList. Method AddServiceList is replaced by method ExcursionList. Method StatusList is replaced by method OrderStatusList. Method SearchHotel is renamed to SearchHotels. Method GetOvertime is renamed to GetHotelOvertime. Method SearchExcursion is renamed to SearchExcursions.</p>	53

	<p>Method SearchTransfer is renamed to SearchTransfers. Method GetOrderInfo is renamed to GetOrder.</p> <p>Method RegionList: <i>Request:</i> - the «country» attribute is replaced by the «Country» element</p> <p>Method CityList: <i>Request:</i> - the «country» attribute is replaced by the «Country» element - the «region» attribute is replaced by the «Region» element</p> <p>Method HotelList: <i>Request:</i> - the «city» attribute is replaced by the «City» element <i>Response:</i> - element «HotelItem» is renamed to «Hotel» (typo in the documentation) - in the «Hotel» element the «class» attribute is deleted, and the «HotelClass» element has been added instead - a set of «Photo» elements were moved to the «Photos» element - a set of «Hotelservice» elements were moved to the «Hotelservices» element</p> <p>Method RoomList: <i>Request:</i> - the «hotel» attribute is replaced by the «Hotel» element <i>Response:</i> - element «Placement» is replaced by element «PlacementType», in which the name of the type of placement is returned in the specified language - a set of «Photo» elements were moved to the «Photos» element - a set of «PlacementType» elements were moved to the «PlacementTypes» element - a set of «FoodType» elements were moved to the «FoodTypes» element</p> <p>Method ExcursionList: <i>Request:</i> - the «city» attribute is replaced by the «City» element <i>Response:</i> - a set of «Photo» elements were moved to the «Photos» element</p> <p>Method TransferList: <i>Request:</i> - the «city» attribute is replaced by the «City» element <i>Response:</i> - a set of «Photo» elements were moved to the «Photos»</p>	
--	---	--

	<p>element</p> <p>Method SearchHotels:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the elements «Currency», «City», «Hotel» the attribute «id» is deleted, the identifier value is now contained inside the elements - the «Date» element is renamed to «DateRange» - the «Adult» element is renamed to «Adults» - in the «Children» element added «count» attribute - the «Available» element is renamed to «Availability» <p><i>Response:</i></p> <ul style="list-style-type: none"> - in container of the room «Room» the «Placement» element is replaced by the «PlacementType» element - in container of the room «Room» the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - in the «RoomType» and «RoomCategory» elements the «name» attribute is deleted - the «Access» element is replaced by the «Availability» element <p>Method GetHotelOvertime:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the «Currency» element the attribute «id» is deleted, the identifier value is now contained inside the element - the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - the «Placement» element is replaced by the «PlacementType» element - the «Date» element is renamed to «DateRange» <p><i>Response:</i></p> <ul style="list-style-type: none"> - the «Overtime» element is renamed to «HotelOvertime» <p>Метод GetHotelSurcharges:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the elements «Currency» and «Hotel» the attribute «id» is deleted, the identifier value is now contained inside the elements <p>Method SearchExcursions:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the elements «Currency» and «City» the attribute «id» is deleted, the identifier value is now contained inside the elements - the «Date» element is renamed to «DateRange» - the «Adult» element is renamed to «Adults» <p>Method SearchTransfers:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the «Currency» and the «City» elements the «id» attribute is deleted, the identifier value is now contained inside the elements - the «Adult» element is renamed to «Adults» 	
--	---	--

	<p>Method AddOrder:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the «Hotel», «Excursion» and «Transfer» elements the «Date» element is renamed to «DateRange» - in container of the room «Room» the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - in container of the room «Room» the «Placement» element is replaced by the «PlacementType» element - in the «Amount» element the «Option» element is deleted, and the «EarlyCheckIn», the «LateCheckOut» and the «ClientNote» elements has been added instead - in containers of the excursion «Excursion» and the transfer «Transfer» the «MettingPlace» element is renamed to «MeetingPlace» <p><i>Response:</i></p> <ul style="list-style-type: none"> - in the «Order» element the «Type» attribute is deleted <p>Method SearchOrders:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the «DateFrom» and the «DateTo» elements the «date» attribute is deleted, the date value is now contained inside the elements - in the «Hotel» element the «id» attribute is deleted, the identifier value is now contained inside the element <p><i>Response:</i></p> <ul style="list-style-type: none"> - the «DateUpdate» element now contains date and time value - in the «Order» element the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements has been added <p>Method GetOrder:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - in the «Order» element the «Type» attribute is deleted, information can be obtained only for the entire general order <p><i>Response:</i></p> <ul style="list-style-type: none"> - the «GeneralOrder» element is renamed to «Order» - in the «Order» element the «currency», «TotalPrice» and «OrderStatus» attributes are deleted, and the «Currency», «TotalPrice» and «OrderStatus» elements has been added instead respectively - the «OrderHotels» element is renamed to «OrderHotel» - the «OrderExcursions» element is renamed to «OrderExcursion» - the «OrderTransfers» element is renamed to «OrderTransfer» - in the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements the «PenaltyEdit» and «PenaltyCancel» attributes are deleted, and the 	
--	--	--

	<p>«PenaltyEdit» and «PenaltyCancel» elements has been added instead respectively</p> <ul style="list-style-type: none"> - in the «OrderHotel», «OrderExcursion» and «OrderTransfer» the «OrderStatus» and «Price» elements has been added - in the «Hotel», «Excursion» and «Transfer» elements the «Date» element is renamed to «DateRange» - in container of the room «Room» the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - in container of the room «Room» the «Placement» element is replaced by the «PlacementType» element - in the «Amount» element the «Option» element is deleted, and the «EarlyCheckIn», the «LateCheckOut» and the «ClientNote» elements has been added instead - in containers of the excursion «Excursion» and the transfer «Transfer» the «MettingPlace» element is renamed to «MeetingPlace» <p>Method EditOrder:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - the «UniqueID» element is deleted, and the «Order» element with the «UniqueID» attribute has been added instead - in the «Hotel», «Excursion» and «Transfer» elements the «Date» element is renamed to «DateRange» - in container of the room «Room» the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - in container of the room «Room» the «Placement» element is replaced by the «PlacementType» element - in the «Amount» element the «Option» element is deleted, and the «EarlyCheckIn», the «LateCheckOut» and the «ClientNote» elements has been added instead - in containers of the excursion «Excursion» and the transfer «Transfer» the «MettingPlace» element is renamed to «MeetingPlace» <p><i>Response:</i></p> <ul style="list-style-type: none"> - in the «Order» element the «Type» attribute is deleted <p>Method SimulateEditOrder:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - the «UniqueID» element is deleted, and the «Order» element with the «UniqueID» attribute has been added instead - in the «Hotel» element the «Date» element is renamed to «DateRange» - in container of the room «Room» the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - in container of the room «Room» the «Placement» element is replaced by the «PlacementType» element - in the «Amount» element the «Option» element is 	
--	---	--

	<p>deleted, and the «EarlyCheckIn», the «LateCheckOut» and the «ClientNote» elements has been added instead</p> <p><i>Response:</i></p> <ul style="list-style-type: none"> - the «GeneralOrder» element is renamed to «Order» - in the «Order» element the «currency», «TotalPrice» and «OrderStatus» attributes are deleted, and the «Currency», «TotalPrice» and «OrderStatus» elements has been added instead respectively - the «OrderHotels» element is renamed to «OrderHotel» - in the «OrderHotel» element the «PenaltyEdit» and «PenaltyCancel» attributes are deleted, and the «PenaltyEdit» and «PenaltyCancel» elements has been added instead respectively - in the «OrderHotel» element the «OrderStatus» and «Price» elements has been added - in the «Hotel» element the «Date» element is renamed to «DateRange» - in container of the room «Room» the «RoomActionTariff» element is replaced by the «HotelRatePlan» element - in container of the room «Room» the «Placement» element is replaced by the «PlacementType» element - in the «Amount» element the «Option» element is deleted, and the «EarlyCheckIn», the «LateCheckOut» and the «ClientNote» elements has been added instead <p>Method ConfirmOrder:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - it is possible to confirm only one general order or several unit orders of one general order - in the «Order» element the «Type» attribute is deleted - in the «Order» element the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements has been added <p><i>Response:</i></p> <ul style="list-style-type: none"> - in the «Order» element the «Type» attribute is deleted - in the «Order» element the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements has been added <p>Method CancelOrder:</p> <p><i>Request:</i></p> <ul style="list-style-type: none"> - it is possible to cancel only one general order or several unit orders of one general order - in the «Order» element the «Type» attribute is deleted - in the «Order» element the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements has been added <p><i>Response:</i></p> <ul style="list-style-type: none"> - in the «Order» element the «Type» attribute is deleted - in the «Order» element the «OrderHotel», 	
--	---	--

New Logic Booking API

		«OrderExcursion» and «OrderTransfer» elements has been added	
3.1	30.12.2020	Method SearchHotels: <i>Request:</i> - the «OneRoom», «Page», «PerPage» elements has been added <i>Response:</i> - the «Page», «TotalPages» elements has been added	54

Table of contents

1	General information	12
2	Getting static data.....	13
2.1	CountryList – getting the list of countries	13
2.2	RegionList – getting the list of regions.....	13
2.3	CityList – getting the list of cities.....	14
2.4	HotelList – getting the list of hotels.....	15
2.5	RoomList – getting the list of hotel rooms	16
2.6	HotelClassList – getting the list of hotel classes.....	18
2.7	RoomTypeList – getting the list of hotel room types.....	18
2.8	RoomCategoryList – getting the list of hotel room categories	19
2.9	PlacementTypeList – getting the list of placement types	20
2.10	FoodTypeList – getting the list of food types.....	21
2.11	ExcursionList – getting the list of excursions	22
2.12	TransferList – getting the list of transfers	23
2.13	CurrencyList – getting the list of currencies	24
2.14	OrderStatusList – getting the list of order statuses	25
3	Search of content for order.....	26
3.1	SearchHotels – search of hotels	26
3.2	GetHotelOvertime – getting prices for early check-in and late check-out of the hotel	31
3.3	GetHotelSurcharges – getting the list of hotel surcharges.....	34
3.4	SearchExcursions – search of excursions	35
3.5	SearchTransfers – search of transfers.....	36
4	Operations with orders.....	38
4.1	AddOrder – ordering.....	38
4.2	SearchOrders – search of orders	40
4.3	GetOrder – getting information about the order.....	42
4.4	EditOrder – order modification.....	45
4.5	SimulateEditOrder – simulating order modification.....	47
4.6	ConfirmOrder – order confirmation	49
4.7	CancelOrder – order cancellation.....	51
5	Contacts	54

1 General information

Client-server data exchange is carried out by the SOAP protocol. Every user receives login (email), password and a range of IP addresses for connection with the server. A mandatory element in any request is the <Credentials>, which includes login and password.

In order to decrease the size of the transferred XML files, the replacement of the static data with the corresponding ID is used. For example, for receiving a list of all hotels in a specified city, it is necessary to set not the city name but it's ID. The process of getting all static data is described in details in the corresponding parts of this document.

To connect to the API version 3, you must use the following URL:

<http://api.newlogic.ua/api/v3/wSDL/> – URL of wsdl-schema for production server

<http://apitest.newlogic.ua/api/v3/wSDL/> – URL of wsdl-schema for test server

To obtain text information in the required language, it is necessary to specify the <Language> element in each request. Available languages:

- ua – Ukrainian
- en – English
- ru – Russian

Example for initialization of service and getting the list of countries (programming language is PHP):

```
$client = new SoapClient('http://api.newlogic.ua/api/v3/wSDL/');
$params = array(
    'Credentials' => array(
        'Login' => 'test@test.com',
        'Password' => '***'
    ),
    'Language' => 'en'
);
$RS = $client->GetCountryList($params);
```

2 Getting static data

This part explains how to get necessary static information about countries, regions, cities, hotels, excursions, transfers and etc.

2.1 CountryList – getting the list of countries

Description:

Getting all active countries. The country has attributes:

- id – unique country identifier
- name – country name on the specified language

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CountryListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </CountryListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CountryListRS>
      <Countries>
        <Country id="30" name="Ukraine"/>
      </Countries>
    </CountryListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.2 RegionList – getting the list of regions

Description:

Getting regions of the specified country. The region has attributes:

- id – unique region identifier
- name – region name on the specified language

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RegionListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Country>30</Country>
    </RegionListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RegionListRS>
      <Regions>
        <Region id="47" name="Crimea"/>
        <Region id="46" name="Dnipropetrovsk"/>
        <Region id="44" name="Donetsk"/>
      </Regions>
    </RegionListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Provided above example demonstrates selection of all regions of the country with «id» – 30. The language of representation is «en».

2.3 CityList – getting the list of cities

Description:

Getting cities from the region or the country. The city has attributes:

- id – unique city identifier
- name – city name on the specified language

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CityListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </CityListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<Country>30</Country>
<Region>38</Region>
</CityListRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CityListRS>
      <Cities>
        <City id="8912" name="Bila Tserkva"/>
        <City id="29" name="Kiev"/>
      </Cities>
    </CityListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Provided above example demonstrates selection of all cities from the country with «id» – 30 and the region with «id» – 38. The language of selection is «en».

2.4 HotelList – getting the list of hotels

Description:

Getting hotels from selected city. Hotel has attributes:

- id – unique hotel identifier
- name – hotel name on the specified language

Elements:

- HotelClass – identifier and name of hotel class. List of all hotel classes can be obtained using the «HotelClassList» method
- TimeArrival – check-in date
- TimeDeparture – check-out date
- Description – hotel description on the specified language
- Address – hotel address on the specified language
- Latitude – hotel’s geographic coordinates (latitude)
- Longitude – hotel’s geographic coordinates (longitude)
- Photos – container of hotel’s photos
- Photo – hotel’s photo
- Hotelservices – container of services, provided in the hotel
- HotelService – list of services, provided in the hotel

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<SOAP-ENV:Body>
  <HotelListRQ>
    <Credentials>
      <Login>test@test.com</Login>
      <Password>***</Password>
    </Credentials>
    <Language>en</Language>
    <City>40</City>
  </HotelListRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <HotelListRS>
      <Hotels>
        <Hotel id="1" name="Hotel name">
          <HotelClass id="11" name="4*" />
          <TimeArrival>14:00</TimeArrival>
          <TimeDeparture>12:00</TimeDeparture>
          <Description>Some description.</Description>
          <Address>9, Minskaya Str.</Address>
          <Latitude>47.865961451122</Latitude>
          <Longitude>35.054122209549</Longitude>
          <Photos>
            <Photo>http://url/to/photo/1.jpg</Photo>
            <Photo>http://url/to/photo/2.jpg</Photo>
            <Photo>http://url/to/photo/3.jpg</Photo>
          </Photos>
          <Hotelservices>
            <Hotelservice>24 Hour Room Service</Hotelservice>
            <Hotelservice>Wi-Fi internet access</Hotelservice>
            <Hotelservice>Parking</Hotelservice>
          </Hotelservices>
        </Hotel>
      </Hotels>
    </HotelListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

This example demonstrates the selection of hotels in the city with «id» – 40. The language is «en».

2.5 RoomList - getting the list of hotel rooms

Description:

Getting rooms from the selected hotel. Room has attributes:

- id – unique room identifier
- name – room name on the specified language

Elements:

- RoomType – identifier and name of room type
- RoomCategory – identifier and name of room category
- Description – room description on the specified language
- Photos – container of room's photos
- Photo – room's photo
- PlacementTypes – container of available room placement types
- PlacementType – identifier and name of placement type on the specified language. List of all placement types can be obtained using the «PlacementTypeList» method
- FoodTypes – container of available room food types
- FoodType – identifier and name of food type. List of all food types can be obtained using the «FoodTypeList» method

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RoomListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Hotel>80</Hotel>
    </RoomListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RoomListRS>
      <Rooms>
        <Room id="36" name="Double Economy">
          <RoomType id="9" name="DBL/TWN"/>
          <RoomCategory id="1" name="Standard"/>
          <Description>Some description.</Description>
          <Photos>
            <Photo>http://url/to/photo/1.jpg</Photo>
            <Photo>http://url/to/photo/2.jpg</Photo>
            <Photo>http://url/to/photo/3.jpg</Photo>
          </Photos>
          <PlacementTypes>
            <PlacementType id="1" name="1"/>
            <PlacementType id="2" name="1 + 1 adult"/>
            <PlacementType id="3" name="1 + 1 child"/>
            <PlacementType id="4" name="2"/>
          </PlacementTypes>
          <FoodTypes>
            <FoodType id="15" name="BB"/>
          </FoodTypes>
        </Room>
      </Rooms>
    </RoomListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
</Room>
</Rooms>
</RoomListRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

This example demonstrates the selection of hotel rooms in the hotel with «id» – 80. The language is «en».

2.6 HotelClassList – getting the list of hotel classes

Description:

Getting hotel classes. Hotel class has attributes:

- id – unique hotel class identifier
- name – hotel class name

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <HotelClassListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </HotelClassListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <HotelClassListRS>
      <HotelClasses>
        <HotelClass id="15" name="1 *"/>
        <HotelClass id="13" name="2 *"/>
        <HotelClass id="10" name="3 *"/>
        <HotelClass id="11" name="4 *"/>
        <HotelClass id="12" name="5 *"/>
      </HotelClasses>
    </HotelClassListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.7 RoomTypeList – getting the list of hotel room types

Description:

Getting room types. Room type has attributes:

- id – unique room type identifier
- name – room type name on the specified language

Request example:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RoomTypeListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </RoomTypeListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RoomTypeListRS>
      <RoomTypes>
        <RoomType id="7" name="SGL"/>
        <RoomType id="8" name="DBL"/>
        <RoomType id="9" name="DBL/TWN"/>
      </RoomTypes>
    </RoomTypeListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.8 RoomCategoryList – getting the list of hotel room categories

Description:

Getting room categories. Room category has attributes:

- id – unique room category identifier
- name – room category name on the specified language

Request example:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RoomCategoryListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </RoomCategoryListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

    </RoomCategoryListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <RoomCategoryListRS>
      <RoomCategories>
        <RoomCategory id="1" name="Standard"/>
        <RoomCategory id="2" name="Superior"/>
        <RoomCategory id="3" name="Suite"/>
      </RoomCategories>
    </RoomCategoryListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

2.9 PlacementTypeList – getting the list of placement types**Description:**

Getting types of placement. Type of placement has attributes:

- id – unique placement type identifier
- name – placement type name on the specified language

Элементы:

- MainAdults – number of main adults for this type of placement
- AdditionalAdults – number of additional adults for this type of placement
- AdditionalChildren – number of additional children for this type of placement

Request example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <PlacementTypeListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </PlacementTypeListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>

```

```

<PlacementTypeListRS>
  <PlacementTypes>
    <PlacementType id="1" name="1">
      <MainAdults>1</MainAdults>
      <AdditionalAdults>0</AdditionalAdults>
      <AdditionalChildren>0</AdditionalChildren>
    </PlacementType>
    <PlacementType id="2" name="1 + 1 adult">
      <MainAdults>1</MainAdults>
      <AdditionalAdults>1</AdditionalAdults>
      <AdditionalChildren>0</AdditionalChildren>
    </PlacementType>
    <PlacementType id="3" name="1 + 1 child">
      <MainAdults>1</MainAdults>
      <AdditionalAdults>0</AdditionalAdults>
      <AdditionalChildren>1</AdditionalChildren>
    </PlacementType>
  </PlacementTypes>
</PlacementTypeListRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

2.10 FoodTypeList – getting the list of food types

Description:

Getting types of food. Type of food has attributes:

- id – unique food type identifier
- name – food type name on the specified language

Elements:

- Description – food type description on the specified language

Request example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <FoodTypeListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </FoodTypeListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>

```

```

<FoodTypeListRS>
  <FoodTypes>
    <FoodType id="14" name="RO">
      <Description>Room only</Description>
    </FoodType>
    <FoodType id="15" name="BB">
      <Description>Bed and breakfast</Description>
    </FoodType>
    <FoodType id="19" name="HB">
      <Description>Half board</Description>
    </FoodType>
  </FoodTypeListRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

2.11 ExcursionList – getting the list of excursions

Description:

Getting excursions from selected city. Excursion has attributes:

- id – unique excursion identifier
- name – excursion name on the specified language

Elements:

- Description – excursion description on the specified language
- Photos – container of excursion's photos
- Photo – excursion's photo

Request example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ExcursionListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <City>29</City>
    </ExcursionListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ExcursionListRS>
      <Excursions>
        <Excursion id="66" name="Excursion">
          <Description>Some description.</Description>

```

```

    <Photos>
      <Photo>http://url/to/photo/1.jpg</Photo>
      <Photo>http://url/to/photo/2.jpg</Photo>
      <Photo>http://url/to/photo/3.jpg</Photo>
    </Photos>
  </Excursion>
</Excursions>
</ExcursionListRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

This example demonstrates the selection of excursions in the city with «id» – 29. The language is «en».

2.12 TransferList – getting the list of transfers

Description:

Getting transfers from selected city. Transfer has attributes:

- id – unique transfer identifier
- name – transfer name on the specified language

Elements:

- Description – transfer description on the specified language
- Photos – container of transfer's photos
- Photo – transfer's photo

Request example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <TransferListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <City>29</City>
    </TransferListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <TransferListRS>
      <Transfers>
        <Transfer id="38" name="Transfer">

```

```
<Description>Some description.</Description>
<Photos>
  <Photo>http://url/to/photo/1.jpg</Photo>
  <Photo>http://url/to/photo/2.jpg</Photo>
  <Photo>http://url/to/photo/3.jpg</Photo>
</Photos>
</Transfer>
</Transfers>
</TransferListRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

This example demonstrates the selection of transfers in the city with «id» – 29. The language is «en».

2.13 CurrencyList – getting the list of currencies

Description:

Getting currencies. Currency has attributes:

- id – unique currency identifier
- name – currency name

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CurrencyListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </CurrencyListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CurrencyListRS>
      <Currencies>
        <Currency id="17" name="UAH"/>
        <Currency id="18" name="USD"/>
      </Currencies>
    </CurrencyListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


2.14 OrderStatusList – getting the list of order statuses

Description:

Getting statuses of order. Order status has attributes:

- id – unique order status identifier
- name – order status name

Elements:

- Description – order status description on the specified language

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <OrderStatusListRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
    </OrderStatusListRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <OrderStatusListRS>
      <OrderStatuses>
        <OrderStatus id="3" name="modified">
          <Description>Order modified. Need reconfirmation by provider.</Description>
        </OrderStatus>
      </OrderStatuses>
    </OrderStatusListRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3 Search of content for order

3.1 SearchHotels – search of hotels

Description:

Search of hotels on the set parameters. Request elements (search parameters):

- Currency – currency identifier
- City – city identifier
- Hotel – hotel identifier
- DateRange – period of stay. Attributes:
 - from – check-in date
 - to – check-out date
- Adults – residents adults. Attributes:
 - count – number of adults
- Children – residents children. Attributes:
 - count – number of children
- Child – information about resident child. Number of the Child elements must be equal to value of the «count» attribute of the Children element. Attributes:
 - age – age of the child
- Availability – availability of the hotel rooms. Valid values:
 - online – available only for instant booking
 - onrequest – available only on request
 - all – all hotel rooms
- PriceDetail – detailed price for each day. This option is disabled by default. Valid values:
 - 0 – disabled detailed price for each day. If detailed price for each day is off, in the response comes only total price for each room.
 - 1 – enabled detailed price for each day. If detailed price for each day is on, in the response comes total price for each room as well as detailed daily price information.
- OneRoom – get one room or all rooms of the hotel. This option is disabled by default. Valid values:
 - 0 – option is disabled. If the option is disabled, then all hotel rooms with all rate plans are received in the response.
 - 1 – option is enabled. If the option is enabled, then only one hotel room with all rate plans is received in the response. As a rule, it is the cheapest room that matches the search parameters.
- Page – page number. This parameter allows you to specify the page number for the "page" (portionwise) retrieval of the list of hotels by the specified search parameters. If the parameter is not specified in the request or the value is "0", then the response will contain all search results. To get results in several portions ("pages"), you need to specify the value "1", then send another request with the same parameters, but specify the value "2" in the Page parameter, then another request with the value "3" and so on until all the search results are obtained. The number of pages in the result is specified in each response in the TotalPages parameter.
- PerPage – number of hotels on the page. Allows you to manage the number of hotels on each page in the search results. Optional parameter. If the parameter is not specified or the value "0" is specified, then each page will contain 10 hotels. If the previous Page

parameter is not specified, then the PerPage parameter is ignored, even if specified in the request.

Note: options PriceDetail, OneRoom, Page, PerPage allow you to control the amount of data received and reduce the processing time of requests. For example, a city search (with a city ID but no hotel ID) may take longer than a hotel search (with a hotel ID). The amount of data processed can also vary significantly. Therefore, to speed up the processing time of a city search request, you can use page-by-page data retrieval (Page=1, PerPage=10) with one room (OneRoom=1) without price detail (PriceDetail=0). If you want to view all hotel rooms (OneRoom=0), then you can skip using page-by-page data retrieval (Page=0, PerPage=0) and enable price detailing to get more accurate information on prices and availability (PriceDetail=1).

As a result of the search arrives the data at the following structure. Response attributes and elements:

- Hotel – container hotel. Includes a plurality of containers Room. Attributes:
 - id – hotel identifier
- Room – container hotel room. It contains information about the hotel room. Attributes:
 - id – hotel room identifier
- HotelRatePlan – rate plan. Attributes:
 - id – unique rate plan identifier
 - name – rate plan name
- RoomType – hotel room type. Attributes:
 - id – unique hotel room type identifier
- RoomCategory – hotel room category. Attributes:
 - id – unique hotel room category identifier
- PlacementType – type of placement in hotel room. Attributes:
 - id – placement type identifier
- FoodType – type of food in hotel room. Attributes:
 - id – food type identifier
- Price – price for hotel room for all period. Currency of price is currency that has been specified in the request parameters
- PricePerDay – prices for room per each day. This element is included in a response only if in a request was set price detailing per each day (parameter PriceDetail = 1). It contains many containers PricePerDayDetails
- PricePerDayDetails – price detailing per room for each day. Contains elements:
 - Date – date
 - Price – price
 - Availability – room availability on current day
 - Allotment – count of available allotments on current day
- Availability – room availability for all period. Possible values:
 - online – available for instant booking
 - onrequest – available on request
- Allotment – count of available allotments for all period
- CancellationPolicy – policy penalties on cancellation. Elements:
 - Price – penalty price

- DateTimeFrom – date of penalty begin
- BookingPolicyInfo – information about reservation rules on the specified language
- CancellationPolicyInfo – information about cancellation rules on the specified language
- Page – page number (if paginated data retrieval is used)
- TotalPages – total number of pages (if paginated data retrieval is used)

Request example for search by city:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchHotelsRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>ru</Language>
      <Currency>17</Currency>
      <City>29</City>
      <Hotel>0</Hotel>
      <DateRange from="2014-08-13" to="2014-08-16"/>
      <Adults count="1"/>
      <Children count="0"/>
      <Availability>all</Availability>
      <PriceDetail>0</PriceDetail>
      <OneRoom>1</OneRoom>
      <Page>1</Page>
      <PerPage>10</PerPage>
    </SearchHotelsRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example for search by city:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchHotelsRS>
      <Hotel id="185">
        <Room id="313">
          <HotelRatePlan id="1" name="Регулярный тариф"/>
          <RoomType id="9"/>
          <RoomCategory id="1"/>
          <PlacementType id="1"/>
          <FoodType id="15"/>
          <Price>3030</Price>
          <Availability>online</Availability>
          <Allotment>10</Allotment>
          <CancellationPolicy>
            <Price>0</Price>
            <DateTimeFrom>2014-08-11 12:00:00</DateTimeFrom>
          </CancellationPolicy>
          <BookingPolicyInfo>Бронирование является гарантированным до 18:00 в день заезда. Пребывающие после 18:00 гости должны уведомить об этом в поле «Примечания» во время бронирования или сообщением по
```

электронной почте, с указанием номера бронирования до 14:00 дня заезда. В противном случае, New Logic имеет право отменить бронирование на условиях незаезда (см. Условия аннуляции).</BookingPolicyInfo>

<CancellationPolicyInfo>Заказ может быть аннулирован бесплатно до времени наступления штрафных санкций. В случае поздней аннуляции или незаезда гостя система бронирования взимает 100% стоимости номера за 1 сутки.</CancellationPolicyInfo>

</Room>

</Hotel>

<Hotel id="160">

<Room id="375">

<HotelRatePlan id="1" name="Регулярный тариф"/>

<RoomType id="8"/>

<RoomCategory id="1"/>

<PlacementType id="1"/>

<FoodType id="15"/>

<Price>3501</Price>

<Availability>onrequest</Availability>

<Allotment>10</Allotment>

<CancellationPolicy>

<Price>0</Price>

<DateTimeFrom>2014-08-12 14:00:00</DateTimeFrom>

</CancellationPolicy>

<BookingPolicyInfo>Бронирование является гарантированным до 18:00 в день заезда. Пребывающие после 18:00 гости должны уведомить об этом в поле «Примечания» во время бронирования или сообщением по электронной почте, с указанием номера бронирования до 14:00 дня заезда. В противном случае, New Logic имеет право отменить бронирование на условиях незаезда (см. Условия аннуляции).</BookingPolicyInfo>

<CancellationPolicyInfo>Заказ может быть аннулирован бесплатно до времени наступления штрафных санкций. В случае поздней аннуляции или незаезда гостя система бронирования взимает 100% стоимости номера за 1 сутки.</CancellationPolicyInfo>

</Room>

</Hotel>

<Page>1</Page>

<TotalPages>7</TotalPages>

</SearchHotelsRS>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Request example for search by hotel:

```
<?xml version="1.0" encoding="UTF-8"?>
```

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

```
<SOAP-ENV:Body>
```

```
<SearchHotelsRQ>
```

```
<Credentials>
```

```
<Login>test@test.com</Login>
```

```
<Password>***</Password>
```

```
</Credentials>
```

```
<Language>ru</Language>
```

```
<Currency>19</Currency>
```

```
<City>0</City>
```

```
<Hotel>80</Hotel>
```

```
<DateRange from="2014-08-13" to="2014-08-16"/>
```

```
<Adults count="1"/>
```

```
<Children count="2">
```

```
<Child age="4"/>
```

```
<Child age="6"/>
```

```
</Children>
```

```
<Availability>all</Availability>
```

```
<PriceDetail>1</PriceDetail>
```

```

<OneRoom>0</OneRoom>
<Page>0</Page>
<PerPage>0</PerPage>
</SearchHotelsRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example for search by hotel:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchHotelsRS>
      <Hotel id="300">
        <Room id="1218">
          <HotelRatePlan id="1" name="Регулярный тариф"/>
          <RoomType id="7"/>
          <RoomCategory id="1"/>
          <PlacementType id="3"/>
          <FoodType id="19"/>
          <Price>4002</Price>
          <PricePerDay>
            <PricePerDayDetails>
              <Date>2014-08-13</Date>
              <Price>1334</Price>
              <Availability>onrequest</Availability>
              <Allotment>10</Allotment>
            </PricePerDayDetails>
            <PricePerDayDetails>
              <Date>2014-08-14</Date>
              <Price>1334</Price>
              <Availability>onrequest</Availability>
              <Allotment>10</Allotment>
            </PricePerDayDetails>
            <PricePerDayDetails>
              <Date>2014-08-15</Date>
              <Price>1334</Price>
              <Availability>onrequest</Availability>
              <Allotment>10</Allotment>
            </PricePerDayDetails>
          </PricePerDay>
          <Availability>onrequest</Availability>
          <Allotment>10</Allotment>
          <CancellationPolicy>
            <Price>1334</Price>
            <DateTimeFrom>2014-08-10 14:00:00</DateTimeFrom>
          </CancellationPolicy>
          <BookingPolicyInfo>Бронирование является гарантированным до 18:00 в день заезда. Пребывающие после 18:00 гости должны уведомить об этом в поле «Примечания» во время бронирования или сообщением по электронной почте, с указанием номера бронирования до 14:00 дня заезда. В противном случае, New Logic имеет право отменить бронирование на условиях незаезда (см. Условия аннуляции).</BookingPolicyInfo>
          <CancellationPolicyInfo>Заказ может быть аннулирован бесплатно до времени наступления штрафных санкций. В случае поздней аннуляции или незаезда гостя система бронирования взимает 100% стоимости номера за 1 сутки.</CancellationPolicyInfo>
        </Room>
        <Room id="1219">
          <HotelRatePlan id="2" name="Невозвратный тариф"/>
          <RoomType id="9"/>

```

```

<RoomCategory id="1"/>
<PlacementType id="3"/>
<FoodType id="19"/>
<Price>6843</Price>
<PricePerDay>
  <PricePerDayDetails>
    <Date>2014-08-13</Date>
    <Price>2281</Price>
    <Availability>online</Availability>
    <Allotment>5</Allotment>
  </PricePerDayDetails>
  <PricePerDayDetails>
    <Date>2014-08-14</Date>
    <Price>2281</Price>
    <Availability>online</Availability>
    <Allotment>1</Allotment>
  </PricePerDayDetails>
  <PricePerDayDetails>
    <Date>2014-08-15</Date>
    <Price>2281</Price>
    <Availability>online</Availability>
    <Allotment>1</Allotment>
  </PricePerDayDetails>
</PricePerDay>
<Availability>online</Availability>
<Allotment>1</Allotment>
<CancellationPolicy>
  <Price>2281</Price>
  <DateTimeFrom>2014-08-10 14:00:00</DateTimeFrom>
</CancellationPolicy>
<BookingPolicyInfo>Бронирование является гарантированным до 18:00 в день заезда. Пребывающие
после 18:00 гости должны уведомить об этом в поле «Примечания» во время бронирования или сообщением по
электронной почте, с указанием номера бронирования до 14:00 дня заезда. В противном случае, New Logic имеет
право отменить бронирование на условиях незаезда (см. Условия аннуляции).</BookingPolicyInfo>
<CancellationPolicyInfo>Заказ может быть аннулирован бесплатно до времени наступления штрафных
санкций. В случае поздней аннуляции или незаезда гостя система бронирования взимает 100% стоимости номера
за 1 сутки.</CancellationPolicyInfo>
</Room>
</Hotel>
<Page>0</Page>
<TotalPages>0</TotalPages>
</SearchHotelsRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

3.2 GetHotelOvertime – getting prices for early check-in and late check-out of the hotel

Description:

Getting prices for early check-in and late check-out of the hotel. Using this method is not necessary. It can be used if need to book early check-in and/or late check-out to show value fee for early check-in/late check-out. This method does not affect the process of booking and is informative.

Prices for early check-in/late check-out may be different at different times – for example, price for late check-out at 16:00 – 15 USD, and at 18:00 – 20 USD.

There are two types of early check-in – non-guaranteed and guaranteed. Their prices may be different, so the response given both types of early check-in.

Request elements:

- Currency – currency identifier
- Room – room identifier
- HotelRatePlan – rate plan identifier
- PlacementType – placement type identifier
- FoodType – food type identifier
- DateRange – period of stay. Attributes:
 - from – check-in date
 - to – check-out date
- EarlyCheckIn – early check-in. Attributes (optional):
 - Time – if you need to get price of early check-in at the exact time
 - TimeFrom – if you need to get prices of early check-in for the period "from" a specified time
 - TimeTo – if you need to get prices of early check-in for the period "to" specified time

Note: if set none of the attributes, all prices of early check-in are presented in the response
- LateCheckOut – late check-out. Attributes (optional):
 - Time – if you need to get price of late check-out at the exact time
 - TimeFrom – if you need to get prices of late check-out for the period "from" a specified time
 - TimeTo – if you need to get prices of late check-out for the period "to" specified time

Note: if set none of the attributes, all prices of late check-out are presented in the response

As a result arrives the data with prices at the following structure. Response attributes and elements:

- HotelOvertime – container for hotel overtime price; contains price and period of time during which this price is operates
- Type – type of overtime. Available values:
 - Nonguaranteed early check-in
 - Guaranteed early check-in
 - Late check-out
- TimeFrom – the time, from which this price is operates
- TimeTo – the time, to which this price is operates
- Price – price for overtime

Note: prices for early check-in and late check-out of the hotel can vary on different days. In the response are presented prices for a particular booking period specified in the request parameter DateRange.

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <GetHotelOvertimeRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Currency>17</Currency>
      <Room>298</Room>
      <HotelRatePlan>2094</HotelRatePlan>
      <PlacementType>3</PlacementType>
      <FoodType>15</FoodType>
      <DateRange from="2015-08-25" to="2015-08-31"/>
      <EarlyCheckIn TimeFrom="09:00" TimeTo="13:00"/>
      <LateCheckOut Time="14:10"/>
    </GetHotelOvertimeRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <GetHotelOvertimeRS>
      <HotelOvertime>
        <Type>Nonguaranteed early check-in</Type>
        <TimeFrom>08:00:00</TimeFrom>
        <TimeTo>09:59:00</TimeTo>
        <Price>426</Price>
      </HotelOvertime>
      <HotelOvertime>
        <Type>Nonguaranteed early check-in</Type>
        <TimeFrom>10:00:00</TimeFrom>
        <TimeTo>11:59:00</TimeTo>
        <Price>535</Price>
      </HotelOvertime>
      <HotelOvertime>
        <Type>Nonguaranteed early check-in</Type>
        <TimeFrom>12:00:00</TimeFrom>
        <TimeTo>13:59:00</TimeTo>
        <Price>562</Price>
      </HotelOvertime>
      <HotelOvertime>
        <Type>Guaranteed early check-in</Type>
        <TimeFrom>00:00:00</TimeFrom>
        <TimeTo>13:59:00</TimeTo>
        <Price>719</Price>
      </HotelOvertime>
      <HotelOvertime>
        <Type>Late check-out</Type>
        <TimeFrom>14:01:00</TimeFrom>
        <TimeTo>15:00:00</TimeTo>
        <Price>408</Price>
      </HotelOvertime>
    </GetHotelOvertimeRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
</HotelOvertime>
</GetHotelOvertimeRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.3 GetHotelSurcharges – getting the list of hotel surcharges

Description:

For some hotels there may be surcharges that apply when ordering these hotels. Surcharges may be several for one hotel. All of them are added to the order. This method is designed to obtain a list of hotel surcharges.

For example, if the cost of living in a hotel is 200 USD and the hotel has a surcharge for booking 20 USD and an extra charge for tourist tax of 30 USD, then the total amount of the reservation will be 250 USD.

Request elements:

- Currency – currency identifier
- Hotel – hotel identifier

As a result arrives the data with surcharges at the following structure. Response attributes and elements:

- HotelSurcharge – hotel surcharge. Attributes:
 - id – surcharge identifier
 - name – surcharge name on the specified language
 - price – surcharge price in the specified currency

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <GetHotelSurchargesRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Currency>17</Currency>
      <Hotel>80</Hotel>
    </GetHotelSurchargesRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <GetHotelSurchargesRS>
```

```
<HotelSurcharge id="1" name="Booking fee" price="350"/>
<HotelSurcharge id="2" name="Tourist tax" price="500"/>
<HotelSurcharge id="3" name="Services" price="200"/>
</GetHotelSurchargesRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.4 SearchExcursions – search of excursions

This method is temporary not allowed.

Description:

Search of excursions on the set parameters. Request elements (search parameters):

- Currency – currency identifier
- City – city identifier
- DateRange – excursion period. Attributes:
 - from – date begin of the excursion
 - to – date end of the excursion
- Adults – count of people for whom it will be holding the excursion. Attributes:
 - count – number of adults

As a result of the search arrives the data at the following structure. Response attributes and elements:

- Excursion – container excursion. Attributes:
 - id – excursion identifier
- Price – price for excursion. Currency of price is currency that has been specified in the request parameters
- DateExcursion – date of excursion (for specification, if search was made for period of more than one day)
- CountPeopleMax – maximum possible count of people, for which can be holding the excursion
- CancellationPolicy – policy penalties on cancellation. Elements:
 - Price – penalty price
 - DateTimeFrom – date of penalty begin
- BookingPolicyInfo – information about reservation rules on the specified language
- CancellationPolicyInfo – information about cancellation rules on the specified language

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchExcursionRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
```

```
<Currency>18</Currency>
<City>29</City>
<DateRange from="2012-12-29" to="2012-12-30"/>
<Adults count="1"/>
</SearchExcursionRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchExcursionRS>
      <Excursion id="33">
        <Price>96</Price>
        <DateExcursion>2012-12-29</DateExcursion>
        <CountPeopleMax>3</CountPeopleMax>
        <CancellationPolicy>
          <Price>96</Price>
          <DateTimeFrom>2012-12-28 14:00:00</DateTimeFrom>
        </CancellationPolicy>
        <BookingPolicyInfo>Rule of booking of excursion</BookingPolicyInfo>
        <CancellationPolicyInfo>Rule of cancel of excursion</CancellationPolicyInfo>
      </Excursion>
    </SearchExcursionRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.5 SearchTransfers – search of transfers

This method is temporary not allowed.

Description:

Search of transfers on the set parameters. Request elements (search parameters):

- Currency – currency identifier
- City – city identifier
- Date – period of holding the transfer
- Adults – count of people for whom it will be holding the transfer. Attributes:
 - count – number of adults

As a result of the search arrives the data at the following structure. Response attributes and elements:

- Transfer – container transfer. Attributes:
 - id – transfer identifier
- Price – price for transfer. Currency of price is currency that has been specified in the request parameters
- CancellationPolicy – policy penalties on cancellation. Elements:
 - Price – penalty price
 - DateTimeFrom – date of penalty begin

- BookingPolicyInfo – information about reservation rules on the specified language
- CancellationPolicyInfo – information about cancellation rules on the specified language

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchTransferRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Currency>18</Currency>
      <City>29</City>
      <Date>2012-12-30</Date>
      <Adults count="1"/>
    </SearchTransferRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchTransferRS>
      <Transfer id="32">
        <Price>45</Price>
        <CancellationPolicy>
          <Price>45</Price>
          <DateTimeFrom>2012-12-28 14:00:00</DateTimeFrom>
        </CancellationPolicy>
        <BookingPolicyInfo>Rule of booking of transfer</BookingPolicyInfo>
        <CancellationPolicyInfo>Rule of cancel of transfer</CancellationPolicyInfo>
      </Transfer>
    </SearchTransferRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4 Operations with orders

4.1 AddOrder – ordering

Description:

This paragraph describes the process of making new general order for the customer. General order is order which may consist of several unit orders: hotels, excursions, transfers.

To booking the hotel uses the following parameters:

- hotel id
- period of stay (check-in date and check-out date)
- room id
- rate plan
- type of placement
- type of food
- count of hotel rooms, which includes the names of guests (for child is also required parameter is the age of the child), early check-in, late check-out, additional wishes of the client

Note: as there are two types of early check-in (non-guaranteed and guaranteed), when ordering early check-in you need to specify its type (parameter «type» in the EarlyCheckIn element); available values:

- 1 – non-guaranteed early check-in
- 2 – guaranteed early check-in

If in the request is not specified parameter «type» in the EarlyCheckIn element, by default ordering non-guaranteed early check-in.

To booking the excursion uses the following parameters:

- excursion id
- period of holding the excursion
- count of people participating in the excursion
- the start time of the excursion
- the meeting place
- additional wishes (optional)

To booking the transfer uses the following parameters:

- transfer id
- period of time for the transfer
- count of people to whom the transfer is offered
- the start time of the transfer
- the meeting place
- the delivery place
- additional wishes (optional)

Upon successful order creation comes the response with the following elements and attributes:

- Order – container order. Attributes:
 - UniqueID – unique order identifier

Request example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <AddOrderRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Currency>18</Currency>
      <Hotel id="80">
        <DateRange from="2012-12-29" to="2012-12-30"/>
        <Room id="31">
          <HotelRatePlan>3</HotelRatePlan>
          <PlacementType>3</PlacementType>
          <FoodType>15</FoodType>
          <Amount>
            <Adult name="Test1"/>
            <Child name="Test2" age="4"/>
            <EarlyCheckIn time="09:00" type="2"/>
            <LateCheckOut time="18:00"/>
            <ClientNote>Some client note</ClientNote>
          </Amount>
          <Amount>
            <Adult name="Test3"/>
            <Child name="Test4" age="2"/>
          </Amount>
        </Room>
        <Room id="36">
          <HotelRatePlan>1</HotelRatePlan>
          <PlacementType>1</PlacementType>
          <FoodType>15</FoodType>
          <Amount>
            <Adult name="Test5"/>
          </Amount>
        </Room>
      </Hotel>
      <Excursion id="33">
        <DateRange from="2012-12-29" to="2012-12-29"/>
        <Persons>3</Persons>
        <Time>07:15</Time>
        <MeetingPlace>Airport</MeetingPlace>
        <Note>Some text</Note>
      </Excursion>
      <Transfer id="38">
        <DateRange from="2012-12-29" to="2012-12-29"/>
        <Persons>3</Persons>
        <Time>10:30</Time>
        <MeetingPlace>Airport</MeetingPlace>
        <DeliveryPlace>Dnepr</DeliveryPlace>
        <Note>Some text</Note>
      </Transfer>
    </AddOrderRQ>
  </SOAP-ENV:Body>

```

</SOAP-ENV:Envelope>

The above example shows how to issue one general order, consisting of three unit orders: hotel, excursion and transfer.

Unit order of hotel (id – 80) consist of three rooms (id – 31 (2 pcs.), id – 36) on period from 29.12.2012 to 30.12.2012.

First two rooms (id – 31) have: rate plan – 3, type of placement – 3, type of food – 15. In the first of them will be living one adult, whose name is «Test1», and one child, whose name is «Test2» and age 4 years; this room has the following additional options: «EarlyCheckIn» – early check-in at 09:00 (attribute «type» = 2 indicates that ordered guaranteed early check-in), «LateCheckOut» – late check-out at 18:00, «ClientNote» – the client wishes «Some client note». In the second room will be living one adult, whose name is «Test3», and one child, whose name is «Test4» and age 2 years; early check-in, late check-out and client wishes for this room are not set.

The third room (id – 36) has: rate plan – 1, type of placement – 1, type of food – 15. In this room will be living one adult, whose name is «Test5».

Unit order excursion (id – 33) will be ordered on 29.12.2012 for three people whom will meet at 07:15 in «Airport» and will take into account wish «Some text».

Unit order transfer (id – 38) will be ordered on 29.12.2012 for three people whom will meet at 10:30 in «Airport» and will taken to «Dnepr», also take into account wish «Some text».

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <AddOrderRS>
      <Order UniqueID="4G02YS"/>
    </AddOrderRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2 SearchOrders – search of orders

Description:

Search of orders can be made on the check-in date, check-out date and hotel. Request elements (search parameters):

- DateFrom – start date of the period (optional parameter). If in request this parameter is absent, then as the value of this parameter is substituted current date
- DateTo – end date of the period (optional parameter). If in request this parameter is absent, then in result will be all orders that satisfy the previous parameter
- Hotel – hotel identifier (optional parameter). This parameter must be present in request, if it is necessary to obtain a list of orders for a specific hotel

As a result, comes the answer, that contains a list of orders, in which check-in/check-out date is in period from «DateFrom» to «DateTo», ordered in the hotel «Hotel». Response attributes and elements:

- Order – container of general order. Attributes:
 - UniqueID – unique general order identifier
- OrderStatus – status of the order
- DateUpdate – date and time of last modification
- OrderHotel – container of unit hotel order. Attributes:
 - UniqueID – unique unit order identifier
- OrderExcursion – container of unit excursion order. Attributes:
 - UniqueID – unique unit order identifier
- OrderTransfer – container of unit transfer order. Attributes:
 - UniqueID – unique unit order identifier

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchOrdersRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <DateFrom>2015-07-01</DateFrom>
      <DateTo>2015-07-16</DateTo>
      <Hotel>80</Hotel>
    </SearchOrdersRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SearchOrdersRS>
      <Order UniqueID="WLAWWV">
        <OrderStatus>5</OrderStatus>
        <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        <OrderHotel UniqueID="WLAWWVA">
          <OrderStatus>5</OrderStatus>
          <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        </OrderHotel>
      </Order>
      <Order UniqueID="CDN34N">
        <OrderStatus>1</OrderStatus>
        <DateUpdate>2015-04-09 07:23:40</DateUpdate>
        <OrderHotel UniqueID="CDN34NA">
          <OrderStatus>1</OrderStatus>
          <DateUpdate>2015-04-09 07:23:40</DateUpdate>
        </OrderHotel>
    </SearchOrdersRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

<OrderHotel UniqueID="CDN34NB">
  <OrderStatus>6</OrderStatus>
  <DateUpdate>2015-04-08 10:15:40</DateUpdate>
</OrderHotel>
</Order>
<Order UniqueID="NIT2DF">
  <OrderStatus>5</OrderStatus>
  <DateUpdate>2015-04-24 12:07:15</DateUpdate>
  <OrderHotel UniqueID="NIT2DFA">
    <OrderStatus>5</OrderStatus>
    <DateUpdate>2015-04-24 12:07:15</DateUpdate>
  </OrderHotel>
  <OrderExcursion UniqueID="NIT2DFB">
    <OrderStatus>5</OrderStatus>
    <DateUpdate>2015-04-24 12:07:15</DateUpdate>
  </OrderExcursion>
  <OrderTransfer UniqueID="NIT2DFC">
    <OrderStatus>5</OrderStatus>
    <DateUpdate>2015-04-24 12:07:15</DateUpdate>
  </OrderTransfer>
</Order>
</SearchOrdersRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.3 GetOrder - getting information about the order

Description:

Getting information about the order made by the following parameters:

- Order – container order. Attributes:
 - UniqueID – unique general order identifier

As a result, comes complete information about the order. Response attributes and elements:

- Order – container of general order. Attributes:
 - UniqueID – identifier of general order
- OrderStatus – status of general or unit order (depending on which element the given element is in)
- Currency – order currency
- TotalPrice – total price of order
- OrderHotel – container of unit order hotel. Attributes:
 - UniqueID – identifier of unit order
- Price – price for unit order
- PenaltyEdit – price of penalties for modification the unit order (*Note*: price of penalties is included in price of unit order, in the element Price)
- PenaltyCancel – price of penalties for cancellation the unit order (*Note*: price of penalties is included in price of unit order, in the element Price)
- Hotel – container hotel. Attributes:
 - id – hotel identifier
- DateRange – period of stay in hotel/date of excursion/date of transfer. Attributes:

- from – period begin
- to – period end
- Room – container hotel room. Attributes:
 - id – hotel room identifier
- HotelRatePlan – rate plan. Attributes:
 - id – unique rate plan identifier
 - name – rate plan name on the specified language
- PlacementType – placement type identifier
- FoodType – food type identifier
- Amount – count of hotel rooms. It is determined from the count of residents
- Adult – adult. Attributes:
 - name – name of adult
- Child – child. Attributes:
 - name – name of child
 - age – age of child
- EarlyCheckIn – early check-in. Attributes:
 - time – time of early check-in
 - type – type of early check-in. Available values:
 - 1 – non-guaranteed early check-in
 - 2 – guaranteed early check-in
- LateCheckOut – late check-out. Attributes:
 - time – time of late check-out
- ClientNote – additional client wishes
- HotelSurcharge – hotel surcharge. Attributes:
 - id – surcharge identifier
 - name – surcharge name
 - price – surcharge price
- CancellationPolicy – policy penalties on cancellation. Elements:
 - Price – penalty price
 - DateTimeFrom – date of penalty begin
- OrderExcursion – container of unit order excursion. Attributes:
 - UniqueID – identifier of unit order
- Excursion – container excursion. Attributes:
 - id – excursion identifier
- Persons – count of people taking part in excursion/transfer
- Time – time of beginning for excursion/transfer
- MeetingPlace – meeting place for excursion/transfer
- Note – note, wish for excursion/transfer
- OrderTransfer – container of unit order transfer. Attributes:
 - UniqueID – identifier of unit order
- Transfer – container transfer. Attributes:
 - id – transfer identifier
- DeliveryPlace – delivery place for transfer

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <GetOrderRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Order UniqueID="RSA1R1"/>
    </GetOrderRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <GetOrderRS>
      <Order UniqueID="RSA1R1">
        <OrderStatus>9</OrderStatus>
        <Currency>18</Currency>
        <TotalPrice>671</TotalPrice>
        <OrderHotel UniqueID="RSA1R1A">
          <OrderStatus>9</OrderStatus>
          <Price>470</Price>
          <PenaltyEdit>20</PenaltyEdit>
          <PenaltyCancel>0</PenaltyCancel>
          <Hotel id="80">
            <DateRange from="2012-12-29" to="2012-12-30"/>
            <Room id="31">
              <HotelRatePlan id="3" name="Non-refundable"/>
              <PlacementType>3</PlacementType>
              <FoodType>15</FoodType>
              <Price>470</Price>
              <Amount>
                <Adult name="test1"/>
                <Child name="test1" age="4"/>
                <EarlyCheckIn time="09:00" type="1"/>
                <LateCheckOut time="23:00"/>
                <ClientNote>Some client note</ClientNote>
              </Amount>
              <Amount>
                <Adult name="test2"/>
                <Child name="test2" age="2"/>
              </Amount>
            </Room>
            <HotelSurcharge id="1" name="Booking fee" price="50"/>
            <HotelSurcharge id="2" name="Tourist tax" price="20"/>
          </Hotel>
          <CancellationPolicy>
            <Price>151</Price>
            <DateTimeFrom>2012-12-28 14:00:00</DateTimeFrom>
          </CancellationPolicy>
        </OrderHotel>
        <OrderExcursion UniqueID="RSA1R1C">
          <OrderStatus>9</OrderStatus>
          <Price>96</Price>
```

```

<PenaltyEdit>10</PenaltyEdit>
<PenaltyCancel>0</PenaltyCancel>
<Excursion id="33">
  <DateRange from="2012-12-29" to="2012-12-29"/>
  <Persons>3</Persons>
  <Time>07:15</Time>
  <MeetingPlace>Airport</MeetingPlace>
  <Note>some text1</Note>
</Excursion>
<CancellationPolicy>
  <Price>96</Price>
  <DateTimeFrom>2012-12-28 14:00:00</DateTimeFrom>
</CancellationPolicy>
</OrderExcursion>
<OrderTransfer UniqueID="RSA1R1D">
  <OrderStatus>9</OrderStatus>
  <Price>35</Price>
  <PenaltyEdit>0</PenaltyEdit>
  <PenaltyCancel>0</PenaltyCancel>
  <Transfer id="38">
    <DateRange from="2012-12-29" to="2012-12-29"/>
    <Persons>3</Persons>
    <Time>10:30</Time>
    <MeetingPlace>Airport</MeetingPlace>
    <DeliveryPlace>Dnepr</DeliveryPlace>
    <Note>some text2</Note>
  </Transfer>
  <CancellationPolicy>
    <Price>35</Price>
    <DateTimeFrom>2012-12-28 14:00:00</DateTimeFrom>
  </CancellationPolicy>
</OrderTransfer>
</Order>
</GetOrderRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.4 EditOrder – order modification

Description:

Editing an order occurs by transferring new parameters for the order. Upon successful verification of all parameters and the possibility of editing with new parameters, the order is updated, the unique identifier of which is transferred in the UniqueID parameter.

Request attributes and elements are similar to the request of the method «[AddOrder](#)». In addition to them, the Order element with the UniqueID attribute is added, which contains a unique identifier of a general order. Also, the UniqueID attribute is added to the Hotel, Excursion and Transfer elements, which contains a unique identifier of a unit order.

Upon successful order modification comes the answer with the following elements and attributes:

- Order – container order. Has attributes:
 - UniqueID – unique order identifier

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <EditOrderRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Order UniqueID="RSA1R1"/>
      <Currency>18</Currency>
      <Hotel id="80" UniqueID="RSA1R1A">
        <DateRange from="2012-12-29" to="2012-12-30"/>
        <Room id="31">
          <HotelRatePlan>3</HotelRatePlan>
          <PlacementType>3</PlacementType>
          <FoodType>15</FoodType>
          <Amount>
            <Adult name="Test1"/>
            <Child name="Test2" age="4"/>
            <EarlyCheckIn time="09:00" type="1"/>
            <LateCheckOut time="18:00"/>
            <ClientNote>Some client note</ClientNote>
          </Amount>
          <Amount>
            <Adult name="Test3"/>
            <Child name="Test4" age="2"/>
          </Amount>
        </Room>
      </Hotel>
      <Excursion id="33" UniqueID="RSA1R1C">
        <DateRange from="2012-12-29" to="2012-12-29"/>
        <Persons>3</Persons>
        <Time>07:15</Time>
        <MeetingPlace>Airport</MeetingPlace>
        <Note>Some text</Note>
      </Excursion>
      <Transfer id="38" UniqueID="RSA1R1D">
        <DateRange from="2012-12-29" to="2012-12-29"/>
        <Persons>3</Persons>
        <Time>10:30</Time>
        <MeetingPlace>Airport</MeetingPlace>
        <DeliveryPlace>Dnepr</DeliveryPlace>
        <Note>Some text</Note>
      </Transfer>
    </EditOrderRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
```

```
<EditOrderRS>
  <Order UniqueID="RSA1R1"/>
</EditOrderRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.5 SimulateEditOrder – simulating order modification

Description:

This method is designed to simulate order modification in order to see what it will look like after modification. It is not necessary to use this method. It is intended for informative purpose only.

Attention! This method *does not* edit the order, but only simulates it. To edit an order, you must use the «[EditOrder](#)» method. Between the execution of the «SimulateEditOrder» method and the execution of the «EditOrder» method, the information on the server may change. Therefore, to view the information of the modified order after the «[EditOrder](#)» method is executed, you need to use the «[GetOrder](#)» method.

Note: simulating order modification works only for hotel orders. For excursion orders and transfer orders this method does not work.

Request attributes and elements in this method are the same as in the request of the «[EditOrder](#)» method. Response attributes and elements in this method are the same as in the response of the «[GetOrder](#)» method.

Request example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SimulateEditOrderRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Order UniqueID="RSA1R1"/>
      <Currency>18</Currency>
      <Hotel id="80" UniqueID="RSA1R1A">
        <DateRange from="2012-12-29" to="2012-12-30"/>
        <Room id="31">
          <HotelRatePlan>3</HotelRatePlan>
          <PlacementType>3</PlacementType>
          <FoodType>15</FoodType>
          <Amount>
            <Adult name="Test1"/>
            <Child name="Test2" age="4"/>
            <EarlyCheckIn time="09:00" type="1"/>
            <LateCheckOut time="18:00"/>
            <ClientNote>Some client note</ClientNote>
          </Amount>
          <Amount>
            <Adult name="Test3"/>
          </Amount>
        </Room>
      </Hotel>
    </SimulateEditOrderRQ>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        <Child name="Test4" age="2"/>
    </Amount>
</Room>
</Hotel>
</SimulateEditOrderRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response example:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SimulateEditOrderRS>
      <Order UniqueID="RSA1R1">
        <OrderStatus>9</OrderStatus>
        <Currency>18</Currency>
        <TotalPrice>671</TotalPrice>
        <OrderHotel UniqueID="RSA1R1A">
          <OrderStatus>9</OrderStatus>
          <Price>470</Price>
          <PenaltyEdit>20</PenaltyEdit>
          <PenaltyCancel>0</PenaltyCancel>
          <Hotel id="80">
            <DateRange from="2012-12-29" to="2012-12-30"/>
            <Room id="31">
              <HotelRatePlan id="3" name="Non-refundable"/>
              <PlacementType>3</PlacementType>
              <FoodType>15</FoodType>
              <Price>470</Price>
              <Amount>
                <Adult name="test1"/>
                <Child name="test1" age="4"/>
                <EarlyCheckIn time="09:00" type="1"/>
                <LateCheckOut time="23:00"/>
                <ClientNote>Some client note</ClientNote>
              </Amount>
              <Amount>
                <Adult name="test2"/>
                <Child name="test2" age="2"/>
              </Amount>
            </Room>
            <HotelSurcharge id="1" name="Booking fee" price="50"/>
            <HotelSurcharge id="2" name="Tourist tax" price="20"/>
          </Hotel>
          <CancellationPolicy>
            <Price>151</Price>
            <DateTimeFrom>2012-12-28 14:00:00</DateTimeFrom>
          </CancellationPolicy>
        </OrderHotel>
      </Order>
    </SimulateEditOrderRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```


4.6 ConfirmOrder – order confirmation

Description:

Orders, that are available for instant booking (online), confirmed immediately at the time of booking, thus further confirming its not need. This method applies only to orders that are available on request (onrequest).

When booking order, available on request (onrequest), a request is sent to the supplier for availability. This order meanwhile transferred to the status «requested». The supplier can confirm the order (in case of availability), or cancel the order (in case of not availability). If the provider confirms the order, it is transferred to the status «confirmed». In this case, the client will have to confirm his booking. This is why this method is needed.

There are two types of order – general and unit. General order consists of a plurality of unit orders. It is possible to confirm only one general order or several unit orders in one general order. In case of confirmation all single orders included into general one, general order is confirmed automatically. In case of confirmation general order, all single orders included into it are also confirmed automatically (only those that is in status «confirmed»). If it is necessary to confirm all unit orders, included into general one, it will be enough to send request for the confirmation of general order.

Note: you can confirm only orders in status «confirmed». If in request were set orders in other statuses – such orders will be ignored.

For the confirmation of general order, you need to send request that will contain the order identifier (UniqueID). For the confirmation of only one or several unit orders in one general order, you need to set their identifiers in the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements respectively.

Response will contain the status of general order and the statuses of all its unit orders.

Elements and attributes:

- Order – container of general order. Attributes:
 - UniqueID – unique general order identifier
- OrderStatus – status of the order
- DateUpdate – date and time of last modification
- OrderHotel – container of unit hotel order. Attributes:
 - UniqueID – unique unit order identifier
- OrderExcursion – container of unit excursion order. Attributes:
 - UniqueID – unique unit order identifier
- OrderTransfer – container of unit transfer order. Attributes:
 - UniqueID – unique unit order identifier

Request example of confirmation whole general order:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ConfirmOrderRQ>
      <Credentials>
```

```
<Login>test@test.com</Login>
<Password>***</Password>
</Credentials>
<Language>en</Language>
<Order UniqueID="SG69XF"/>
</ConfirmOrderRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example for previous request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ConfirmOrderRS>
      <Order UniqueID="SG69XF">
        <OrderStatus>5</OrderStatus>
        <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        <OrderHotel UniqueID="SG69XFA">
          <OrderStatus>5</OrderStatus>
          <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        </OrderHotel>
        <OrderHotel UniqueID="SG69XFB">
          <OrderStatus>5</OrderStatus>
          <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        </OrderHotel>
      </Order>
    </ConfirmOrderRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Request example of confirmation one unit order from general order:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ConfirmOrderRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Order UniqueID="SG69XF"/>
      <Order UniqueID="SG69XFA"/>
    </Order>
  </ConfirmOrderRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example for previous request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ConfirmOrderRS>
      <Order UniqueID="SG69XF">
        <OrderStatus>6</OrderStatus>
        <DateUpdate>2015-04-08 18:45:30</DateUpdate>
      </Order>
    </ConfirmOrderRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

<OrderHotel UniqueID="SG69XFA">
  <OrderStatus>8</OrderStatus>
  <DateUpdate>2015-04-08 18:45:30</DateUpdate>
</OrderHotel>
<OrderHotel UniqueID="SG69XFB">
  <OrderStatus>6</OrderStatus>
  <DateUpdate>2015-04-07 10:15:00</DateUpdate>
</OrderHotel>
</Order>
</ConfirmOrderRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.7 CancelOrder – order cancellation

Description:

There are two types of order – general and unit. General order consists of a plurality of unit orders. It is possible to cancel only one general order or several unit orders in one general order. In case of cancellation all single orders included into general one, general order is cancelled automatically. In case of cancellation general order, all single orders included into it are also cancelled automatically. If it is necessary to cancel all unit orders, included into general one, it will be enough to send request for the cancellation of general order.

Note: you can cancel orders in any status.

For the cancellation of general order, you need to send request that will contain the order identifier (UniqueID). For the cancellation of only one or several unit orders in one general order, you need to set their identifiers in the «OrderHotel», «OrderExcursion» and «OrderTransfer» elements respectively.

Response will contain the status of general order and the statuses of all its unit orders.

Elements and attributes:

- Order – container of general order. Attributes:
 - UniqueID – unique general order identifier
- OrderStatus – status of the order
- DateUpdate – date and time of last modification
- OrderHotel – container of unit hotel order. Attributes:
 - UniqueID – unique unit order identifier
- OrderExcursion – container of unit excursion order. Attributes:
 - UniqueID – unique unit order identifier
- OrderTransfer – container of unit transfer order. Attributes:
 - UniqueID – unique unit order identifier

Request example of cancellation whole general order:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CancelOrderRQ>
      <Credentials>

```

```
<Login>test@test.com</Login>
<Password>***</Password>
</Credentials>
<Language>en</Language>
<Order UniqueID="SG69XF"/>
</CancelOrderRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example for previous request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CancelOrderRS>
      <Order UniqueID="SG69XF">
        <OrderStatus>5</OrderStatus>
        <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        <OrderHotel UniqueID="SG69XFA">
          <OrderStatus>5</OrderStatus>
          <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        </OrderHotel>
        <OrderHotel UniqueID="SG69XFB">
          <OrderStatus>5</OrderStatus>
          <DateUpdate>2015-04-08 18:45:30</DateUpdate>
        </OrderHotel>
      </Order>
    </CancelOrderRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Request example of cancellation one unit order from general order:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CancelOrderRQ>
      <Credentials>
        <Login>test@test.com</Login>
        <Password>***</Password>
      </Credentials>
      <Language>en</Language>
      <Order UniqueID="SG69XF"/>
      <Order UniqueID="SG69XFA"/>
    </Order>
  </CancelOrderRQ>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response example for previous request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <CancelOrderRS>
      <Order UniqueID="SG69XF">
        <OrderStatus>6</OrderStatus>
        <DateUpdate>2015-04-08 18:45:30</DateUpdate>
      </Order>
    </CancelOrderRS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<OrderHotel UniqueID="SG69XFA">
  <OrderStatus>8</OrderStatus>
  <DateUpdate>2015-04-08 18:45:30</DateUpdate>
</OrderHotel>
<OrderHotel UniqueID="SG69XFB">
  <OrderStatus>6</OrderStatus>
  <DateUpdate>2015-04-07 10:15:00</DateUpdate>
</OrderHotel>
</Order>
</CancelOrderRS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5 Contacts

If you have any questions, you can ask them the following contacts:

Organizational issues:

Igor Iatsenko – director

igor@newlogic.ua

+38-044-206-33-22

Technical issues:

Oleksandr Trelin – lead developer

o.trelin@newlogic.ua