LeaderStay

The Villa Distribution System

# THE TOUR OPERATOR – SYSTEM COMMUNICATION

The reservation administration system uses the SOAP protocol for the communication between tour operators and LeaderStay – which is in essence wrapped XML information. It is implemented with the use of NuSOAP, a SOAP library.

The implementation of this protocol from making reservations demands only to include the necessary SOAP library, construct an array with the data of the reservation, make a SOAP call, and handle the answer you'll receive.

WSDL can also be used with the service

The system registers ten methods:

- reserve
- amend
- cancel
- getAvailableInfo
- getPriceAvailability
- getInfoAvailabilityList
- getPriceAvailabilityList
- getYearAvailability
- getPriceCatalogue
- getAreaStructure
- getGlobalHash

# Methods Description

## A. Reserve

The **reserve method** is used for the reservation.

It takes as an input the following variables:

1. id: The tour operator’s id
2. code: The tour operator’s order id (it can be left empty)
3. firstname: The first name of the customer
4. lastname: The last name of the customer
5. email: The email of the customer. It can be left empty, and we will use the email of the tour operator instead
6. telnumber: The telephone of the customer
7. countryofresidence: The country of residence of the customer
8. propertycode: The code property
9. checkindate: The check-in date
10. checkintime: The check-in time
11. checkouttime: The check-out time
12. numberofnights: The number of nights that the property will be reserved
13. asksuitability: It should be used only in the case when there is a suitability item with a value 2 on the ‘quantity’ field[1]. If you want to request more than one items, you should write these items and put the character ‘|’ between them. P.ex. “pets|smokers”. Optionally, it is preferred for you to write a detailed request in the notes. P.ex. ‘We ask if we are allowed to bring a small trained dog.’ )
14. persons: The persons that will live in the property

It provides as output the result of the procedure method and the confirmation number of the order which is the reference number of the order in the system.

When the order reaches the system, the method performs the following:

- Checks the id of the tour operator
- Checks if there is an already existing order with this code from the tour operator
- Check if the property is reserved in the specified dates
- Estimates the price of the tour operator
- Checks if there exists an offer and remeasures the total sums
- Finds if the property is on request. If it is not on request, it means that the reservation will be automatically confirmed
- Sends email to the owner and the company’s administrator with links to accept or decline the reservation and a link to handle the no-shows of the customers

[1] You will know if there is a suitability item for which you should ask, using the getAvailableInfo call described below.

## B. Cancel

The **cancel method** is used to let he tour operator cancel an existing reservation.

It takes as input:

1. id: The tour operator's id
2. code: The tour operator's reservation id (this field can be left empty, as long as the next field will not be empty)
3. LScode: The reference number of the order in the LeaderStay system (this field can be left empty, as long as the previous field will not be empty)

**Not both codes are required**

The method performs the following:

- Checks if there is an existing reservation with the same tour operator code and confirmation number
- Cancels it

It returns the result of the cancellation.

## C. Amend

The **amend method** is used for the amendment of an order.

It takes as input:

1. id: The tour operator's id
2. old_code: The tour operator's old reservation id (this field can be left empty, as long as the next field will not be empty)
3. old_LScode: The reference number of the old order in the LeaderStay system (this field can be left empty, as long as the previous field will not be empty)
4. code: The tour operator's order id (it can be left empty)
5. firstname: The first name of the customer
6. lastname: The last name of the customer
7. email: The email of the customer. It can be left empty, and we will use the email of the tour operator instead
8. telnumber: The telephone of the customer
9. countryofresidence: The country of residence of the customer
10. checkindate: The check-in date
11. checkintime: The check-in time
12. checkouttime: The check-out time
13. numberofnights: The number of nights that the property will be reserved
14. persons: The persons that will live in the property
15. asksuitability: It should be used only in the case when there is a suitability item with a value 2 on the 'quantity' field, as described above in the reserve call.

The method performs the following:

- Cancels the already existing order (but without applying the cancellation policy that holds for the specified property)
- Inserts the new order with the updated values

It returns the result of the amendment and the new confirmation number if it was successful.

## D. GetAvailableInfo

The getAvailableInfo method can be used for multiple tasks, depending on its input variables:

- Get a list with the ids of all the available properties.
- Get all the available information for a certain property.
- Check whether a specified property has any changes in its information.

It takes as input the following variables:

1. id: The tour operator's id
2. villaid: The id of the property whose information are requested. (This variable is optional. If it is left empty, then the function returns the basic information[2] for all the properties that are available for the specified tour operator. )
3. check: This variable is **optional**. If you call this function and set this variable to 1, then the function will only return the villaid and the hash_key of each villa, so as you will be able to quickly check if there has been any changes in the specified villa.
4. getIds: This variable is **optional**. If you call this function and set this variable to 1, then the function will return the **ids** of the country-region-area output instead of their names. This may help you to organize your database easier.

It provides as output the following information:

- villaid: The code property
- hash_key: The hash key of the specified villa.
- name: The property name
- is_on_request: 1 if this property is on request (the owner must verify that he accepts the reservation), 0 otherwise.
- country: The country where the property is located
- region: The name of the region where the property is located
- area: The name of the area where the property is located
- street_address: The streed address where the property is located
- property_type: E.g. "villa", "apartment", etc…
- built: The year the property had been built
- renovated: The year the property had been renovated, or 0 if it has not been renovated
- size_SQM: Square meters of the building

[2] Basic information are the following: villaid, name, is_on_request, country, region, area, street_address, property_type, built, renovated, size_SQM, max_persons, floors_no, namegrounds_SQM, view, change_of_sheets, checkin_time, checkout_time.

- max_persons: The maximum number of persons that can live in the property
- floors_no: The number of floors
- grounds_SQM: Square meters of the ground
- view: The view of the property
- change_of_sheets: The frequency that the sheets are being changed
- checkin_time: The earlier time when the customer is allowed to check-in
- checkout_time: The latest time when the customer is allowed to check-out
- amenitiesList: An array whose items represent the amenities of the specified property. Each amenity has the variables below:
    - name: The amenity name
- attractionsList: An array whose items represent the attractions of the specified property. Each attraction has the variables below:
    - name: The attraction name
- bathroom_facilitiesList: An array whose items represent the bathroom facilities of the specified property. Each bathroom facility has the variables below:
    - name: The bathroom facility name
    - room: The room where this bathroom facility is. If the property has more than one bathrooms, then each bathroom is indicated by the symbol #i, where i is the number of the bathroom. E.g. bathroom #1, bathroom #2 etc…
- bedList: An array whose items represent the beds of the specified property. Each bed has the variables below:
    - name: The bed type
    - room: The room where this bed is. If the property has more than one bedrooms, then each bedroom is indicated by the symbol #i, where i is the number of the bedroom. E.g. single bedroom #1, single bedroom #2, double bedroom #1 etc…
    - quantity: The number of this types of bed that are in the specified bedroom
- distancesList: An array whose items represent the distances of the specified property from specified spots. Each distance has the variables below:
    - kind: The general spot (e.g. "from nearest airport")
    - name: The name of this spot
    - km: The distance counted in kilometers
- internetList: An array whose items represent the internet type of the specified property (if any). Each internet type has the variables below:
    - name: The internet type name
- kitchen_amenitiesList: An array whose items represent the kitchen amenities of the specified property. Each kitchen amenity has the variables below:
    - name: The kitchen amenity name
- leisure_activitiesList: An array whose items represent the leisure activities near the specified property. Each leisure activity has the variables below:
    - name: The leisure activity name
- local_area_activitiesList: An array whose items represent the activities of the local area where the specified property is. Each activity has the variables below:
    - name: The activity name

- location_typeList: An array whose items represent the type of the location where the specified property is. Each location type has the variables below:
    - name: The location type name
- nearby_attraction_facilitiesList: An array whose items represent the attraction facilities nearby of the specified property. Each attraction facility has the variables below:
    - name: The attraction facility name
- outside_amenitiesList: An array whose items represent the outside amenities of the specified property. Each outside amenity has the variables below:
    - name: The outside amenity name
- poolList: An array whose items represent the pools of the specified property. Each pool has the variables below:
    - name: The pool type
    - quantity: The number of the specified pool type
    - heated: 1 if the pool is heated, 0 otherwise
    - size: The size of the pool (in square meters)
- roomsList: An array whose items represent the rooms of the specified property. Each room has the variables below:
    - name: The room name
    - quantity: The number of these rooms
- roomsacList: An array whose items represent the air condition units of the specified property. Each air condition unit has the variables below:
    - name: The room name where the air condition unit is placed. If the property has more than one rooms with the same name, then we distinguish these rooms by the symbol #i (like we indicated them in bathroom_facilitiesList and bedList)
- stuffList: An array whose items represent the stuff of the specified property. Each stuff has the variables below:
    - name: The stuff name
- suitabilityList: An array whose items represent the suitabilities of the specified property. Each suitability has the variables below:
    - name: The suitability name
    - quantity: 0 if it is not accepted, 1 if it accepted, and 2 if you should ask the owner
- imageList: An array whose items represent the images of the specified property. Each image has the variables below:
    - path: The path where the specified image is stored
    - priority: The priority of this image. (The main image of the property has priority number 1)
- extraList: An array whose items represent the extras of the specified property
  Each extra has the variables below:
    - name: The name of the specified extra
    - price: The price of the specified extra
    - currency: The currency of this price
- terms: The terms and conditions of the specified property

- description: The description of the specified property
- instructions: The instructions of the specified property
- googleCoordinates: An array which represents the google coordinates of the specified property:
    - latitude
    - longitude
    - is_accurate: 1 if these coordinates point to the exact spot of the property. 0 if these coordinates are quite close to the spot

The lists above have some standard values in which we have been based, in order to describe each villa. However some rare times a property may have a list item which is not in the list of standard values. The reason we don't put it in this list is because the owner needs us to show this value, but it is not often met in properties. So, if you use the lists of standard values, you should always know that you may meet a value not belonging in this list.

You will find this list in the file "list_of_standard_items.xls"

## E. GetPriceAvailability

The **getPriceAvailability method** is used to check if a property is available for a specified date and retrieve the price for a certain reservation. Its use is optional for the tour operator if the rates are already agreed and locally stored, since the availability can also be checked through the *reserve* method.

It takes as input:

1. id: The tour operator's id
2. propertycode: The code property
3. checkindate: The check-in date
4. checkintime: The check-in time
5. checkouttime: The check-out time
6. numberofnights: The number of nights that the property will be reserved
7. persons: The persons that will live in the property

It provides as output the following information:

1. is_available: 1 if the property is available, 0 otherwise
2. price: The price of the reservation
3. currency
4. offer: A string that describes the offer that could be in this property-date (if any)

## F. GetInfoAvailabilityList

The **getInfoAvailability method** is used to return all the available properties in a specified time period, and their corresponding prices and basic information. You can use this function with different criteria.

It takes as input:

1. id: The tour operator's id
2. country: The country search criterion (optional)
3. region: The region search criterion (optional)
4. area: The area search criterion (optional)
   (Although the above 3 fields are optional, you **must** give one of them as input)
5. checkindate: The check-in date (optional)
6. nightsno: The number of nights that the property will be reserved (optional)
   If checkindate is given, then nightsno **must** also be given as input)
7. personsno: The persons that will live in the property (optional)
   (if checkindate is given but not personsno, then we consider that personsno=2 by default)
8. pool: Write the word "pool" in this field, in case you want to get only properties with pool. (optional)
9. internet: Write the word "internet" in this field, in case you want to get only properties with internet access. (optional)
10. sortby: (optional) Sort the output according to the distance you specify through this variable. You may fill this variable with one of these words:
    a. airport
    b. bar
    c. beach
    d. city
    e. clinic
    f. hospital
    g. port
    h. restaurant
    i. shopping
    j. supermarket

    For example if this variable equals to beach then the output is sorted according to its distance from the nearest beach. The properties that have no distance from the beach stored are returned in the end of the list.

    If this variable is not null, the output contains the variables: distancename and distancekm (see next page)

11. limit_from: (When this function is used, the user may want to group the results in pages, for time and space reasons. Therefore we always return the results beginning from limit_from and ending to limit_from+limit_amount. (limit_from begins from value 0)

If the results are N in total, where N<(limit_from+limit_amount) and N>=limit_from then we return these results [limit_from, N].
If N<limit_from then we return an empty response. )

12. limit_amount
13. price_per_day: If you choose a preffered **daily** price, then only the properties with the specified price will be returned. (optional)
14. price_diversion: If you choose a diversion for the above daily price, then only the properties with price P will be returned, where
    (price_per_day*(1-price_diversion))<=P<=(price_per_day*(1+price_diversion))
    Always remember that price_diversion should be a decimal number less than 1, because it represents the percentage of diversion. (optional)
15. offer_only: If you choose to see only properties with offer. It applies only if you also give *checkindate* and *nightsno* as input. (optional)
16. no_price: It is set to 1 if you don’t want us to compute and return the price of a property, even if you have provided *checkindate* and *nightsno* as input. It can be useful if you combine giving offer_only as input too. The combination of these two inputs returns all the properties that have offers for the period defined by *checkindate* and *nightsno*, without restricting the property to be available during the whole season.

    For example you could request to see the properties with offer and set checkindate=’15/02/2013’ and nightsno=’365’. This will return all the properties that have offers for the next year (if you book it today) without excluding a property if it has some reserved days in this year.

It provides as output a list, whose items contain the following information:

1. villaid: The id of the available property
2. vname: The name of this property
3. is_on_request: 1 if this property is on request, 0 otherwise.
4. region: The region where this property is located
5. area: The area where this property is located
6. description: The description of this property
7. image: The path of the main image of this property
8. persons: The maximum number of persons that can stay in this property
9. bedroom_no: The total number of bedrooms of this property
10. bathroom_no: The total number of bathrooms of this property
11. price: The price of the specified property in the specified days (if you had given a checkindate and nightsno as input)
12. currency
13. distancename: Name of distance item. This is being returned if the input had the variable sortby declared (for details see previous page).

    If for example sortby equals to “beach” then this variable is the name of the nearest beach. If a property has not any information regarding distance from beach then this variable (and the next one) is null.

14. distancekm: Distance (in km). This is being returned if the input had the variable sortby declared (for details see previous page).
    If for example sortby equals to "beach" then this variable is the distance from the nearest beach. If a property has not any information regarding distance from beach then this variable (and the previous one) is null.
15. offer: A string that describes the offer that could be in this property-date (if any)

## G. GetPriceAvailabilityList

The **getPriceAvailability method** is used to return all the available properties in a specified time period, and their corresponding prices.

However, in case you want to show the user some basic characteristics or the available properties and not **only** the prices, it is advised to use the **getInfoAvailabilityList instead**, since you will show your results much faster**.**

Its use is optional for the tour operator if the rates are already agreed and locally stored, since the availability can also be checked through the *reserve* method.

It takes as input:

1. id: The tour operator's id
2. checkindate: The check-in date
3. numberofnights: The number of nights that the property will be reserved
4. persons: The persons that will live in the property

It provides as output a list, whose items contain the following information:

1. villaid: The id of the available property
2. price: The price of the specified property in the specified days
3. currency
4. offer: A string that describes the offer that could be in this property-date (if any)

## H. GetYearAvailability

The **getYearAvailability method** is used to return the availability of a property during a whole year.

It takes as input:

1. id: The tour operator's id
2. propertycode: The code property
3. year: The year whose reservation days will be sent

It provides as output the following information:

1. availabilitystring: A string of length 365. Each character represents a specified date, beginning from the 1st of January of the input year, and ending to the 31st of December of the specified year. Each character has one of the following values:
    a. 0: available
    b. 1: booked
    c. 2: on request
    d. 3: option
2. msg: An error message (if any)

### I. getPriceCatalogue

The **getPriceCatalogue method** is used to show the price catalogue of a specified property.

It takes as input:

1. id: The tour operator's id
2. propertycode: The code property

It provides as output the following information an array. Each array item represents a season, and its values are:

1. price
2. sfrom: The starting date of the specified season
3. sto: The ending date of the specified season
4. pax: The number of persons that correspond to the specified price, or NULL if the price is not depended from the number of persons.
5. min_stay: The minimum stay in the specified season

### J. GetAreaStructure

The **getAreaStructure method** is used to return the structure of Countries-Regions-Areas.

It takes as input:

1. id: The tour operator's id

It provides as output the country-region-area structure of our system.

Actually, it outputs not only the names and ids of all the available countries, regions and area, but it also provides the information "which region belongs to each country" and "which area belongs to each region".

The structure of the output is the following:

1. countryList: An array which represents the list of the countries, with all the necessary information:

    a. id: The id of the specified country
    b. name: The name of the specified country

2. regionList: An array which represents the list of the regions, with all the necessary information:
    a. id: The id of the specified region
    b. name: The name of the specified region
    c. country_id: The id of the country where it belongs to.
3. areaList: An array which represents the list of the areas, with all the necessary information:
    a. id: The id of the specified country
    b. name: The name of the specified country
    c. region_id: The id of the region where it belongs to.

### K. GetGlobalHash

The **getGlobalHash method** is used to check if there has been any change in any property. It is useful in order to save time: If it returns the same hash key with the previous time you used this function, this means that all your villas information are updated and you don't need to do any other checks.

It takes as input:

1. id: The tour operator's id

It provides as output:

1. hash: the global hash key of the properties information

## Testing

For the testing period, the following information must be considered.

- The address for testing is http://www.leaderstay.com/testing/leaderstay.php
- The value corresponding to the tour operator **id** field must be requested from our Support Team. An alternative is to use "1" (without the quotes) as id.

When testing is over the following steps must be taken to switch to the live system.

- Switch LeaderStay address to: http://www.leaderstay.com/leaderstay.php
- Request the value corresponding to the tour operator id from our Support Team.