



Technical Manual regarding the Web Services provided
by Navins

Navins Web Service Documentation

Web Data Management

DiastasyS Ltd

Contents

1.	Introduction	2
2.	Reference Data	3
3.	Methods	4
4.	Custom Fields.....	8
5.	Static Data.....	9
5.1	Trailer.....	9
5.2	Discount Type	9
5.3	Owner Type.....	9
5.4	Partner Type.....	9
5.5	Make Type	9
5.6	Construction Material.....	10
5.7	Unit of Measure (LOA)	10
5.8	Unit of Measure (Beam).....	10
5.9	Unit of Measure (Draft).....	10
5.10	Fuel Type.....	10
5.11	Convertible.....	11
5.12	Engine Type.....	11
5.13	Gender	11
5.14	Relation Type.....	11
6.	Appendix.....	12
6.1	Versions.....	12

1. Introduction

DIASTASYS provides Web Services, through Navins, which allow partners to invoice offers of the automotive industry. The current technical manual describes in detail the methods used by the Web Service, to obtain the codes for the items used by each individual Web Service.

To consume the Web Service of Navins, it is necessary to use the URL which executes the call to WS. The additional username and password which will be sent to you by the company.

For any information that is not covered in this document, you can contact the IT department of our company on Phone 210 68 90 400.

2. Reference Data

Web Service "NavInsDataMgt" gives the ability to download codes for Web Services objects via web. The Web Service functionality is described in the following paragraphs.

3. Methods

The following paragraph lists all the methods that can be called, along with their additional arguments. In every method, depending on the method, `iTable` is an object of type `NavInsProposal.Root/ NavInsProposal.ROOT1/ NavInsProposal.Root2` and is being sent as a reference for each method and to which the values for each object are assigned.

`AttachedDocumentTypes(ref Root iTable)`
Returns all file types that can be attached.

`Banks(ref Root iTable)`
Returns all bank codes.

`Brokers(ref Root iTable)`
Returns all broker codes.

`Charges(ref Root2 iTable, string insCategoryCode, string insCompanyCode)`

Returns all item charge codes of the system. The corresponding parameters of the method the industry code and the insurance company code.

Attention, each packet gets different charges, which you can be get through the `Chargesbypackage` method. This method should be used as a reference to identify the codes of the automatic charges that are returned by the system.

`ChargesByPackage(ref Root2 iTable, string packageCode, string insCategoryCode, string insCompanyCode)`

Returns the item charges for a specific package. The corresponding parameters of the method are the package code, the industry code and the insurance company code. You can get the additional codes by calling the additional methods for each object.

`Colors(ref Root iTable)`
Returns all color codes.

`CoverCategories(ref Root iTable, string insCategoryCode)`
Returns all codes of the cover categories. The corresponding parameter is the insurance industry code.

`Covers(ref Root1 iTable, string insCategoryCode, string insCompanyCode)`
Returns all covers for a specific insurance category and a specific insurance company. The corresponding parameters of the method are the insurance industry code and the insurance company code. As in the case of charges, the method returns all covers and should be used as reference. To retrieve the covers of a package you must call the `Coversbypackage` method.

CoversByPackage([ref Root1](#) iTable, [string](#) packageCode, [string](#) insObjectTypeCode, [string](#) insObjectTypeGroupCode, [ref DateTime](#) referenceDate)

Returns the covers of each packet. The corresponding parameters of the method are the package code, the insured object usage code, the insured object type group code and a reference date. The last parameter specifies the desired date of the package covers that you want to retrieve. If no value is assigned in the referenceDate parameter, then the method will use today as the additional date.

Notes:

The insObjectTypeGroupCode parameter is not mandatory and it should only be populated if the corresponding usage has groups of insured objects (the codes group types are returned by the InsObjectTypeGroups method). If there are no related codes, the parameter must be empty.

The method does not return the funds and the exemptions of the covers per packet. These amounts are user-defined and are sent by the company with the underwriting rules.

There are three types of covers: optional, Mandatory and Package Optional. Mandatory are the mandatory covers of the package, optional are the optional covers of the package, and package Optional are covers that are part of the package but there is a possibility of removal (so by a technical perspective, this makes them optional like the rest).

CustomerDiscountGroups([ref Root](#) iTable)

Returns the discount groups that apply to customers.

Discounts([ref Root2](#) iTable, [string](#) insCategoryCode, [string](#) insCompanyCode)

Returns all the discount codes available in the system. The corresponding parameters of the method are the insurance industry code and the insurance company code.

Attention, each package contains different discounts, which you can retrieve using the DiscountsByPackage method. This method should be used as a reference to identify the automatic discount codes that are available in the system.

DiscountsByPackage([ref Root2](#) iTable, [string](#) packageCode, [string](#) insCategoryCode, [string](#) insCompanyCode)

Returns the discounts for a specific package. The corresponding parameters of the method are the package code, the insurance industry code and the insurance company code. You can retrieve the additional codes calling the corresponding methods for each object.

DurationTypes([ref Root](#) iTable)

Returns the codes for the policies durations.

EndorsementReasons([ref Root](#) iTable)

Returns the codes for the endorsement types.

InsCategories([ref Root](#) iTable)

Returns the codes for the insurance categories.

InsCompanies([ref Root](#) iTable)

Returns the codes to the insurance companies.

InsObjectTypeGroups([ref Root](#) iTable, [string](#) insCategoryCode, [string](#) insSubCategoryCode)

Returns the codes for the insured object type groups. The corresponding parameters of the method are the insurance category code and sub-category insurance code.

InsObjectTypes([ref Root](#) iTable, [string](#) insCategoryCode)

Returns the types of insured objects.

InsSubCategories([string](#) username, [string](#) password, [ref Root](#) iTable, [string](#) insCategoryCode)

Returns the codes for the insurance sub-categories. The corresponding parameter of the method is the insurance branch.

Makes([ref Root](#) iTable)

Returns the make codes.

MakesByType([ref Root](#) iTable, [string](#) Type)

Returns the make codes per category. Each make may belong to a single category of 'Vehicle-Moto', 'Vessel', 'Engines' or belong to 'All Categories '. The make codes for each category are defined in the "Static Data" section of the manual.

Example:

- If the selected category is "All Categories", then the method will return all the makes of the database.
- If the selected category is "Vehicles-Moto", then the method will return all the makes that belong to the category "Vehicles-Moto" and all the makes that belong to the category "All Categories".

Models([ref Root](#) iTable, [string](#) makeCode)

Returns all the model codes for a specific make. The corresponding parameter is the make code.

Packages([ref Root](#) iTable, [string](#) insCategoryCode, [string](#) insCompanyCode, [string](#) insObjectTypeCode)

Returns the codes of the available packages of a company. The corresponding parameters are the insurance category code, the insurance company code and the insured object type code.

Professions([ref Root](#) iTable)

Returns the profession codes.

TaxOffices([ref Root](#) iTable)

Returns the tax office codes.

VehicleTypes([ref Root](#) iTable)

Returns the vehicle type codes.

VesselTypes([ref Root](#) iTable)

Returns the vessel type codes.

ModelTypesSearch (string makeCode, string modelNo, string makeDescription, string modelDescription, string typeDescription, integer cylinderCapacity, integer taxableHP, integer builtYear, integer builtMonth, ref ModelTypes iTable)

Returns results from Model Types database.

The allowed values of builtMonth parameter are 1 to 12 and limit the search to month-level. Any different value is being ignored and the search is done by year.

ModelTypesSearchWithPrices (string makeCode, string modelNo, string makeDescription, string modelDescription, string typeDescription, integer cylinderCapacity, integer taxableHP, integer builtYear, integer builtMonth, ref ModelTypes iTable)

Returns results from Model Types database along with the suggested vehicle value (if there are any).

The allowed values of builtMonth parameter are 1 to 12 and limit the search to month-level. Any different value is being ignored, and the search is done by year.

4. Custom Fields

Custom Fields are some extra information needed for a specific insurance. They are elements related to the insured object and are customizable per industry or per package. General information are mentioned below, along with the additional workflow that you must follow in order to populate the fields properly.

By calling:

CustomFields([ref Root](#) iTable)

You can see the list of all available custom fields. The method returns the code, the description, and the type of the field. The field type can be Alphanumeric, Integer, Decimal, Date, Time, or Fixed Value. If the formula is Fixed Value, then you can get the valid values of the custom field using the method:

CustomFieldsValues([ref Root](#) iTable, [string](#) Code)

The corresponding parameter for this method is the code of the custom field. It will return the additional valid values.

Custom Fields Groups.

Custom fields are grouped in Templates, depending on the use of the insured object or the package. For the correct billing of a quote, you must fill in the custom fields of the corresponding template.

The complete set of the templates can be seen by calling the method:

CustomFieldsTemplates([ref Root](#) iTable)

This method returns the codes of the customer fields templates. You can see the custom fields that are included in a template, by calling the method:

CustomFieldsPerTemplate([ref Root](#) iTable, [string](#) templateCode)

This method returns the codes of the custom fields for a specific Template. The corresponding parameters is the template code.

5. Static Data

5.1 Trailer

Code	Description
0	Blank
1	2 wheels
2	4 wheels
3	Vessel
4	Caravan

5.2 Discount Type

Code	Description
0	Discount
1	Charge

5.3 Owner Type

Code	Description
0	Driver
1	Customer
2	Bank

5.4 Partner Type

Code	Description
0	Blank
1	Company
2	Person

5.5 Make Type

Code	Description
0	All Categories
1	Vehicles - Moto
2	Vessels
3	Engines

5.6 Construction Material

Code	Description
0	Steel
1	Aluminum
2	Wood
3	Fiberglass
4	PVC
5	Plastic
6	Polyester
7	Other

5.7 Unit of Measure (LOA)

Code	Description
0	Meters
1	Feet

5.8 Unit of Measure (Beam)

Code	Description
0	Meters
1	Feet

5.9 Unit of Measure (Draft)

Code	Description
0	Meters
1	Feet

5.10 Fuel Type

Code	Description
0	Blank
1	Petrol
2	Diesel
3	Electric
4	LPG
5	Modification to LPG

6	CNG
7	Modification to CNG

5.11 Convertible

Code	Description
0	Blank
1	Hard Top
2	Soft Top
3	No Top

5.12 Engine Type

Code	Description
0	Conventional
1	Hybrid

5.13 Gender

Code	Description
0	Blank
1	Male
2	Female

5.14 Relation Type

Code	Description
0	Child
1	Wife
2	Protected Member

6. Appendix

6.1 Versions

August 2016

- Notes added in CoversByPackage method.

October 2016

- Custom Fields section added.

January 2017

- New parameter added in CoversByPackage, along with the additional callout.

November 2022

- Addition of ModelTypesSearch & ModelTypesSearchWithPrices methods.