



Data Management
& Customer Care

Interface documentation

Web service AVS registration form

Status: October 1st, 2018

Table of Contents

1	General	3
2	Security	4
2.1	TAG generation	4
2.2	Interface.....	4
2.3	Spa administration	4
2.4	User ID	4
2.5	Session ID	5
2.6	Processing.....	5
2.6.1	Currently valid processing code	5
2.7	Version	5
3	Interface description	6
3.1	Interface format	6
3.2	Operation get registration form:	8th
3.2.1	Object get registration form	8th
3.2.2	Meaning of the fields of the object "getMeldeschein"	8
3.2.3	Object registration form	9
3.2.4	Meaning of the fields of the object "meldeschein"	11
3.3	Book operation registration form	25
3.4	Print operation registration form	26
3.4.1	Print registration formsRequest	26
3.4.2	Meaning of the fields of the object "meldeschein"	26
3.4.3	print registration forms response	27
3.4.4	Meaning of the field of the object "ms-pdf"	27
3.5	Method getKonfigurationsListe	27
3.5.1	Method configuration lists	28
3.5.2	Meaning of the fields of the object "configuration lists"	29
4th	Behavior in the event of an error	30
5	Use cases	32
6th	SOAP request for the function "bucheMeldeschein"	39
7th	SOAP request "getMeldeschein"	42
7.1	Request get registration form with booking number	43
7.2	Request get registration form with registration form number	43
7.3	SOAP response "getMeldeschein"	44
8th	SOAP request "getKonfigurationsListe"	46
9	Examples	48
9.1	Request book registration form group registration form	48
9.2	Response faulty login	49
10	Support requests	49
11	Related sources:	49

1. General

This document describes the data exchange via the web service "JMeldescheinWebservices" of the registration form system of AVS GmbH in Bayreuth. The interface is provided by AVS as a web service. It represents the point at which all transactions in the AVS registration card system are triggered. The requirements for this web service are usually made by a technical system.

The web service expects a predefined request object, which must be filled with attributes according to the given schema definition. The object is validated after general partner authorization, which takes place via an http BasicAuthentication and is transferred to the background system for further processing. The result is sent to the calling technical system in the form of a response object. If a transaction fails, an error message is added to the response object.

The interface specification is passed on to the partner by AVS in the form of a WSDL file.

Left

WS url

<https://meldeschein.avs.de/meldeschein-ws-prod/JMeldescheinWebservices>

WSDL

<https://meldeschein.avs.de/meldeschein-ws-prod/JMeldescheinWebservices?wsdl>

Interface links without authentication

<https://meldeschein.avs.de/meldeschein-ws-prod/wsdl/JMeldescheinWebservices.wsdl>

https://meldeschein.avs.de/meldeschein-ws-prod/wsdl/meldeschein_webservices.xsd https://meldeschein.avs.de/meldeschein-ws-prod/wsdl/meldeschein_buchen.xsd

https://meldeschein.avs.de/meldeschein-ws-prod/wsdl/meldeschein_datotypen.xsd

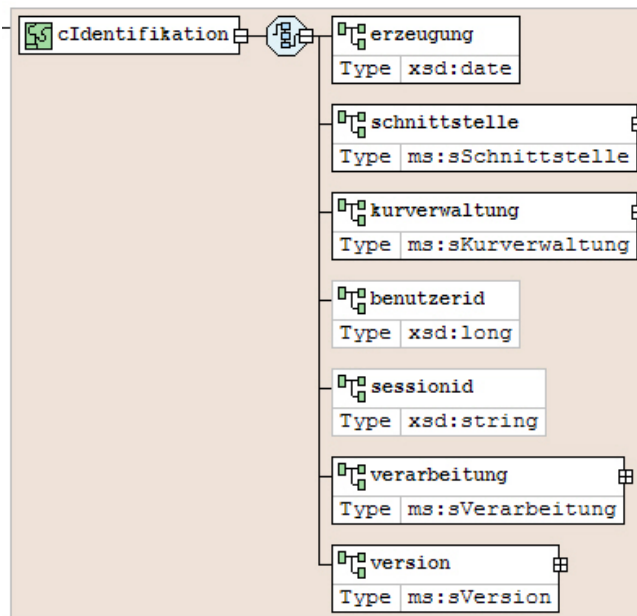
2 security

The web service is secured via HTTP authentication. The access data is communicated to the partner by the AVS.

The authentication takes place via http basic authentication.

The authorization takes place through the identification part of the SOAP body.

Structure of the identification part:



Example:

```
< ID >  
  < generation > 2016-07-01 < /generation >  
  < interface > XXX < /interface >  
  < spa administration > 1 < / spa administration >  
  < userid > 1 < / userid >  
  < sessionid > string < / sessionid >  
  < processing > MS-HOLEN < /processing >  
  < version > string < /version >  
< /ID >
```

2.1 TAG generation

This field contains the current date. Format according to XML schema type **xsd:date** • Example: **2016-07-12**

2.2 interface

Enter the user name of the WS user for authorization here. The user name is agreed with the respective interface operator as part of the initial connection.

The user name decides which functions and communities can be called and which cannot (error code 11900).

2.3 Spa administration

This field contains the corresponding ID of the spa administration (client / location ID) for which the following action is carried out. It is the internal identification key to the associated location.

2.4 User ID

This attribute optionally contains a user ID. The user of the hotel reservation system can use transferred here.

2.5 session ID

This field contains a session key generated uniquely by the partner, which can be used for research purposes. (HTTP SessionId for web applications).

2.6 Processing

This attribute is used to differentiate which function is to be triggered via the corresponding processing indicator.

2.6.1 Currently valid processing code

- **MS-HOLEN** (Retrieving a complete registration form)
- **BOOK** (Booking a registration form)
- **MS PRINTING** (Printing a registration form)
- **CONFIG-L- < identifier>** (Calling up various configuration lists)

2.7 version

Contains the version number of the web service, the current version is returned by AVS in the respective response.

3 Interface description

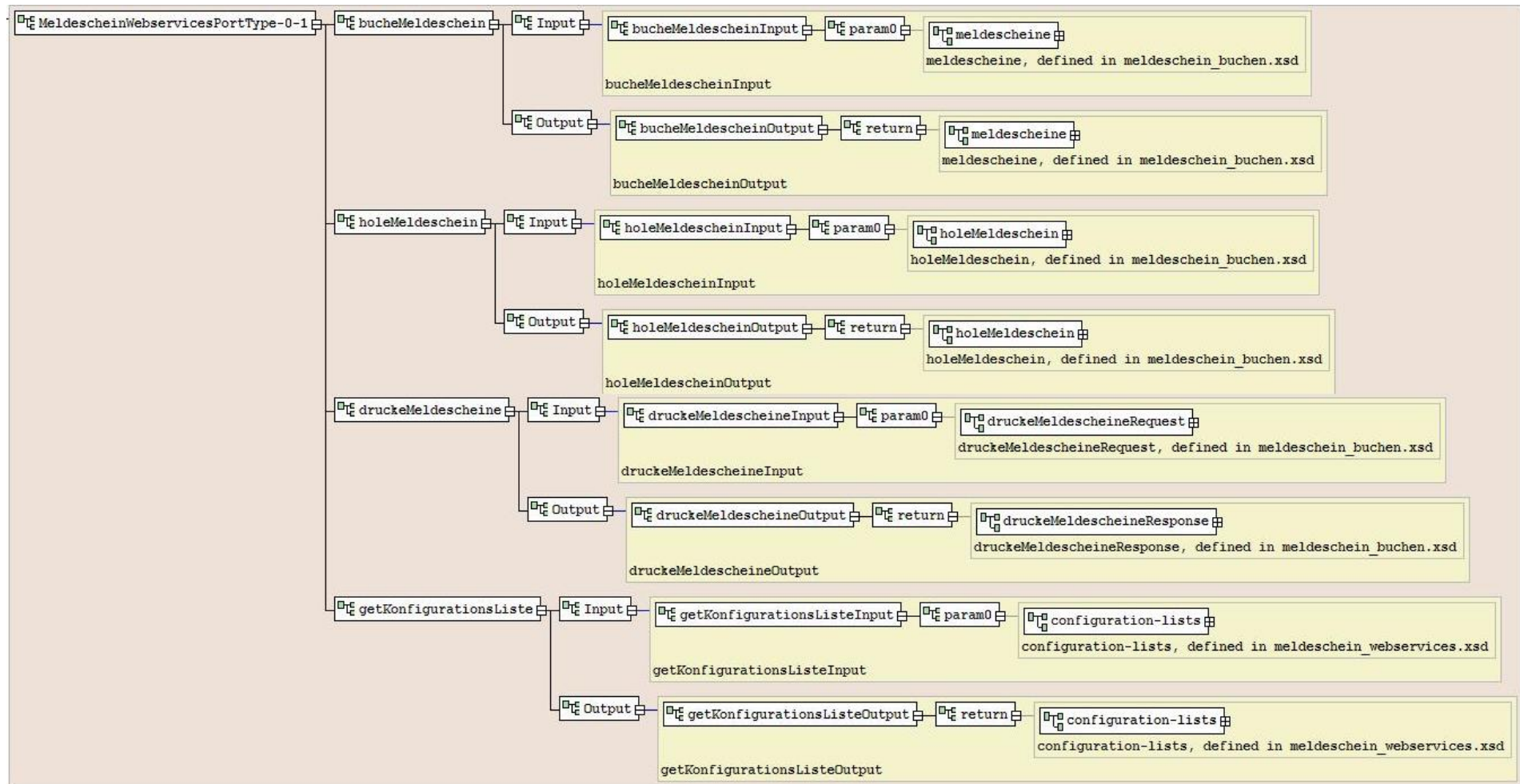
3.1 Interface format

The interface is provided by AVS as a web service.

The interface includes the following functions

- **get registration form**
This allows the act. Registration form data can be obtained. Also makes the update process easier (see [book registration form](#))
- **book registration form**
Creates or changes a registration form
- **Print registration form**
Generates the registration form PDF
- **getConfiguratiOnList**
Retrieves various configuration lists in order to fill the other web service functions with the correct ID's (salutation ID, company ID, etc.)

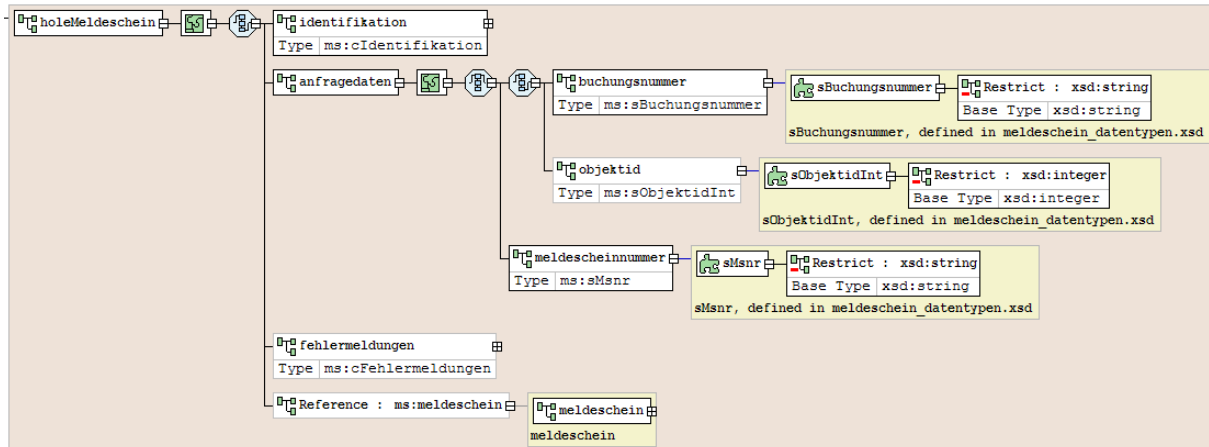
Schematic diagram of the web service functions (WSDL port types):



3.2 Get operation registration form:

entrance	output
get registration form	get registration form • registration form

3.2.1 Get object registration form



3.2.2 Meaning of the fields of the object "getMeldeschein"

The search parameters are filled in a request block in this object. In the response object, the request element is expanded to include the elements " *error messages* " and " *registration form* " expanded.

3.2.2.1 Request data

In the inquiry data, the registration form can either be based on the booking number + ObjektId **or** the registration number can be obtained.

3.2.2.2 Booking number

The booking number assigned by the reservation system (unique throughout the property!). Search queries always in connection with <objektid>.

3.2.2.3 ObjektId

The unique identification key of the accommodation object assigned by the AVS registration form. Search queries always in connection with <booking number>.

3.2.2.4 Registration form number

Any known registration number assigned by AVSMeldeschein.

3.2.3 Object registration form

In the return element for successful inquiry requests, the element meldeschein is delivered. Elements framed in black are absolutely necessary fields for the correct booking (mandatory fields), the remaining fields are optional.



3.2.4 Meaning of the fields of the "registration certificate" object

3.2.4.1 Booking number

Contains the booking number of the reservation system, which is unique for each "client + object".

Special case of manual registration form:

Manual registration forms can be re-entered as a "special process". This usually only makes sense through the mandate (spa administration, tourist information). If you are interested in this process, please contact AVS directly first.

If a registration form is created for a manual MS company, the manual MS number maintained in the AVS system must be entered here.

3.2.4.2 kz_manuelle_ms

If a manual MS number is booked, as described in section 3.2.3.1 for the "special case", this *boolean* Flag to inform the system of this directly. In this case it is not necessary to transfer the company and object ID, these values are then determined via the configuration in the AVS registration card system.

3.2.4.3 Registration form number

Contains the unique registration number for each client assigned by the AVS registration card system.

3.2.4.4 CompanyId

The company's unique identification key from the AVS registration form.

3.2.4.5 ObjectId

The unique identification key of the accommodation object assigned by the AVS registration form (a company can own one or more objects (its own or brokered)).

3.2.4.6 Arrival

The main person's arrival date.

3.2.4.7 Departure

The main person's departure date.

3.2.4.8 CategoryId

Contains the unique identification key of the person category (e.g. adult, child, business traveler, handicapped) assigned by the AVS registration form for each client. You can obtain the configuration list (interface identification) from the spa administration.

3.2.4.9 Salutation ID

Contains the unique identification key of the salutation assigned by the AVS registration form for each client.

3.2.4.10 Name

Name of the main person.

3.2.4.11 First name

First name of the main person.

3.2.4.12 Street

Street of residence of the main person.

3.2.4.13 House number

House number of the main person's place of residence.

3.2.4.14 zip code

Postal code of the main person's residence.

3.2.4.15 Place

Place of residence of the main person.

3.2.4.16 Country

For the main person on the registration form, it is possible to transfer a country (domicile of the main person) that deviates from the basic setting:

Will be used within the `parent-TAG`'s < country >` the `child-TAG < country id >` **or** the `childTAG < land-iso >` then it is checked whether there is a value within this `childTAG`'s` has been passed and whether this value is valid. If the value is valid, it is accepted.

Will be used within the `parent-TAG`'s < country >` the `child-TAG < country id >` **and** the `child-TAG < land-iso >` then the `child-TAG < country id >` checked. If the transferred value is valid, this value is accepted and saved. If the transferred value is not transferred correctly or is empty, then the `child-TAG < land-iso >` checked. If the transferred value is valid, this value is accepted and saved.

Are both `child-TAG`'s` filled with invalid values or passed empty, then a check is made whether the individual `DAY < landid >` is present and filled. Is the value in `DAY < landid >` valid, then this is adopted, otherwise the preset default value is automatically adopted and saved.

The ones in the `TAG's < country >` **and** `< landid >` IDs to be transferred can be called up using the L-CONFIG-LAND method.

Examples for the handover:

1)

```
< country >  
  < country id > 1 < / country-id >  
< /country >
```

The country ID in the DB (and which can be called up via CONFIG-L-LAND) is adopted.

2)

```
< country >  
  < land-iso > DE < / land-iso >  
< /country >
```

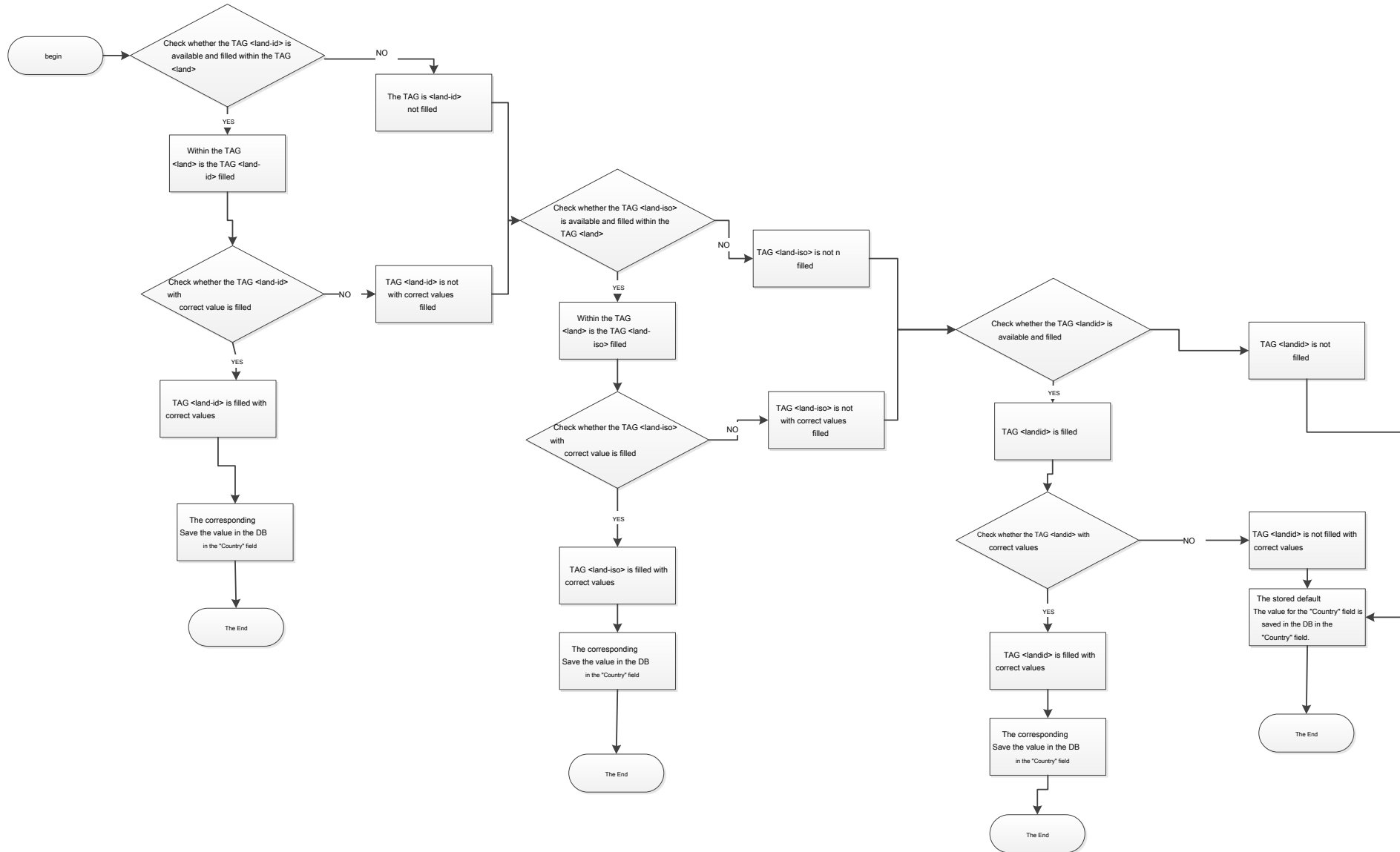
3)

```
< country >  
  < country id > 1 < / country-id >  
  < land-iso > DE < / land-iso >  
< /country >
```

4)

```
< landid > 1 < / landid >
```

Flowchart:



3.2.4.17 Nationality

Will be used within the parent-TAG's < nationality > the child-TAG < country id > or the child-TAG < land-iso > then it is checked whether there is a value within this child-TAG's has been passed and whether this value is valid. If the value is valid, it is accepted.

Will be used within the parent-TAG's < nationality > the child-TAG < country id > and the child-TAG < land-iso > then the child-TAG < country id > checked. If the transferred value is valid, this value is accepted and saved. If the transferred value is not transferred correctly or is empty, then the child-TAG < land-iso > checked. If the transferred value is valid, this value is accepted and saved.

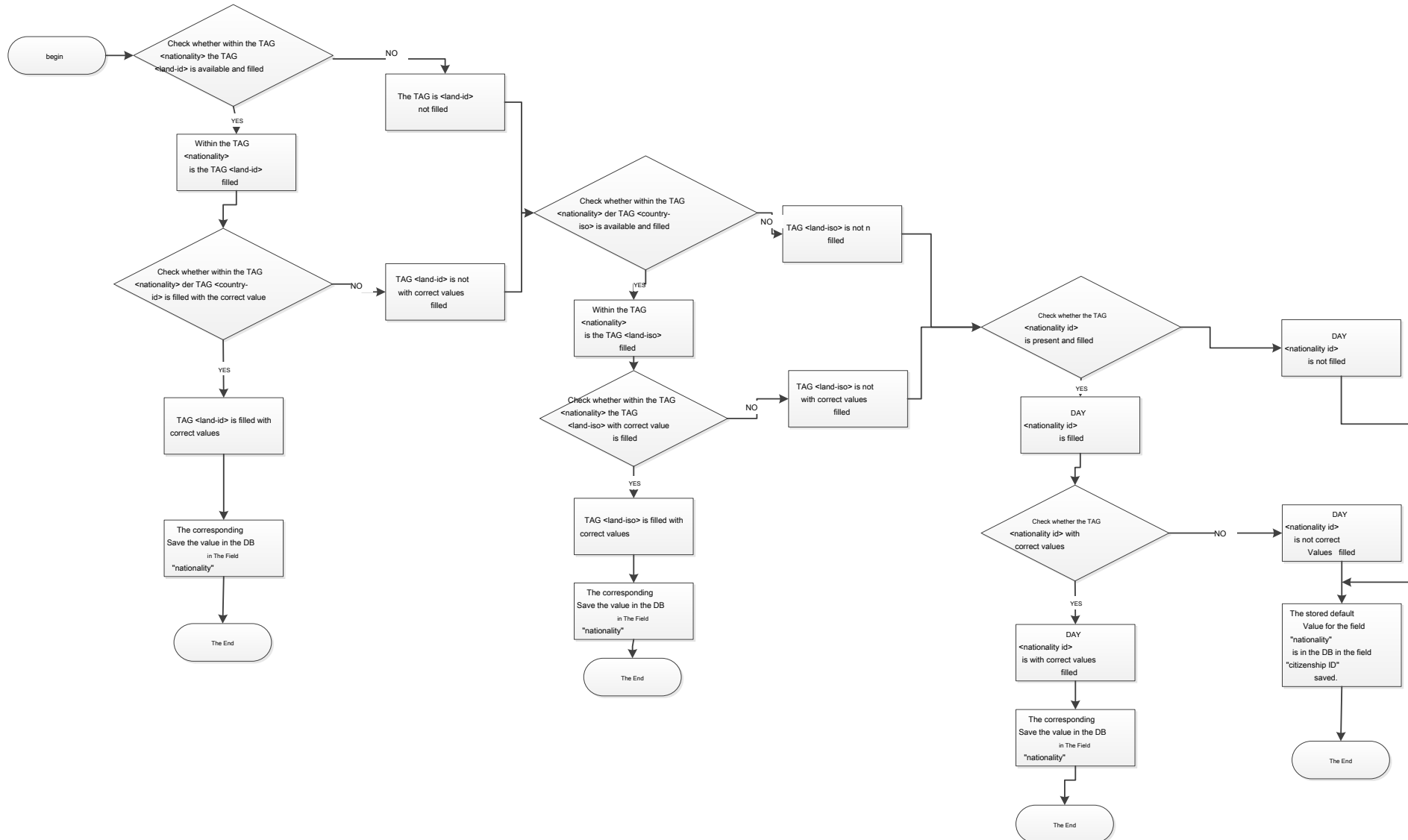
Are both child-TAG's filled with invalid values or passed empty, then a check is made whether the individual DAY < nationalityid > is present and filled. Is the value in DAY < nationalityid > valid, then this is adopted, otherwise the preset default value is automatically adopted and saved.

The ones in the TAG's < country > and < landid > IDs to be transferred can be called up using the L-CONFIG-LAND method.

Examples of the handover:

- 1)
< nationality >
 < country id > 1 < / country-id >
< /nationality >
- 2)
< nationality >
 < landi-iso > DE < / land-iso >
< /nationality >
- 3)
< nationality >
 < country id > 1 < / landid >
 < landi-iso > DE < / land-iso >
< /nationality >
- 4)
< nationalityid > 1 < nationalityid >.

Flowchart:



3.2.4.18 ID number (ID card number)

The ID number / ID number can be provided for the main traveler and the accompanying person (s).

3.2.4.19 Further information

Field for further information.

3.2.4.20 License plate

Vehicle registration number of the person arriving.

3.2.4.21 Date of birth

Date of birth of the main person.

3.2.4.22 Amount

The amount of the registration form calculated by the AVS registration form.

If the amount is in DAY < registration form > then the amount refers to the entire registration form, not just to the main person.

If the amount is with the accompanying person or accompanying group, it is the amount only for the respective accompanying person or accompanying group.

3.2.4.23 Billing status Id

The status of the registration form assigned by the AVS registration form.

The following values / status are possible:

1	Created
2	Printed
3	Billed
4th	Canceled
5	Paid
6th	Invalid

3.2.4.24 Category-related additional fields

The municipalities / clients can have specially predefined additional fields set up for individual categories, which can also be filled in via the web service.

You can inquire about the respective individual configuration from the municipality / client. The categories or fields that have been set up are also listed in the configuration overviews under the item "Category-related additional fields". For more information, see the PDF document "AVS Information Letter Interfaces Hotel Programs".

Category-related additional fields are only possible for the main person. There are no category-related additional fields for accompanying persons and accompanying groups.

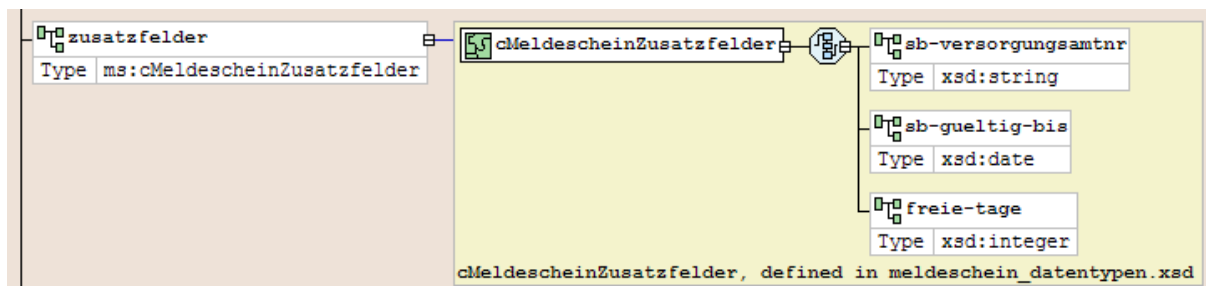
It is not possible to create a registration form in which the accompanying person or group is assigned a category with a category-related additional field. This registration form is rejected by the system or an error message is returned.

The additional fields are within the **parent-TAG`< additional fields >** to hand over .

Inside this **parent-TAG`<** either one or more additional fields can be transferred. Becomes a **child-TAG** If you pass empty or with incorrect values, the value will not be transferred to the AVS registration form and the field will accordingly be displayed empty.

The following category-related additional fields are generally possible:

DAY	example
< sb-valid-until >	< sb-valid-until > 2017-01-15 </ sb-valid-until >
< sb pension office no >	< sb pension office no > ABC123 </ sb pension office no >
< company name >	< company name > company </ company name >
< free days >	< free days > 5 </ free days >



An example request with "all" additional fields:

```

< ? xml version = "1.0"? >
< ms: registration cards xmlns: ms = "http://www.avs.meldeschein.de/ns/"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < ID >
    < generation > 2016-07-01 </ generation >
    < interface > xxx </ interface >
    < spa administration > 1 </ spa administration >
    < userid > 1 </ userid >
    < sessionid > string </ sessionid >
    < processing > BOOK </ processing >
  
```

```

    < version > string </ version >
  </ ID >
  < ms: registration form >
    < booking number > 30501 </ booking number >
    < company id > 222 </ company id >
    < objectid > 222 </ objectid >
    < getting there > 2016-07-01 </ getting there >
    < departure > 2016-07-02 </ departure >
    < category ID > 2 </ category ID >
    < address > 1 </ address >
    < Surname > Test name </ Surname >
    < First name > Test first name </ First name >
    < road > Test street </ road >
    < House number > 1 </ House number >
    < Postcode > 12345 </ Postcode >
    < place > Test city </ place >
    < landid > 1 </ landid >
    < nationalityid > 1 </ nationalityid >
    < More information > 12345678 </ More information >
    < ID number > 12345678ABC </ ID number >
    < License Plate > BT TB-44 </ License Plate >
    < Date of birth > 1972-01-21 </ Date of birth >
    < additional fields >
      < company name > Test company </ company name >
      < free days > 5 </ free days >
      < sb-valid-until > 2017-01-21 </ sb-valid-until >
      < sb pension office no > ABC123 </ sb pension office no >
    </ additional fields >
  </ ms: registration form >
</ ms: registration cards >

```

Example of the transfer of a non-valid category (category-related additional field) to the accompanying person:

Request:

```

...
< Companion >
  < Surname > Test name </ Surname >
  < First name > Test first name </ First name >
  < category ID > 17th </ category ID >
  < getting there > 2016-07-01 </ getting there >
  < departure > 2016-07-02 </ departure >
</ Companion >
...

```

Response:

```

...

```

```

< error messages >
  < error >
    < code > 10122 </ code >
    < description > Category invalid. Allowed for the main traveler only </ description >

    < reference > Registration number: BA1234422425 - Accompanying person: TestName </ reference >

  </ error >
</ error messages >
...

```

This registration form is rejected or not posted by the system.

3.2.4.25 Individual additional fields

The municipalities / clients can have individual additional fields set up, which can also be filled in via the web service.

The respective individual configuration of these additional fields can be requested from the municipality / client. The categories or fields that have been set up are also listed in the configuration overviews under the item "Individual additional fields".

The following is an excerpt from the PDF document with sample fields:

Individuelle Felder Übersicht:		
Name	Typ	Max. Länge der Eingabe
Reisegrund	Combobox	Vorgegebene Werte
Vermerk	Text-Field	30

Vorgegebene Werte für das Feld Reisegrund

Wert
Messe
Kongress
Business
Freizeit

For more information, see the PDF document "AVS Information Letter Interfaces Hotel Programs".

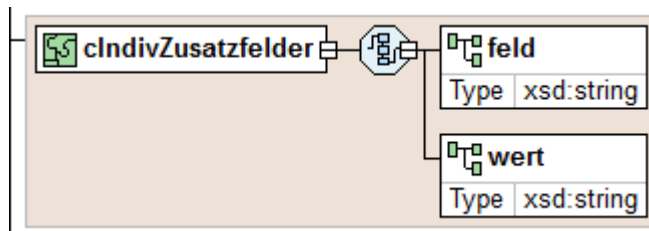
Individual additional fields are generally only possible for the main person. There are no individual additional fields for accompanying persons.
It is therefore not possible to create a registration form in which the accompanying person is assigned a category with an individual additional field.

The individual additional fields are each within the **parent-TAG's** **< individual additional fields >** to hand over . Inside this **parent-TAG's** only one additional field can be transferred at a time. A transfer of several individual additional fields is possible, but then this must always be done within **parent-TAG's** respectively.

There are two different "types" of individual additional fields:

- Comboboxes; Input / transfer of specified / firmly defined values
- Text field; "Free text" input field

Type	Field (example)	Values (examples)	TAG's (examples)
Combobox travel reason		<ul style="list-style-type: none"> • <i>Empty</i> • Fair • congress • leisure 	< individual additional fields > < field > Reason for travel </ field > < value > congress </ value > </ individual additional fields >
Text field	Note	Regular guest	< individual additional fields > < field > Note </ field > < value > Regular guest </ value > </ individual additional fields >



With the "Comboboxes" type, the specified / fixed values must be transferred exactly as they are created in the AVS registration form or as shown in the "Configuration overviews". In the event of deviations, no registration forms are posted and an error message is returned.

"Empty" can also be entered or transferred as a value here.

The transfer of the identifiers in "Field" and "Values" are "case-sensitive", so upper and lower case letters must be observed.

The exception is the value in the text field. This can of course be freely selected.

The booked fields and values are returned in the response. When updating / changing the registration form, please note that the values that have already been booked must always be transferred.

Example of the transfer of the value "Empty" with the type "Combobox":

```

...
< individual additional fields >
  < field > Reason for travel </ field >
  < value > </ value >
</ individual additional fields >
...

```

Example of the transfer of several individual additional fields:

```

...
< individual additional fields >
  < field > Reason for travel </ field >
  < value > congress </ value >
</ individual additional fields >
< individual additional fields >
  < field > Note </ field >
  < value > Regular guest </ value >
</ individual additional fields >
...

```

Examples of disregarding "case-sensitive":

```

...
< individual additional fields >
  < field > reason for travel </ field >
  < value > congress </ value >
</ individual additional fields >
...

```

```

...
< individual additional fields >
  < field > Reason for travel </ field >
  < value > congress </ value >
</ individual additional fields >
...

```

If "case-sensitive" is not observed, the field / value cannot be recognized. The registration form is not posted; a corresponding error message is returned in the response:

```

...
< error messages >
  < error >
    < code > 10128 </ code >
    < description > Additional fields passed incorrectly. </ description > < reference > Registration
      number: BN123456 </ reference >
  </ error >
</ error messages >

```

Example of an incorrect value for the "Combobox" type:

```
...
< individual additional fields >
  < field > Reason for travel </ field >
  < value > Anything </ value >
</ individual additional fields >
...
```

"Something" is not defined / stored as a valid value. The value cannot be recognized. The registration form is not posted; a corresponding error message is returned in the response:

```
...
< error messages >
  < error >
    < code > 10128 </ code >
    < description > Additional fields passed incorrectly. </ description > < reference > Registration
    number: BN123456 </ reference >
  </ error >
</ error messages >
```

A (complete) example request with an individual additional field:

```
< ? xml version = "1.0"? >
< ms: registration cards xmlns: ms = "http://www.av.s.meldeschein.de/ns/"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < ID >
    < generation > 2018-09-01 </ generation >
    < interface > xxx </ interface >
    < spa administration > 1 </ spa administration >
    < userid > 1 </ userid >
    < sessionid > string </ sessionid >
    < processing > BOOK </ processing >
    < version > string </ version >
  </ ID >
  < ms: registration form >
    < booking number > BN0501 </ booking number >
    < company id > 123 </ company id >
    < objectid > 123 </ objectid >
    < getting there > 2018-10-01 </ getting there >
    < departure > 2018-10-02 </ departure >
    < category ID > 2 </ category ID >
    < address > 1 </ address >
    < Surname > Test name </ Surname >
    < First name > Test first name </ First name >
    < road > Test street </ road >
    < House number > 1 </ House number >
    < Postcode > 12345 </ Postcode >
```

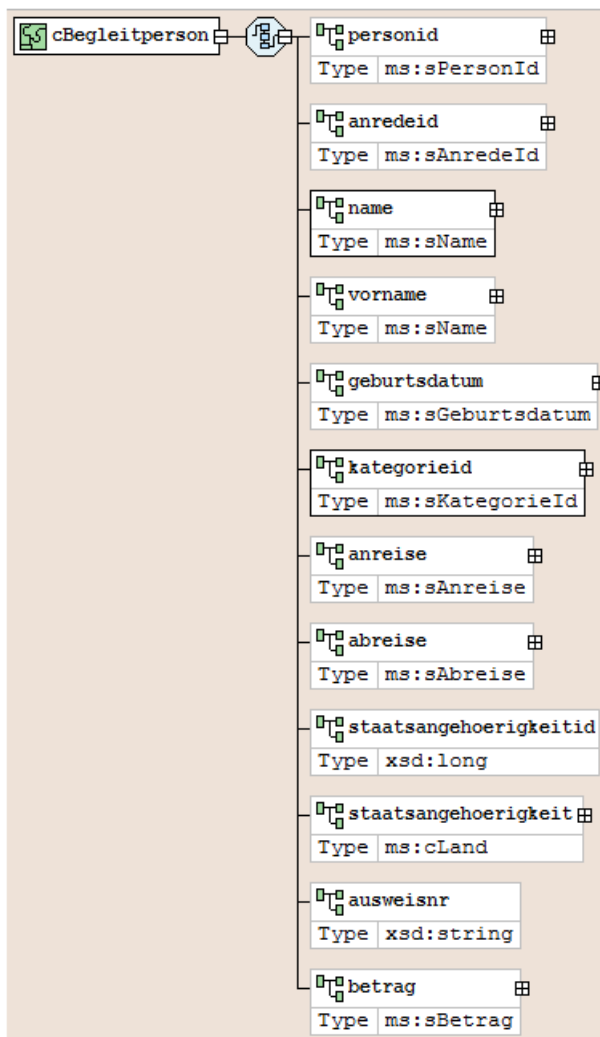
```

< place > Test city </ place >
< landid > 1 </ landid >
< nationalityid > 1 </ nationalityid >
< More information > 12345678 </ More information >
< ID number > 12345678ABC </ ID number >
< License Plate > XX XX-44 </ License Plate >
< Date of birth > 1972-01-21 </ Date of birth >
< individual additional fields >
  < field > Reason for travel </ field >
  < value > congress </ value >
</ individual additional fields >
</ ms: registration form >
</ ms: registration cards >

```

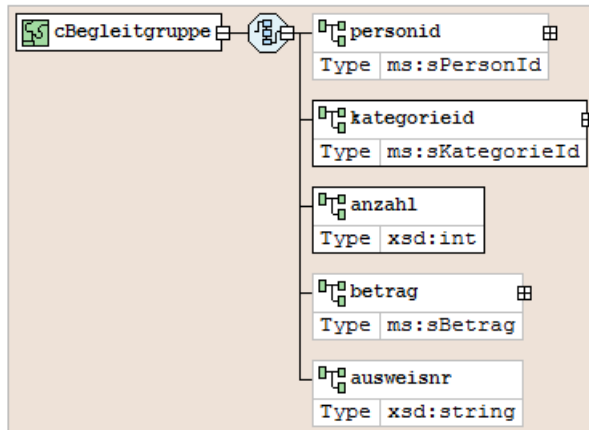
3.2.4.26 accompanying person

Object of the data type accompanying person, which contains the information on accompanying persons.



3.2.4.27 Accompanying group (group registration form)

Object of the data type cBegleitgruppe, which contains the information on accompanying groups.



Important:

The creation (booking) of a group registration form is only possible if this registration form type has been activated for the respective mandate.

When creating a group registration form, it should be noted that the minimum number of people must be observed, otherwise no registration form is possible or an error message will be returned.

3.2.4.28 Accommodation fee / bed tax

The municipalities that have activated the additional module "Accommodation fee / bed tax" can transfer the "Accommodation fee" via the interface.

Day	example
< ue-e-gilt >	< ue-e-gilt > 130.50 < / ue-e-gilt >

It is a numerical field, entry of an "amount / currency", up to two decimal places.

There is a limit to a maximum of 5-digit amounts ("pre-decimal places") in order to "cushion incorrect entries".

3.3 Book operation registration form

Request element	Response element
-----------------	------------------

registration cards	registration cards
--------------------	--------------------

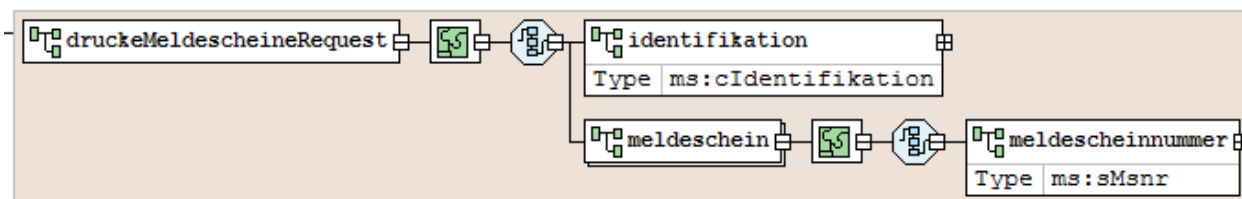
Object registration form

See under the method [get registration form](#) .

3.4 Print notification slip operation

Request element	Response element
Print registration formsRequest	print registration forms response

3.4.1 Print registration formsRequest



Example:

```

< p1: print registration forms request
  xmlns: p1 = "http://www.avs.meldeschein.de/ns/"
  xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < ID >
    < generation > 2016-07-01 < /generation >
    < interface > xxx < /interface >
    < spa administration > 1 < / spa administration >
    < userid > 1 < / userid >
    < sessionid > string < / sessionid >
    < processing > MS PRINTING < /processing >
    < version > string < /version >
  < /ID >
  < ! - element meldeschein, maxOccurs = unbounded-- >
  < registration form >
    < registration number > 100 < / registration number >
  < / registration card >
  < registration form >
    < registration number > 101 < / registration number >
  < / registration card >
< / p1: print registration forms request >

```

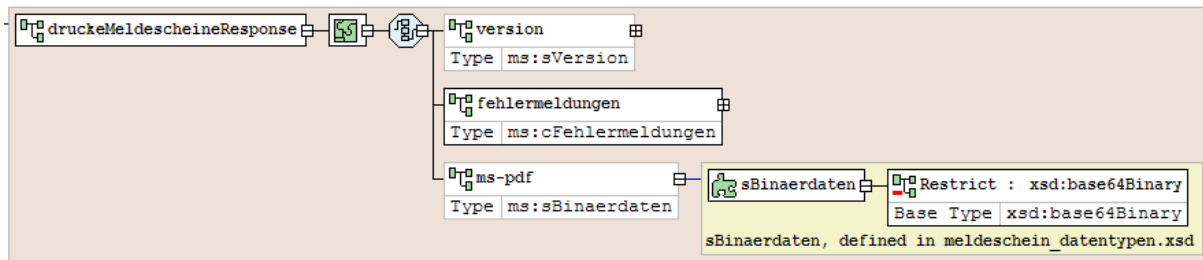
3.4.2 Meaning of the fields of the object "registration certificate"

In this object, the registration form (s) is identified using the registration form number assigned by the AVS registration form.

3.4.2.1 Registration form number

The registration number assigned by the AVS registration certificate.

3.4.3 print registration forms response



Example:

```
< ? xml version = "1.0"? >
< SOAP-ENV: Envelope xmlns: SOAP-
ENV = "http://schemas.xmlsoap.org/soap/envelope/"
xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < SOAP-ENV: Body >
    < print registration forms response xmlns = "http://www.avs.meldeschein.de/ns/" > < version xmlns = "" > 1.2.06-H18
    < /version >
    < error messages xmlns = "" >
      < error >
        < code > 10001 < /code >
        < description > Registration forms successful < /description > < /error >
      < / error messages >
    < ms-pdf xmlns = "" > JVBERi0xLjMKJaqrK0KNCAwIG9iago8PCA ..... < / ms-pdf > < / print registration
    forms response >
  < / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >
```

3.4.4 Meaning of the field of object "ms-pdf"

in the Day `< ms-pdf >` of the xsd: base64Binary type, the serialized form of the PDF is returned in the response.

A Java web service client would receive a byte array for this type, for example, which can simply be saved as a file.

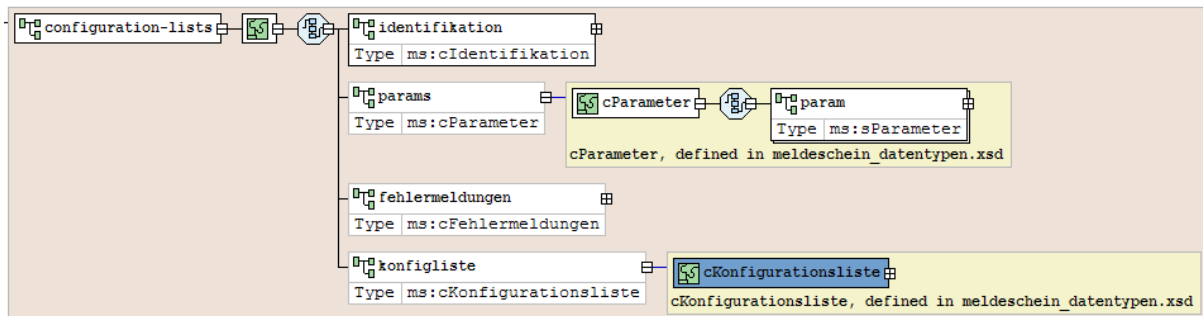
3.5 getKonfigurationsListe method

Request element	Response element
configuration lists	configuration lists

The Day `< processing >` determines which list is requested:

- CONFIG-L ADDRESS Address
- CONFIG-L-LAND countries
- CONFIG-L-CAT Registration form categories

3.5.1 Configuration lists method



The respective configuration list is returned in the response.

Example configuration list "Country" Request:

```
< soapenv : Envelope xmlns: soapenv = "http://schemas.xmlsoap.org/soap/envelope/" xmlns: ns =
"http://www.avs.meldeschein.de/ns/" >
  < soapenv : Header / >
  < soapenv : body >
    < ns : configuration - lists >
      < ID >
        < generation > 2016-07-01 < /generation >
        < interface > xxx < /interface >
        < spa administration > 1 < / spa administration >
        < userid > 1 < / userid >
        < sessionid > string < / sessionid >
        < processing > CONFIG-L-LAND < /processing >
      < /ID >
    < / ns : configuration - lists >
  < / soapenv : body >
< / soapenv : Envelope >
```

Example response (shortened version):

```
< S: Envelope xmlns: S = "http://schemas.xmlsoap.org/soap/envelope/" > < S: :Body >

  < ns2: configuration-lists xmlns: ns2 = "http://www.avs.meldeschein.de/ns/" > < ID >

    < generation > 2016-07-01 < /generation >
    < interface > xxx < /interface >
    < spa administration > 1 < / spa administration >
    < userid > 1 < / userid >
    < sessionid > string < / sessionid >
    < processing > CONFIG-L-LAND < /processing >
```

```

</ID>
  < error messages >
    < error >
      < code > 10001 </code>
      < description > ConfigurationLists query successful </ description> </error>

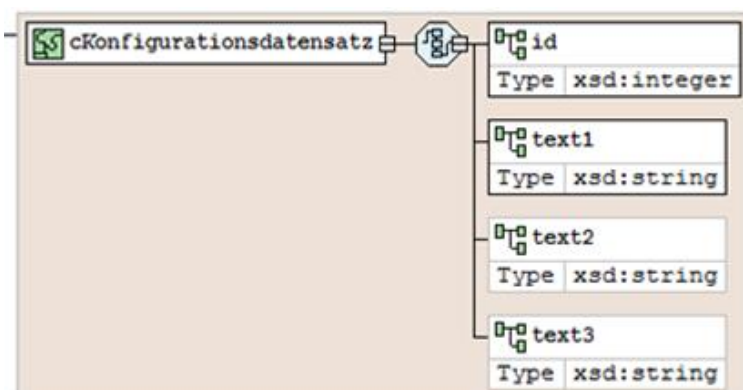
    </ error messages >
  < config list >
    ...
    < config data set >
      < id > 1 </ id>
      < text1 > Germany </ text1>
      < text2 > German </ text2>
      < text3 > DE </ text3>
    </ config data set>
    < config data set >
      < id > 40 </ id>
      < text1 > Arab countries of Asia </ text1>
      < text2 > Arabic </ text2>
      < text3 > AE </ text3>
    </ config data set>
    ...
  </ configlist>
</ ns2: configuration lists>
</ S: Body>
</ S: Envelope>

```

3.5.2 Meaning of the fields of the "configuration lists" object

In this general function for fetching various configuration lists, it is determined in the processing identifier of the identification block which configuration list is requested. Further parameterization of the corresponding configuration list can be done via the 'params' objects.

3.5.2.1 Return element config list



Example:

```
< config list >
  < config data set >
    < id > 123 < / id >
    < text1 > string1 < / text1 >
    < ! - Element text2 is optional-- > < text2 > string2
    < / text2 >
  < / config data set >
  < config data set >
    < id > 234 < / id >
    < text1 > string3 < / text1 >
    < ! - Element text2 is optional-- > < text2 > string4
    < / text2 >
  < / config data set >
  < config data set >
    < id > 345 < / id >
    < text1 > string5 < / text1 >
    < ! - Element text2 is optional-- > < text2 > string6
    < / text2 >
  < / config data set >
< / configlist >
```

Each configuration list data record shows the respective ID (salutation ID ...) as well as a descriptive text. For an alternative text or further information on the data set, this is DAY < text2 > Field provided (may be expanded).

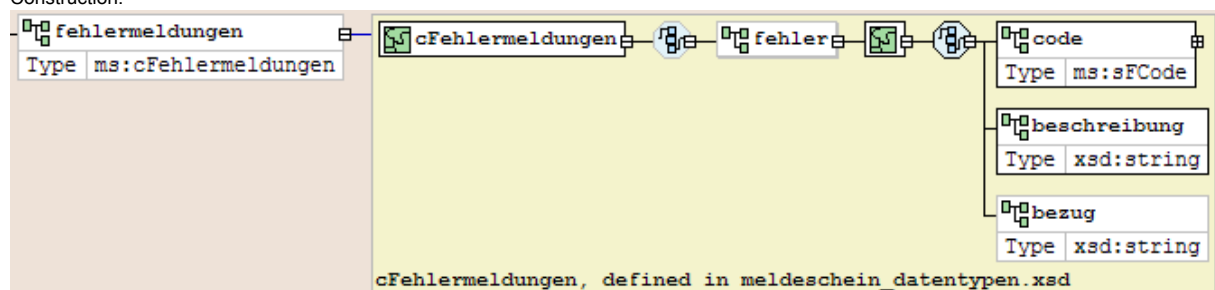
When querying the category list (CONFIG-L-Kat), e.g. DAY < text2 > the category abbreviation entered.

When querying the country list (CONFIG-L-LAND), e.g. DAY < text2 > the Nationality and im DAY < text3 > pass the ISO label.

4 Behavior in the event of an error

In the event of a technical error, the response to the web service client contains an error object of the type "cError messages", which can contain a number of objects of the error type.

Construction:



Example:

```
< error messages >
  < error >
    < code > string < /code >
    < description > string < /description >
    < ! - Element reference is optional- > < reference > string
    < /reference >
  < /error >
  < error >
    < code > string < /code >
    < description > string < /description >
    < reference > string < /reference >
  < /error >
< / error messages >
```

The currently defined errors are listed in the document

"Meldeschein_Webservice_Fehlercodes_.pdf".

In the event of a technical error, a standard SOAP fault is sent as a response.

5 use cases

Use case		Create registration forms								
Brief description		A room booking is created in the booking system. The registration form interface receives the registration form-relevant data from this booking and creates a registration form.								
Actors involved		<table><tr><td><i>Abbreviation</i></td><td><i>designation</i></td></tr><tr><td><i>Ht</i></td><td><i>hotelier</i></td></tr><tr><td><i>WS</i></td><td><i>Registration form interface</i></td></tr><tr><td><i>BS</i></td><td><i>Booking system</i></td></tr></table>	<i>Abbreviation</i>	<i>designation</i>	<i>Ht</i>	<i>hotelier</i>	<i>WS</i>	<i>Registration form interface</i>	<i>BS</i>	<i>Booking system</i>
<i>Abbreviation</i>	<i>designation</i>									
<i>Ht</i>	<i>hotelier</i>									
<i>WS</i>	<i>Registration form interface</i>									
<i>BS</i>	<i>Booking system</i>									
Trigger, preconditions		The booking is recorded and saved								
Result, postconditions		A registration form has been created in the registration form and can be printed out on the client, or the booking system receives an error message.								
		Process, interactions								
number 1		Create registration form								
No.	Actor / system	description								
1.1	Ht	The actor completes his booking in the booking system. The booking								
1.2	BS	system fills an interface object (object registration cards) with the necessary data and transfers the interface object to the WS.								
1.3	WS	<p>The web service validates the interface object validations:</p> <p>The booking number is checked for uniqueness for each object: Error code: 10104</p> <p>When combining company and object, there must be a company-object relationship in the AVS registration form, otherwise the error code: 10101 appears</p> <p>Departure date is greater than or equal to arrival date, otherwise error code: 10100</p> <p>For accompanying persons, the period of stay must be within the period of stay of the main person: Error code: 10102</p> <p>A tariff must be defined for the selected period of stay and the category in order to calculate the tourist tax. Error code: 10103</p>								

		<p>Is a maximum length of stay defined for the category (• Trial guest), this is checked: Error code: 50110</p> <p>The transferred category must be created in the system, otherwise the registration form will not be created. Error code 10121</p> <p>Are additional fields (see Point 3.2.4.24) is defined, if an invalid ID card expires, a</p> <p>Error code supplied: Error code: 10201</p> <p>The registration form and the guest cards can only be printed out when the required minimum number of participants has been reached by the person who has been saved for your spa administration. Error code 10111</p> <p>Notes on the special process for manual registration forms:</p> <p>In the case of manual MS companies, the booking number (corresponds to the manual MS number of the AVS system) is checked for uniqueness • Error codes: 10104 and / or 10114 Check whether manual MS number for the company is maintained in the AVS system, error code 10113</p> <p>If the identifier <kz_manuelle_ms> is used with <i>true</i> transferred (see 3.2.3.2), the company ID and object ID must not be specified. These are determined automatically from the system configuration.</p> <p>Possible error codes: 10101, 10113</p> <p>In the case of manual MS companies, the booking number (corresponds to the manual MS number of the AVS system) is checked for existence • Error codes: 10104 or 10114</p>
1.4	WS	The web service creates the registration form; The interface object (meldescheine object) is provided with the original data, enriched with the registration form number, amount and Billing statusId, returned.
1.5	WS	Exception: The validation of the interface object has failed. The WS returns an error object
1.6	BS	If the registration form has been created successfully, it can now be accessed using the web service function Print registration form to be printed.
		Deviant interactions
No. 2	The WS is not available	

<i>No.</i>	<i>Actor / System</i>	<i>description</i>
2.1	HT	The accommodation provider creates a new booking and saves it.
2.2	BS	The booking system calls the WS. However, this cannot be reached
2.3	BS	The booking system returns an error message that the booking cannot be saved. Alternatively, a Object pool created in which the interface objects for later processing between stored will.

Use case		Change registration forms								
Brief description		An existing room booking is changed in the booking system. The registration form interface receives the registration form-relevant data from this change and updates a registration form.								
Actors involved		<table><tr><td><i>Abbreviation</i></td><td><i>designation</i></td></tr><tr><td>Ht</td><td>hotelier</td></tr><tr><td>WS</td><td>Registration form interface</td></tr><tr><td>BS</td><td>Booking system</td></tr></table>	<i>Abbreviation</i>	<i>designation</i>	Ht	hotelier	WS	Registration form interface	BS	Booking system
<i>Abbreviation</i>	<i>designation</i>									
Ht	hotelier									
WS	Registration form interface									
BS	Booking system									
Trigger, preconditions		An existing room booking is changed and saved.								
Result, postconditions		A registration form has been updated in the registration form or the booking system receives an error message.								
		Process, interactions								
number 1		Create registration form								
No.	Actor / system	description								
1.1	Ht	The actor calls up an existing booking, changes it and saves it.								
1.2	BS	The booking system fetches the booking number + object ID via the interface function MS-HOLEN the current data of the registration form.								
1.3	BS	The interface object is updated by the booking system. The booking system								
1.4	BS	transfers the interface object to the WS function book registration form .								
1.5	WS	The registration form is loaded from the web service using the registration form number. If this is not found in the DB, the error message 10106 is returned.								
1.4	WS	The web service validates the transferred data. The status of the registration form is checked. If the current status does not allow any more changes, the error 10107 is returned. The same validations are then carried out as for the new system. The booking number is an exception. This must not be changed. Accompanying persons / groups are identified via their PersonId (from AVS) identified. Without a PersonId, the accompanying person / group is created. The person ID is supplied after saving or when a registration form is fetched.								

1.5	WS	Exception: The validation of the interface object has failed. The WS returns an error object
		Deviant interactions
No. 2	The WS cannot be reached (see application " Create registration form ").	
No.	Actor / System	description

<i>Use case</i>	Print registration forms	
<i>Brief description</i>	A registration form PDF is generated in the booking system.	
<i>Actors involved</i>		
	<i>Abbreviation</i>	<i>designation</i>
	<i>Ht</i>	<i>hotelier</i>
	<i>WS</i>	<i>Registration form interface</i>
	<i>BS</i>	<i>Booking system</i>
<i>Trigger, preconditions</i>	An existing room booking is printed.	
<i>Result, postconditions</i>	A registration form PDF is generated in the registration form system and the registration form is set to the status "printed".	
	<i>Process, interactions</i>	
number 1	Create registration form	
<i>No.</i>	<i>Actor / system</i>	<i>description</i>
1.1	Ht	The actor calls up an existing booking and prints it. The booking system fetches the
1.2	BS	booking number + object ID via the interface function get registration form the registration number.
1.3	BS	The interface object is printed by the booking system. The booking system
1.4	BS	transfers the registration number to the WS function Print registration form
1.5	WS	The registration form uses the registration form number transferred to set the print status and generates

		<p>Registration form PDF.</p> <p>If it is not found in the DB, error message 10106 is returned.</p> <p>If the registration form can no longer be printed, for example because it has been canceled, error message 10110 is returned. If the minimum number of accompanying persons is not reached for a group registration form, the web service returns the error code 10111.</p>
		Deviant interactions
No. 2	The WS is not available (see Use case "Registration form create")	
No.	Actor / System	description
Use case	Cancel registration forms	
Brief description	An existing room booking is canceled in the booking system.	
Actors involved	Abbreviation	designation
	Ht	hotelier
	WS	Registration form interface
	BS	Booking system
Trigger, preconditions	An existing room booking will be canceled	
Result, postconditions	The registration form has been updated in the registration form or the booking system receives an error message.	
		Process, interactions
number 1	Create registration form	
No.	Actor / system	description
1.1	Ht	The actor calls up an existing booking and cancels it. The booking system fetches
1.2	BS	the booking number + object ID via the interface function get registration form the current data of the registration form.
1.3	BS	The interface object is updated by the booking system: The < billing statusid> will be on 4th set • Cancellation status. The booking system transfers the interface
1.4	BS	object to the WS function book registration form .
1.5	WS	The registration form is loaded from the web service using the registration form number. If it is not found in the DB, error message 10106 is returned

1.4	WS	The web service validates the transferred data. The status of the registration form is checked. If the act. Status no longer allows changing / canceling, the error 10107/10112 is returned.
1.5	WS	Exception: The validation of the interface object has failed. The WS returns an error object.
		<i>Deviant interactions</i>
No. 2	The WS cannot be reached (see application " Create registration form ")	
<i>No.</i>	<i>Actor / System</i>	<i>description</i>

6 SOAP request for the "book registration form" function

This function regulates the creation and modification of registration forms.

Example request new creation:

```
< ? xml version = "1.0"? >
< ms: registration cards xmlns: ms = "http://www.avs.meldeschein.de/ns/"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < ID >
    < generation > 2016-07-01 < /generation >
    < interface > xxx < /interface >
    < spa administration > 1 < / spa administration >
    < userid > 1 < / userid >
    < sessionid > string < / sessionid >
    < processing > BOOK < /processing >
    < version > string < /version >
  < /ID >
  < ms: registration form >
    < booking number > 30501 < /booking number >
    < company id > 222 < / companyid >
    < objectid > 222 < / objectid >
    < getting there > 2016-07-01 < /getting there >
    < departure > 2016-07-02 < /departure >
    < category ID > 2 < / categoryid >
    < address > 1 < / salutation >
    < Surname > Test name < /Surname >
    < First name > Test first name < /First name >
    < road > Test street < /road >
    < House number > 1 < /House number >
    < Postcode > 12345 < /Postcode >
    < place > Test city < /place >
    < landid > 1 < / landid >
    < nationalityid > 1 < / nationalityid >
    < More information > 12345678 < /More information >
    < ID number > 12345678ABC < / ID no >
    < License Plate > BT TB-44 < /License Plate >
    < Date of birth > 1972-01-21 < /Date of birth >
    < Companion >
      < address > 1 < / salutation >
      < Surname > Test-Name-BP < /Surname >
      < First name > Test first name BP < /First name >
      < Date of birth > 1955-03-11 < /Date of birth >
      < category ID > 3 < / categoryid >
      < getting there > 2016-07-01 < /getting there >
      < departure > 2016-07-02 < /departure >
      < ID number > 123478ABC < / ID no >
    < /Companion >
  < /ms: registration form >
< / ms: registration cards >
```

```

    < Companion >
      < address > 2 < / salutation >
      < Surname > string < /Surname >
      < First name > string < /First name >
      < Date of birth > 1999-01-21 < /Date of birth >
      < category ID > 2 < / categoryid >
      < ID number > 45678ABC < / ID no >
    < /Companion >
  < / ms: registration form >
< / ms: registration cards >

```

The Day < processing > - must have the value **BOOK** have in this function.

The Id's for salutation, company etc. can then be found in the corresponding configuration lists. At the Day < nationalityid > the Id from the country config list is used.

Example response:

Important information is then attached to the response (green highlighted).

```

< ? xml version = "1.0"? >
< SOAP-ENV: Envelope
  xmlns: SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < SOAP-ENV: Body >
    < registration cards xmlns = "http://www.avs.meldeschein.de/ns/" > < ID xmlns
      = "" >
      < generation > 2016-07-01 < /generation >
      < interface > xxx < /interface >
      < spa administration > 1 < / spa administration >
      < userid > 1 < / userid >
      < sessionid > string < / sessionid >
      < processing > BOOK < /processing >
      < version > 1.2.06-H03 < /version >
    < /ID >
    < error messages xmlns = "" >
      < error >
        < code > 10001 < /code >
        < description > Registration slip booking successful < /description > < /error >
      < / error messages >
    < registration form >
      < booking number xmlns = "" > 30501 < /booking number >
      < registration number xmlns = "" > 216454 < /meldescheinnummer >
      < company id xmlns = "" > 222 < / companyid >
      < objectid xmlns = "" > 222 < / objectid >
      < getting there xmlns = "" > 2016-07-01 < /getting there >
      < departure xmlns = "" > 2016-07-02 < /departure >
      < category ID xmlns = "" > 2 < / categoryid >
      < address xmlns = "" > 1 < / salutation >
      < Surname xmlns = "" > Test name < /Surname >

```



```

< First name xmlns = "" > Test first name < /First name >
< road xmlns = "" > Test street < /road >
< House number xmlns = "" > 1 < /House number >
< Postcode xmlns = "" > 12345 < /Postcode >
< place xmlns = "" > Test city < /place >
< landid xmlns = "" > 1 < / landid >
< nationalityid xmlns = "" > 1 < / nationalityid > < More information xmlns = "" > 12345678
< /More information >
< ID number xmlns = "" > 12345678ABC < / ID no >
< License Plate xmlns = "" > BT TB-44 < /License Plate >
< Date of birth xmlns = "" > 1972-01-21 < /Date of birth >
< amount xmlns = "" > 10 < / amount >
< billing statusid xmlns = "" > 1 < /billing statusid >
< Companion xmlns = "" >
  < personid > 529955 < /personid >
  < address > 1 < / salutation >
  < Surname > Test-Name-BP < /Surname >
  < First name > Test first name BP < /First name >
  < Date of birth > 1955-03-11 < /Date of birth >
  < category ID > 3 < / categoryid >
  < ID number > 123478ABC < / ID no >
  < getting there > 2016-07-01 < /getting there >
  < departure > 2016-07-02 < /departure >
< /Companion >
< Companion xmlns = "" >
  < personid > 529956 < /personid >
  < address > 2 < / salutation >
  < Surname > string < /Surname >
  < First name > string < /First name >
  < Date of birth > 1999-01-21 < /Date of birth >
  < category ID > 2 < / categoryid >
  < ID number > 45678ABC < / ID no >
< /Companion >
< / registration card >
< / registration cards >
< / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >

```

The **DAY** < registration number > is for that **Change / update** of the registration form is absolutely necessary, otherwise a new registration form is always created! If the same booking number is booked twice for each property, the error code will appear the second time **10104** (" Registration form with the booking number already exists! ") Delivered back.

In detail:

New registration form:

...
< booking number > 1A2B3C4D5E < /booking number >
...

Change / update registration form: _____

...
< registration number > 1A2B3C4D5E < / registration number >
...

or

...
< booking number > 1A2B3C4D5E < /booking number >
< registration number > 1A2B3C4D5E < / registration number >
...

The DAY < personid > (from AVS) is absolutely necessary for the update to clearly identify the accompanying person (s)!

Will not be a DAY < personid > handed over, a new accompanying person is created / added!

7 SOAP request "getMeldeschein"

This function is used to display the current registration form data using the

- Booking number and object
- Registration number

for display or to pick up later updates.

The function "getMeldeschein" is used to query the data of the created registration form from the AVSMeldeschein application. If a change has to be made to the registration form after it has been created, the data can be fetched or displayed using this function and changed using the "book registration form" function.

With the "MS-Get" method, different combinations of DAY `s possible (see below).

Will the DAY < registration number > and the DAY < booking number > hand over together, then it is DAY < registration number > leading. The DAY < booking number > and the DAY < objectid > are only read out in this case if the DAY < registration number > is passed empty.

The DAY < booking number > must always be combined with the DAY < objectid > as these two values together clearly identify a registration form. (Provided the registration form was created with a booking number)

The following combinations of DAY are possible:

booking number	+	Object ID	
Registration number	+	Object ID	
Registration number	+	booking number	
Registration number	+	booking number	+ Object ID
Registration number			

7.1 Request for registration form with booking number

```
< SOAP-ENV: Envelope
  xmlns: SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < SOAP-ENV: Body >
    < get registration form xmlns = "http://www.av.s.meldeschein.de/ns/" > < ID xmlns = ""
    >
      < generation > 2016-07-01 < /generation >
      < interface > xxx < /interface >
      < spa administration > 1 < / spa administration >
      < userid > 1 < / userid >
      < sessionid > string < / sessionid >
      < processing > MS-HOLEN < /processing >
      < version > 1.2.06-H03 < /version >
    < /ID >
    < request data xmlns = "" >
      < booking number > 30001 < /booking number >
      < objectid > 222 < / objectid >
    < / request data >
  < / get registration card >
< / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >
```

7.2 Request get registration form with registration form number

```
< SOAP-ENV: Envelope
  xmlns: SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < SOAP-ENV: Body >
    < get registration form xmlns = "http://www.av.s.meldeschein.de/ns/" > < ID xmlns = ""
    >
      < generation > 2016-07-01 < /generation >
```

```

    < interface > xxx < /interface >
    < spa administration > 1 < / spa administration >
    < userid > 1 < / userid >
    < sessionid > string < / sessionid >
    < processing > MS-HOLEN < /processing >
    < version > 1.2.06-H03 < /version >
  < /ID >
  < request data xmlns = "" >
    < registration number > 30001 < / registration number >
  < / request data >
< / get registration card >
< / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >

```

7.3 SOAP response "getMeldeschein"

```

< SOAP-ENV: Envelope
  xmlns: SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < SOAP-ENV: Body >
    < get registration form xmlns = "http://www.avs.meldeschein.de/ns/" > < ID xmlns = ""
      >
        < generation > 2016-07-01 < /generation >
        < interface > xxx < /interface >
        < spa administration > 1 < / spa administration >
        < userid > 1 < / userid >
        < sessionid > string < / sessionid >
        < processing > MS-HOLEN < /processing >
        < version > 1.2.06-H03 < /version >
      < /ID >
      < request data xmlns = "" >
        < booking number > 30001 < /booking number >
        < objectid > 222 < / objectid >
      < / request data >
      < error messages xmlns = "" >
        < error >
          < code > 10001 < /code >
          < description > Fetching registration forms successful < /description > < /error >
        < / error messages >
      < registration form >
        < booking number xmlns = "" > 30001 < /booking number >
        < registration number xmlns = "" > 123456 < / registration number >
        < company id xmlns = "" > 111 < / companyid >
        < objectid xmlns = "" > 222 < / objectid >
        < getting there xmlns = "" > 2016-07-01 < /getting there >
      < / registration form >
    < / get registration form >
  < / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >

```

```

< departure xmlns = "" > 2016-07-02 < /departure >
< category ID xmlns = "" > 2 < / categoryid >
< address xmlns = "" > 1 < / salutation >
< Surname xmlns = "" > Test name < /Surname >
< First name xmlns = "" > Test first name < /First name >
< road xmlns = "" > Test street < /road >
< House number xmlns = "" > 1 < /House number >
< Postcode xmlns = "" > 12345 < /Postcode >
< place xmlns = "" > Test city < /place >
< landid xmlns = "" > 1 < / landid >
< nationalityid xmlns = "" > 1 < / nationalityid > < More information xmlns = "" > 12345678
< /More information >
< ID number xmlns = "" > 12345678ABC < / ID no >
< License Plate xmlns = "" > BT TB-44 < /License Plate >
< Date of birth xmlns = "" > 1972-01-21 < /Date of birth >
< billing status id xmlns = "" > 1 < / billing statusid >
< Companion xmlns = "" >
  < personid > 1234 < / personid >
  < address > 1 < / salutation >
  < Surname > Test-Name-BP < /Surname >
  < First name > Test first name BP < /First name >
  < Date of birth > 1955-03-11 < /Date of birth >
  < category ID > 3 < / categoryid >
  < getting there > 2016-07-01 < /getting there >
  < departure > 2016-07-02 < /departure >
  < ID number > 123478ABC < / ID no >
  < /Companion >
  < Companion xmlns = "" >
    < personid > 2345 < / personid >
    < address > 2 < / salutation >
    < Surname > string < /Surname >
    < First name > string < /First name >
    < Date of birth > 1999-01-21 < /Date of birth >
    < category ID > 2 < / categoryid >
    < getting there > 2016-07-01 < /getting there >
    < departure > 2016-07-02 < /departure >
    < ID number > 45678ABC < / ID no >
    < /Companion >
  < / registration form >
< / get registration card >
< / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >

```

8 SOAP request "getKonfigurationsListe"

For the category query, the SOAP response would look like this, for example:

```
< ? xml version = "1.0"? >
< SOAP-ENV: Envelope xmlns: SOAP-
ENV = "http://schemas.xmlsoap.org/soap/envelope/"
xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < SOAP-ENV: Body >
    < configuration lists xmlns = "http://www.avs.meldeschein.de/ns/" > < ID xmlns = "" >

      < generation > 2016-07-01 < /generation >
      < interface > xxx < /interface >
      < spa administration > 1 < / spa administration >
      < userid > 1 < / userid >
      < sessionid > string < / sessionid >
      < processing > CONFIG-L-CAT < /processing >
      < version > 1.2.06-H03 < /version >
    < /ID >
    < error messages xmlns = "" >
      < error >
        < code > 10001 < /code >
        < description > ConfigurationLists query successful < /description > < /error >

      < / error messages >
      < config list xmlns = "" >
        < config data set >
          < id > 1 < / id >
          < text1 > Adult < / text1 >
          < text2 > E. < / text2 >
        < / config data set >
        < config data set >
          < id > 5 < / id >
          < text1 > Adult companion < / text1 >
          < text2 > EB < / text2 >
        < / config data set >
        < config data set >
          < id > 2 < / id >
          < text1 > Adult 100% < / text1 >
          < text2 > EE < / text2 >
        < / config data set >
        < config data set >
          < id > 3 < / id >
          < text1 > child < / text1 >
          < text2 > K < / text2 >
        < / config data set >
        < config data set >
          < id > 4th < / id >
```

```
< text1 > Child reduced < / text1 >
< text2 > KE < / text2 >
< / config data set >
< config data set >
  < id > 9 < / id >
  < text1 > no category < / text1 >
  < text2 > 0 < / text2 >
< / config data set >
< / configlist >
< / configuration-lists >
< / SOAP-ENV: Body >
< / SOAP-ENV: Envelope >
```

9 examples

9.1 Request book registration form Group registration form

Example request to create a new group registration form:

```
< ? xml version = "1.0"? >
< ms: registration cards xmlns: ms = "http://www.avs.meldeschein.de/ns/"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" >
  < ID >
    < generation > 2016-07-01 < /generation >
    < interface > xxx < /interface >
    < spa administration > 1 < / spa administration >
    < userid > 1 < / userid >
    < sessionid > string < / sessionid >
    < processing > BOOK < /processing >
    < version > string < /version >
  < /ID >
  < ms: registration form >
    < booking number > 30501 < /booking number >
    < company id > 222 < / companyid >
    < objectid > 222 < / objectid >
    < getting there > 2016-07-01 < /getting there >
    < departure > 2016-07-02 < /departure >
    < category ID > 2 < / categoryid >
    < address > 1 < / salutation >
    < Surname > Test name < /Surname >
    < First name > Test first name < /First name >
    < road > Test street < /road >
    < House number > 1 < /House number >
    < Postcode > 12345 < /Postcode >
    < place > Test city < /place >
    < landid > 1 < / landid >
    < nationalityid > 1 < / nationalityid >
    < More information > 12345678 < /More information >
    < ID number > 12345678ABC < / ID no >
    < License Plate > BT TB-44 < /License Plate >
    < Date of birth > 1972-01-21 < /Date of birth >
    < escort group >
      < category ID > 1 < / categoryid >
      < number > 4th < /number >
      < ID number > 12345678ABCD < / ID no >
    < / escort group >
  < / ms: registration form >
< / ms: registration cards >
```


9.2 Response Incorrect login

If incorrect user data (user name, password) is transferred when calling up the registration form web service, the user receives the HTTP status code

401. With this so-called HTTP basic authentication, the user data are transmitted to the server in base64 encoded form and authorization is only carried out in a further step on the basis of the data contained in the identification block.

Example response incorrect login:

Error 401--Unauthorized

From RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1:

10.4.2 401 Unauthorized

The request requires user authentication. The response MUST include a WWW-Authenticate header field (section 14.46) containing a challenge applicable to the requested resource. The client MAY repeat the request with a suitable Authorization header field (section 14.8). If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user SHOULD be presented the entity that was given in the response, since that entity MAY include relevant diagnostic information. HTTP access authentication is explained in section 11.

10 support requests

Should unforeseen errors occur in the web service, please do not hesitate to contact AVS GmbH. The request should always contain the exact date and time and the request and the corresponding response should also be sent with the error message. A request can only be processed to a limited extent without these details and the answer then takes longer, depending on the research effort.

11 Related sources:

Filename	comment
meldeschein_buchen.xsd	Definitions of the interface objects for registration form management
meldeschein_datotypen.xsd	Basic data types

IJMeldescheinWebservices.wsdl	WSDL definition
Registration form web service error codes 6.3	Directory of all technical error codes that the web service can deliver.