



dpd



DPD Shipper

Manual – DPD Shipper WebServices

Versie 1.0

Table of contents

1. LoginService	3
1.1 Methods.....	3
1.2 Elements.....	6
2. ParcelLifeCycleService	8
2.1 History	8
2.2 Methods.....	8
2.3 Complex Types.....	15
2.4 Elements.....	20
3. ParcelShopFinderService	22
3.1 Methods.....	22
3.2 Complex Types.....	31
3.3 Namespace: types_authentication	41
3.4 Simple Types.....	42
3.5 Elements.....	42
4. ShipmentService	45
4.1 History	45
4.2 Methods.....	45
4.3 Complex Types.....	47
4.4 Simple Types.....	67
4.5 Elements.....	68

1. LoginService

1.1 Methods

Following methods are available for the service LoginService-Public_2_0

Method name	Description
getAuth	Creates an authentication token for the committed user if user name and password are valid. The authentication token is needed for accessing other DPD Web Services.

1.1.1 Method: getAuth

Description

Creates an authentication token for the committed user if user name and password are valid.
The authentication token is needed for accessing other DPD Web Services.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://dpd.com/common/service/types/LoginService/2.0">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getAuth>
      <delisId><your delisId></delisId>
      <password><your password></password>
      <messageLanguage>de_DE</messageLanguage>
    </ns:getAuth>
  </soapenv:Body>
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/LoginService/V2_0/getAuth

JSON-Data for POST-Request (for GET-Request data must send with parameter "request" and for JSONP the Parameter "jsonpcallback" can be used, to send a padding function name):

```
{
  "delisId" : "<your delisId>",
  "password": "<your password>",
  "messageLanguage": "en_EN"
}
```

REST-Result, without error:

```
{
  "getAuthResponse": {"return": {
    "delisId": "<your delisId>",
    "customerUid": "<your customerUid>",
    "authToken": "<your authentication token>",
    "depot": "<your customerUid>"
  }},
  "status": {
    "type": "OK",
    "code": "200",

    "message": "valid"
  }
}
```

```
}
}
```

REST-Result, if an error occurred:

```
{"status": {
  "type": "AuthenticationFault",
  "code": "<Error code, e.g. LOGIN_8>",
  "message": "<Error message, e.g. The combination of user and password is invalid.>"
}}
```

Additional REST-State on system error:

Error-Code: 100

Error-Type: SystemFault

Error-Message: <Message, which describe the error>

Action

<http://dpd.com/common/service/LoginService/2.0/getAuth>

Input

The input for method getAuth is as follows:

Name	Element
parameters	types:getAuth

Output

The output for method getAuth is as follows:

Name	Element
parameters	types:getAuthResponse

Faults

The defined faults for method getAuth are as follows:

Name	Element
AuthenticationFault	types_authentication:authenticationFault

1.1.2 Namespace: types

URL: <http://dpd.com/common/service/types/LoginService/2.0>

Complex Type: Login

Description

Is created when a user logs in and contains its login information.

Field name	Data type / Occurrence / Range	Description
delisId	xsd:string / 1 .. 1 /	The user's DELIS-Id.
customerUid	xsd:string / 1 .. 1 /	The user's customer uid. This is needed for subaccounts, usually this is equal to DELIS-Id
authToken	xsd:string / 1 .. 1 /	The Authtoken, needed for other web service calls.
depot	xsd:string / 1 .. 1 /	The depot, to which the user is assigned.

1.1.2.1 Complex Type: LoginException

Field name	Data type / Occurrence / Range	Description
additionalData	xsd:string / 0 .. 1 /	Additional data for an error.
additionalInfo	xsd:string / 0 .. 1 /	Additional info for an error in user language.
errorClass	xsd:string / 0 .. 1 /	The class which creates the error.
errorCode	xsd:string / 0 .. 1 /	The error code.
fullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in user language.
language	xsd:string / 0 .. 1 /	Language for messages.
message	xsd:string / 0 .. 1 /	The complete message text for the error in user language.
shortMessage	xsd:string / 0 .. 1 /	The short info text for the error in user language.
systemFullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in system language.
systemMessage	xsd:string / 0 .. 1 /	The complete message text for the error in system language.
systemShortMessage	xsd:string / 0 .. 1 /	The short info text for the error in system language.

1.1.2.2 Complex Type: getAuth

Field name	Data type / Occurrence / Range	Description
delisId	xsd:string / 1 .. 1 /	The DELIS-Id of the user.
password	xsd:string / 1 .. 1 /	The password of the user.
messageLanguage	xsd:string / 1 .. 1 / minLength: 5 maxLength: 5	The language (Java format) for messages. "de_DE" for german messages. "en_US" for english messages.

1.1.2.3 Complex Type: getAuthResponse

Field name	Data type / Occurrence / Range	Description
return	types:Login / 1 .. 1 /	

1.1.3 Namespace: types_authentication

URL: <http://dpd.com/common/service/types/Authentication/2.0>

1.1.3.1 Anonymous Complex Type of Element: authentication

Description

The data structure for authentication data.

Field name	Data type / Occurrence / Range	Description
delisId	xsd:string / 1 .. 1 / minLength: 8 maxLength: 10	The delis user id for authentication.
authToken	xsd:string / 1 .. 1 / minLength: 0 maxLength: 64	The token for authentication. Field authToken of Login, as a result of Method "getAuth" of LoginService.
messageLanguage	xsd:string / 1 .. 1 / minLength: 5 maxLength: 5	The language (Java format) for messages. "de_DE" for german messages. "en_US" for english messages.

1.1.3.2 Anonymous Complex Type of Element: authenticationFault

Description

The data structure for authentication errors

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 /	The error code. Valid codes: NoAccessAuthenticationFault=-1 NoRightsAuthenticationFault=-2
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	The error message text.

1.2 Elements

1.2.1 Namespace: types

URL: <http://dpd.com/common/service/types/LoginService/2.0>

1.2.1.1 Element: LoginException

Field name	Data type / Occurrence / Range	Description
LoginException	types:LoginException / 1 .. 1 /	Thrown if authentication fails.

1.2.1.2 Element: *getAuth*

Field name	Data type / Occurrence / Range	Description
getAuth	types:getAuth / 1 .. 1 /	Input parameter of getAuth.

1.2.1.3 Namespace: *types_authentication*

URL: <http://dpd.com/common/service/types/Authentication/2.0>

Field name	Data type / Occurrence / Range	Description
getAuthResponse	types:getAuthResponse / 1 .. 1 /	Output parameter of getAuth.

1.2.1.4 Element: *authentication*

Anonymous Complex Type *types_authentication:authentication*

1.2.1.5 Element: *authenticationFault*

Anonymous Complex Type *types_authentication:authenticationFault*

2. ParcelLifeCycleService

2.1 History

Version 2.0.0

- Changed parameter parcellabelnumber to parcelLabelNumber for method getTrackingData
- New method getParcelLabelNumberForWebNumber added

Methods

Following methods are available for the service ParcelLifeCycleService-Public_2_0

Method name	Description
getParcelLabelNumberForWebNumber	Provides the parcel label number for the given web number.
getTrackingData	Deliver parcel tracking data for a parcel label number. An exception is thrown, if an error has occurred.

2.1.1 Method: getParcelLabelNumberForWebNumber

Description

Provides the parcel label number for the given web number.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://dpd.com/common/service/types/ParcelLifeCycleService/2.0">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getParcelLabelNumberForWebNumber>
      <delisId>delis123</delisId>
      <webNumber>IO1234567</webNumber>
      <messageLanguage>en_EN</messageLanguage>
    </ns:getParcelLabelNumberForWebNumber>
  </soapenv:Body>
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/ParcelLifeCycleService/V2_0/getParcelLabelNumberForWebNumber
JSON data for POST request (for GET request data has to be sent with parameter "request" and for JSONP with parameter "jsonpcallback" to obtain a padding function name):

```
{ "authentication" : {
  "delisId" : "<your delisId>",
  "authToken" : "<your authToken>",
  "messageLanguage" : "en_EN"
},
"getParcelLabelNumberForWebNumber" : {
  "webNumber" : "IO1234567"
}
}
```

REST result without error:

```
{
```



```
"getParcelLabelNumberForWebNumberResponse": {"parcelLabelNumber": "01234567890123"},
"status": {
  "type": "OK",
  "code": "200",

  "message": "valid"
}
}
```

REST result if an error occurred:

```
{"status": {
  "type": "SystemFault",
  "code": "<Error code, e.g. 100>",
  "message": "<Error message>"
}}
```

Additional REST state on system error:

```
Error-Code: 100
Error-Type: SystemFault
Error-Message: <Message describing the error>
```

Action

<http://dpd.com/common/service/ParcelLifeCycleService/2.0/getParcelLabelNumberForWebNumber>

Input

The input for method `getParcelLabelNumberForWebNumber` is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:getParcelLabelNumberForWebNumber

* header parameter

Output

The output for method `getParcelLabelNumberForWebNumber` is as follows:

Name	Element
parameters	types:getParcelLabelNumberForWebNumberResponse

Faults

The defined faults for method `getParcelLabelNumberForWebNumber` are as follows:

Name	Element
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

2.1.2 Method: getTrackingData

Description

Deliver parcel tracking data for a parcel label number.

An exception is thrown, if an error has occurred.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0"
xmlns:ns1="http://dpd.com/common/service/types/ParcelLifecycleService/2.0">
  <soapenv:Header>
    <ns:authentication>
      <delisId>your delisId</delisId>
      <authToken>your authToken</authToken>
      <messageLanguage>en_EN</messageLanguage>
    </ns:authentication>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:getTrackingData>
      <parcelLabelNumber>01405400945058</parcelLabelNumber>
    </ns1:getTrackingData>
  </soapenv:Body>
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/ParcelLifecycleService/V2_0/getTrackingData

JSON data for POST request (for GET request data has to be sent with parameter "request" and for JSONP with parameter "jsonpcallback" to obtain a padding function name):

```
{ "authentication" : {
  "delisId" : "<your delisId>",
  "authToken" : "<your authToken>",
  "messageLanguage" : "en_EN"
},
  "getTrackingData" : {
    "parcelLabelNumber" : "01405400945058"
  }
}
```

REST result without error:

```
{
  "getTrackingDataResponse": {"trackingresult": {
    "shipmentInfo": {
      "receiver": null,
      "predictInformation": null,
      "serviceDescription": {
        "label": {
          "content": "Your DPD service: ",
          "bold": false,
          "paragraph": false
        },
        "content": [
          {
            "content": "DPD CLASSIC",

```

```

        "bold": false,
        "paragraph": false
    }},
    "linkTarget": ""
},
"additionalServiceElements": null,
"status": "SHIPMENT",
"label": {
    "content": "Shipment information",
    "bold": true,
    "paragraph": false
},
"description": {
    "label": null,
    "content": [
        {
            "content": "Details of your shipment",
            "bold": false,
            "paragraph": false
        }
    ]
},
"linkTarget": ""
},
"statusHasBeenReached": false,
"isCurrentStatus": false,
"showContactInfo": false,
"location": null,
"date": null,
"normalItems": null,
"importantItems": null,
"errorItems": null
},
"statusInfo":
{
    "status": "ACCEPTED",
    "label": {
        "content": "Parcel handed to DPD",
        "bold": true,
        "paragraph": false
    },
    "description": {
        "label": null,
        "content": [
            {
                "content": "DPD has received your parcel.",
                "bold": false,
                "paragraph": false
            }
        ]
    },
    "linkTarget": ""
},
"statusHasBeenReached": true,
"isCurrentStatus": false,
"showContactInfo": false,
"location": null,
"date": null,
"normalItems": null,
"importantItems": null,
"errorItems": null
},
{
    "status": "AT_SENDING_DEPOT",
    "label": {

```

```

"content": "At dispatch depot",
"bold": true,
"paragraph": false
},
"description": {
  "label": null,
  "content": [
    {
      "content": "The parcel is at the dispatch depot.",
      "bold": false,
      "paragraph": false
    }
  ],
  "linkTarget": ""
},
"statusHasBeenReached": true,
"isCurrentStatus": false,
"showContactInfo": false,
"location": {
  "content": "Duisburg (DE)",

  "bold": true,
  "paragraph": false
},
"date": {
  "content": "16/08/2012, 17:00 ",
  "bold": true,
  "paragraph": false
},
"normalItems": null,
"importantItems": null,
"errorItems": null
},
{
  "status": "ON_THE_ROAD",
  "label": {
    "content": "In transit",
    "bold": true,
    "paragraph": false
  },
  "description": {
    "label": null,
    "content": [
      {
        "content": "Your parcel is on its way to the recipient depot.",
        "bold": false,
        "paragraph": false
      }
    ],
    "linkTarget": ""
  },
  "statusHasBeenReached": true,
  "isCurrentStatus": false,
  "showContactInfo": false,
  "location": null,
  "date": null,
  "normalItems": null,
  "importantItems": null,
  "errorItems": null
},
{
  "status": "AT_DELIVERY_DEPOT",
  "label": {

```

```

"content": "At recipient depot",
"bold": true,
"paragraph": false
},
"description": {
  "label": null,

  "content": [
    {
      "content": "The parcel is at the recipient depot.",
      "bold": false,
      "paragraph": false
    }
  ],
  "linkTarget": ""
},
"statusHasBeenReached": true,
"isCurrentStatus": true,
"showContactInfo": false,
"location": {
  "content": "Gudensberg (DE)",
  "bold": true,
  "paragraph": false
},
"date": {
  "content": "17/08/2012, 07:08 ",
  "bold": true,
  "paragraph": false
},
"normalItems": null,
"importantItems": null,
"errorItems": null
},
{
  "status": "DELIVERED",
  "label": {
    "content": "Parcel delivered",
    "bold": true,
    "paragraph": false
  },
  "description": {
    "label": null,
    "content": [
      {
        "content": "Your parcel has been delivered successfully.",
        "bold": false,
        "paragraph": false
      }
    ],
    "linkTarget": ""
  },
  "statusHasBeenReached": false,
  "isCurrentStatus": false,
  "showContactInfo": false,
  "location": null,
  "date": null,
  "normalItems": null,

  "importantItems": null,
  "errorItems": null
}
],
"contactInfo": null

```

```

}},
"status": {
  "type": "OK",
  "code": "200",
  "message": "valid"
}
}

```

REST result if an error occurred:

```

{"status": {
  "type": "SystemFault",
  "code": "<Error code, e.g. 100>",
  "message": "<Error message>"
}}

```

Additional REST state on system error:

```

Error-Code: 100
Error-Type: SystemFault
Error-Message: <Message describing the error>

```

Action

<http://dpd.com/common/service/ParcelLifeCycleService/2.0/getTrackingData>

Input

The input for method getTrackingData is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:getTrackingData

* header parameter

Output

The output for method getTrackingData is as follows:

Name	Element
parameters	types:getTrackingDataResponse

Faults

The defined faults for method getTrackingData are as follows:

Name	Element
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

2.2 Complex Types

2.2.1 Namespace: types

URL: <http://dpd.com/common/service/types/ParcelLifeCycleService/2.0>

2.2.1.1 Complex Type: ContentItem

Description

Field name	Data type / Occurrence / Range	Description
label	types:ContentLine / 0 .. 1 /	Identifier for content item.
content	types:ContentLine / 1 .. * /	Content of content item.
linkTarget	xsd:string / 0 .. 1 /	If this field is filled, a link will be created from the content item.

2.2.1.2 Complex Type: ContentLine

Description

Contains one text line and information about how the text has to be displayed.

Field name	Data type / Occurrence / Range	Description
content	xsd:string / 1 .. 1 /	Text element
bold	xsd:boolean / 1 .. 1 /	Indicates if text has to be printed bold.
paragraph	xsd:boolean / 1 .. 1 /	Indicates if there has to be a line break after text element.

2.2.1.3 Complex Type: DataFault

Description

Is thrown if a data fault occurs.

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 / minLength: 1 maxLength: 4	Error code.
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	Error message text.

2.2.1.4 Complex Type: Fault

Description

The data structure for error data.

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 / minLength: 1 maxLength: 4	Error code.
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	Error message text.

2.2.1.5 Complex Type: ShipmentInfo

Description

Contains general shipment information (e.g. service code).

Field name	Data type / Occurrence / Range	Description
receiver	types:ContentItem / 0 .. 1 /	Receiver of parcel.
predictInformation	types:ContentItem / 0 .. 1 /	Information about estimated delivery time of parcel (DPD Predict).
serviceDescription	types:ContentItem / 0 .. 1 /	Name of DPD product of the parcel (e.g. DPD Classic).
additionalService Elements	types:ContentItem / 0 .. 1 /	Additional services for DPD product.
trackingProperty	types:TrackingProperty / 0 .. * /	Properties.
status	xsd:string / 1 .. 1 / SHIPMENT ACCEPTED AT_SENDING_DEPOT ON_THE_ROAD AT_DELIVERY_DEPOT DELIVERED	Status
label	types:ContentLine / 1 .. 1 /	Name of status.
description	types:ContentItem / 1 .. 1 /	Description of status.
statusHasBeenReached	xsd:boolean / 1 .. 1 /	Indicates if status has been reached.
isCurrentStatus	xsd:boolean / 1 .. 1 /	Indicates if status is current status.
showContactInfo	xsd:boolean / 1 .. 1 /	Defines if contact information shall be displayed on detail page of the status.

location	types:ContentLine / 0 .. 1 /	Contains the location of the last scan.
date	types:ContentLine / 0 .. 1 /	Contains date and time of the last scan.
normalItems	types:ContentItem / 0 .. * /	A list of content items which describe events classified as NORMAL.
importantItems	types:ContentItem / 0 .. * /	A list of content items which describe events classified as IMPORTANT.
errorItems	types:ContentItem / 0 .. * /	A list of content items which describe events classified as ERROR.

2.2.1.6 Complex Type: StatusInfo

Description

Contains all information and scan data for a status.

Field name	Data type / Occurrence / Range	Description
status	xsd:string / 1 .. 1 / SHIPMENT ACCEPTED AT_SENDING_DEPOT ON_THE_ROAD AT_DELIVERY_DEPOT DELIVERED	Status
label	types:ContentLine / 1 .. 1 /	Name of status.
description	types:ContentItem / 1 .. 1 /	Description of status.
statusHasBeenReached	xsd:boolean / 1 .. 1 /	Indicates if status has been reached.
isCurrentStatus	xsd:boolean / 1 .. 1 /	Indicates if status is current status.
showContactInfo	xsd:boolean / 1 .. 1 /	Defines if contact information shall be displayed on detail page of the status.
location	types:ContentLine / 0 .. 1 /	Contains the location of the last scan.
date	types:ContentLine / 0 .. 1 /	Contains date and time of the last scan.
normalItems	types:ContentItem / 0 .. * /	A list of content items which describe events classified as NORMAL.
importantItems	types:ContentItem / 0 .. * /	A list of content items which describe events classified as IMPORTANT.
errorItems	types:ContentItem / 0 .. * /	A list of content items which describe events classified as ERROR.

2.2.1.7 Complex Type: SystemFault

Description

Is thrown if an internal system fault occurs.

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 / minLength: 1 maxLength: 4	Error code.
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	Error message text.

2.2.1.8 Complex Type: TrackingProperty

Field name	Data type / Occurrence / Range	Description
key	xsd:string / 1 .. 1 /	Property key. Possible keys: - showFollowMyParcelLink (value: true)
value	xsd:string / 1 .. 1 /	Property value.

2.2.1.9 Complex Type: TrackingResult

Description

Result type for tracking data.

Field name	Data type / Occurrence / Range	Description
shipmentInfo	types:ShipmentInfo / 0 .. 1 /	Information about the shipment.
statusInfo	types:StatusInfo / 0 .. * /	Status information.
contactInfo	types:ContentItem / 0 .. 3 /	Contact information.

2.2.1.10 Complex Type: getParcelLabelNumberForWebNumber

Field name	Data type / Occurrence / Range	Description
webNumber	xsd:string / 1 .. 1 / minLength: 1 maxLength: 9	The web information number.

2.2.1.11 Complex Type: *getParcelLabelNumberForWebNumberResponse*

Field name	Data type / Occurrence / Range	Description
parcelLabelNumber	xsd:string / 0 .. 1 /	The parcel label number.

2.2.1.12 Complex Type: *getTrackingData*

Field name	Data type / Occurrence / Range	Description
parcelLabelNumber	xsd:string / 1 .. 1 / length: 14	Parcel label number

2.2.1.13 Complex Type: *getTrackingDataResponse*

Description

Response object of method *getTrackingData*.

Field name	Data type / Occurrence / Range	Description
trackingresult	types:TrackingResult / 1 .. 1 /	Result object with tracking data.

2.2.2 Namespace: *types_authentication*

URL: <http://dpd.com/common/service/types/Authentication/2.0>

2.2.2.1 Anonymous Complex Type of Element: *authentication*

Description

The data structure for authentication data.

Field name	Data type / Occurrence / Range	Description
delisId	xsd:string / 1 .. 1 / minLength: 8 maxLength: 10	The delis user id for authentication.
authToken	xsd:string / 1 .. 1 / minLength: 0 maxLength: 64	The token for authentication. Field authToken of Login, as a result of Method "getAuth" of LoginService.
messageLanguage	xsd:string / 1 .. 1 / minLength: 5 maxLength: 5	The language (Java format) for messages. "de_DE" for german messages. "en_US" for english messages.

2.2.2.2 Anonymous Complex Type of Element: authenticationFault

Description

The data structure for authentication errors.

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 /	The error code. Valid codes: NoAccessAuthenticationFault=-1 NoRightsAuthenticationFault=-2
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	The error message text.

2.3 Elements

2.3.1 Namespace: types

URL: <http://dpd.com/common/service/types/ParcelLifeCycleService/2.0>

2.3.1.1 Element: dataFault

Field name	Data type / Occurrence / Range	Description
dataFault	types:DataFault / 1 .. 1 /	Thrown if a data fault occurs.

2.3.1.2 Element: getParcelLabelNumberForWebNumber

Field name	Data type / Occurrence / Range	Description
getParcelLabelNumber ForWebNumber	types:getParcelLabel NumberForWebNumber / 1 .. 1 /	Input parameter of getParcelLabelNumberForWebNumber.

2.3.1.3 Element: getParcelLabelNumberForWebNumberResponse

Field name	Data type / Occurrence / Range	Description
getParcelLabelNumber ForWebNumber Response	types:getParcelLabel NumberForWebNumber Response / 1 .. 1 /	Output parameter of getParcelLabelNumberForWebNumber.

2.3.1.4 Element: *getTrackingData*

Field name	Data type / Occurrence / Range	Description
getTrackingData	types:getTrackingData / 1 .. 1 /	Input parameter of getTrackingData.

2.3.1.5 Element: *getTrackingDataResponse*

Field name	Data type / Occurrence / Range	Description
getTrackingData Response	types:getTrackingData Response / 1 .. 1 /	Output parameter of getTrackingData.

2.3.1.6 Element: *systemFault*

Field name	Data type / Occurrence / Range	Description
systemFault	types:SystemFault / 1 .. 1 /	Thrown if an internal system fault occurs.

2.3.2 Namespace: *types_authentication*

URL: <http://dpd.com/common/service/types/Authentication/2.0>

2.3.2.1 Element: *authentication*

Anonymous Complex Type *types_authentication:authentication*

2.3.2.2 Element: *authenticationFault*

Anonymous Complex Type *types_authentication:authenticationFault*

3. ParcelShopFinderService

3.1 Methods

Following methods are available for the service ParcelShopFinderService-Public_5_0

Method name	Description
findCities	Find cities with DPD parcel shops by country, zipCode and city.
findParcelShops	Find the DPD parcel shops by country/countries, zipCode and city and additional criteria like flags for properties of parcel shops. The search country can either domestic country or cross border country/countries for the search.
findParcelShopsByGeoData	Find the DPD parcel shops by geo data and additional criteria like flags for properties of parcel shops. The search country can either domestic country or cross border country/countries for the search.
getAvailableServices	Find available services of DPD parcel shops by country.

3.1.1 Method: findCities

Description

Find cities with DPD parcel shops by country, zipCode and city.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0" xmlns:ns1="http://dpd.com/common/service/types/ParcelShopFinderService/5.0">
  <soapenv:Header>
    <ns:authentication>
      <delisId><your delisId></delisId>
      <authToken><your authToken, get by LoginService></authToken>
      <messageLanguage>de_DE</messageLanguage>
    </ns:authentication>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:findCities>
      <country>DE</country>
      <zipCode>63741</zipCode>
      <city>Aschaffenburg</city>
    </ns1:findCities>
  </soapenv:Body>
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/ParcelShopFinderService/V5_0/findCities

JSON-Data for POST-Request (for GET-Request data must send with parameter "request" and for JSONP the Parameter "jsonpcallback" can be used, to send a padding function name):

```
{ "authentication" : {
  "delisId" : "<your delisId>",
  "authToken" : "<your authToken, get by LoginService>",
  "messageLanguage" : "de_DE"
},
  "findCities" : {
```

```

    "country" : "DE",
    "zipCode" : "63741",
    "city" : "Aschaffenburg"
  }
}

```

REST-Result, without error:

```

{
  "findCitiesResponse": {
    "city": [
      {
        "country": "DE",
        "countryNum": "276",
        "zipCode": "63741",
        "name": "Aschaffenburg",
        "town": ""
      },
      {
        "country": "DE",
        "city": "Aschaffenburg",
        "zipCode": "63741",
        "district": "Damm"
      },
      ...
    ],
    "status": {
      "type": "OK",
      "code": "200",
      "message": "valid"
    }
  }
}

```

REST-Result, if an error occurred:

```

{"status": {
  "type": "<Fault type, e.g. DataFault",
  "code": "<Error code, e.g. COMMON_1>",
  "message": "<Error message, e.g. invalid data>"
}}

```

Additional REST-State on system error:

Error-Code: 100

Error-Type: SystemFault

Error-Message: <Message, which describe the error>

Action

<http://dpd.com/common/service/ParcelShopFinderService/5.0/findCities>

Input

The input for method findCities is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:findCities

* header parameter

Output

The output for method findCities is as follows:

Name	Element
parameters	types:findCitiesResponse

Faults

The defined faults for method findCities are as follows:

Name	Element
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

3.1.1.1 Method: findParcelShops

Description

Find the DPD parcel shops by country/countries, zipCode and city and additional criteria like flags for properties of parcel shops.

The search country can either domestic country or cross border country/countries for the search.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0" xmlns:ns1="http://dpd.com/common/service/types/ParcelShopFinderService/5.0">
```

```
  <soapenv:Header>
    <ns:authentication>
      <delisId><your delisId></delisId>
      <authToken><your authToken></authToken>
      <messageLanguage>de_DE</messageLanguage>
    </ns:authentication>
  </soapenv:Header>
```

```
  <soapenv:Body>
    <ns1:findParcelShops>
      <country>DE</country>
      <zipCode>60388</zipCode>
      <street>Casimirstraße</street>
      <limit>10</limit>
      <availabilityDate>2014-11-10 16:00</availabilityDate>
      <hideClosed>true</hideClosed>
    </ns1:findParcelShops>
  </soapenv:Body>
```

```
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/ParcelShopFinderService/V5_0/findParcelShops

JSON-Data for POST-Request (for GET-Request data must send with parameter "request" and for JSONP the Parameter "jsonpcallback" can be used, to send a padding function name):

```
{ "authentication" : {
  "delisId" : "<your delisId>",
  "authToken" : "<authtoken, get by LoginService>",
  "messageLanguage" : "de_DE"
```



```

},
"findParcelShops": {
  "country": "DE",

  "zipCode": "60388",
  "street": "Casimirstraße",
  "limit": "10",
  "availabilityDate": "2014-11-10 16:00",
  "hideClosed": "true"
}
}

```

REST-Result, without error:

```

{
  "parcelSh"findParcelShopsResponse": {"parcelShop": [{
    opld": 272195,
    "pudold": "DE16935",
    "customerNo": "01600000000238770",
    "companyDepotDatabaseld": 132,
    "company": "Hintergässer Hof",
    "street": "Alt Bischofsheim",
    "houseNo": "22",
    "country": "DE",
    "countryNum": 276,
    "state": "Hessen",
    "zipCode": "63477",
    "city": "Maintal",
    "phone": "",
    "fax": "",
    "email": "",
    ...
    "openingHours": [
      {
        "weekday": "Mittwoch",
        "weekdayNum": 3,
        "openMorning": "09:00",
        "closeMorning": "13:00",
        "openAfternoon": "15:00",
        "closeAfternoon": "18:00",
        "dayOff": false
      },
      ...
    ],
    "services": {"service": [
      {
        "code": "100",
        "available": true
      },
      {
        "code": "991",
        "available": true
      },
      {
        "code": "202",
        "available": true,
        "serviceDetail": [
          {"code": "075"},

```

```

        {"code": "048"},
        {"code": "095"},
        ...
    ]
}
]]
...
}},
"status": {
    "type": "OK",
    "code": "200",
    "message": "valid"
}
}

```

REST-Result, if an error occurred:

```

{"status": {
    "type": "<Fault type, e.g. DataFault",
    "code": "<Error code, e.g. COMMON_1>",
    "message": "<Error message, e.g. invalid data>"
}}

```

Additional REST-State on system error:

Error-Code: 100

Error-Type: DataFault

Error-Message: <Message, which describe the error>

Action

<http://dpd.com/common/service/ParcelShopFinderService/5.0/findParcelShops>

Input

The input for method findParcelShops is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:findParcelShops

* header parameter

Output

The output for method findParcelShops is as follows:

Name	Element
parameters	types:findParcelShopsResponse

Faults

The defined faults for method findParcelShops are as follows:

Name	Element
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

3.1.1.2 Method: findParcelShopsByGeoData

Description

Find the DPD parcel shops by geo data and additional criteria like flags for properties of parcel shops. The search country can either domestic country or cross border country/countries for the search.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0" xmlns:ns1="http://dpd.com/common/service/types/ParcelShopFinderService/5.0">
  <soapenv:Header>
    <ns:authentication>
      <delisId><your delisId></delisId>
      <authToken><your authToken></authToken>
      <messageLanguage>de_DE</messageLanguage>
    </ns:authentication>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:findParcelShopsByGeoData>
      <longitude>8.1</longitude>
      <latitude>50.9</latitude>
      <limit>1</limit>
    </ns1:findParcelShopsByGeoData>
  </soapenv:Body>
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/ParcelShopFinderService/V5_0/findParcelShopsByGeoData JSON-Data for POST-Request (for GET-Request data must send with parameter "request" and for JSONP the Parameter "jsonpcallback" can be used, to send a padding function name):

```
{ "authentication" : {
  "delisId" : "<your delisId>",
  "authToken" : "<authToken, get by LoginService>",
  "messageLanguage" : "de_DE"
},
"findParcelShopsByGeoData" : {
  "longitude": "8.1",
  "latitude": "50.9",
  "limit": "1"
}
}
```

REST-Result, without error:

```
{
  "findParcelShopsResponse": {"parcelShop": [ {
    "parcelShopId": 273613,
    "puId": "DE17530",
    "customerNo": "90000008015717530",
    "companyDepotDatabasId": 129,
    "company": "NKD Deutschland GmbH 2580",
    "street": "Hauptmarkt",
    "houseNo": "16-18",
    "country": "DE",
    "countryNum": 276,
    "state": "Nordrhein-Westfalen",
```

```

"zipCode": "57076",
"city": "Siegen",
"phone": "",
"fax": "",
"email": "",
...
"services": {"service": [
  {
    "code": "100",
    "available": true
  },
  {
    "code": "991",
    "available": true
  },
  {
    "code": "200",
    "available": true
  }
]}
...
}},
"status": {
  "type": "OK",
  "code": "200",
  "message": "valid"
}
}

```

REST-Result, if an error occurred:

```

{"status": {
  "type": "<Fault type, e.g. DataFault",
  "code": "<Error code, e.g. COMMON_1>",
  "message": "<Error message, e.g. invalid data>"
}}

```

Additional REST-State on system error:

Error-Code: 100
 Error-Type: DataFault
 Error-Message: <Message, which describe the error>

Action

<http://dpd.com/common/service/ParcelShopFinderService/5.0/findParcelShopsByGeoData>

Input

The input for method findParcelShopsByGeoData is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:findParcelShopsByGeoData

* header parameter

Output

The output for method findParcelShopsByGeoData is as follows:

Name	Element
parameters	types:findParcelShopsByGeoDataResponse

Faults

The defined faults for method findParcelShopsByGeoData are as follows:

Name	Element
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault

Name	Element
dataFault	types:dataFault

3.1.1.3 Method: *getAvailableServices*

Description

Find available services of DPD parcel shops by country.

Sample request:

SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0" xmlns:ns1="http://dpd.com/common/service/types/ParcelShopFinderService/5.0">
  <soapenv:Header>
    <ns:authentication>
      <delisId><your delisId></delisId>
      <authToken><your authToken></authToken>
      <messageLanguage>de_DE</messageLanguage>
    </ns:authentication>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:getAvailableServices />
  </soapenv:Body>
</soapenv:Envelope>
```

REST:

URL: <server:port><path>/ParcelShopFinderService/V5_0/getAvailableServices
 JSON-Data for POST-Request (for GET-Request data must send with parameter "request" and for JSONP the Parameter "jsonpcallback" can be used, to send a padding function name):

```
{ "authentication" : {
  "delisId" : "<your delisId>",
  "authToken" : "<authtoken, get by LoginService>",
  "messageLanguage" : "de_DE"
},
  "getAvailableServices" : {
  }
}
```

REST-Result, without error:

```
{
  "getAvailableServicesResponse": {
    "service": [
      {
        "code": "006",
        "available": "true",
        "description": "Express dropoff",
        "serviceDetail": [
          {
            "code": "002",
            "description": "Express cut off",
          }
        ]
      },
      {
        "code": "003",
        "available": "true",
        "description": "COD",
        "serviceDetail": [
          {
            "code": "001",
            "description": "COD Payment mean",
          }
        ]
      }
    ]
  }
}
```

REST-Result, if an error occurred:

```
{"status": {
  "type": "<Fault type, e.g. DataFault",
  "code": "<Error code, e.g. COMMON_1>",
  "message": "<Error message, e.g. invalid data>"
}}
```

Additional REST-State on system error:

Error-Code: 100

Error-Type: DataFault

Error-Message: <Message, which describe the error>

Action

<http://dpd.com/common/service/ParcelShopFinderService/5.0/getAvailableServices>

Input

The input for method getAvailableServices is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:getAvailableServices

* header parameter

Output

The output for method `getAvailableServices` is as follows:

Name	Element
parameters	types:getAvailableServicesResponse

Faults

The defined faults for method `getAvailableServices` are as follows:

Name	Element
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

3.2 Complex Types

3.2.1 Namespace: types

URL: <http://dpd.com/common/service/types/ParcelShopFinderService/5.0>

3.2.1.1 Complex Type: CityType

Description

The city type with city, zip code and town.

Field name	Data type / Occurrence / Range	Description
country	xsd:string / 0 .. 1 / maxLength: 2	The country code in ISO-A2 format.
countryNum	xsd:int / 0 .. 1 / minInclusive: 0 maxInclusive: 999	Country number in ISO-Num3 format (e.g. Germany = 276).
zipCode	xsd:string / 0 .. 1 /	The zip code of the city.
name	xsd:string / 0 .. 1 /	The name of the city.
town	xsd:string / 0 .. 1 /	The town of the city.

3.2.1.2 Complex Type: DataFaultType

Description

Is thrown if a data fault occurs.

Field name	Data type / Occurrence / Range	Description
additionalData	xsd:string / 0 .. 1 /	Additional data for an error.

additionalInfo	xsd:string / 0 .. 1 /	Additional info for an error in user language.
errorClass	xsd:string / 0 .. 1 /	The class which creates the error.
errorCode	xsd:string / 0 .. 1 /	The error code.
fullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in user language.
language	xsd:string / 0 .. 1 /	The language for messages.
message	xsd:string / 0 .. 1 /	The complete message text for the error in user language.
shortMessage	xsd:string / 0 .. 1 /	The short info text for the error in user language.
systemFullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in system language.
systemMessage	xsd:string / 0 .. 1 /	The complete message text for the error in system language.
systemShortMessage	xsd:string / 0 .. 1 /	The short info text for the error in system language.

3.2.1.3 Complex Type: FaultType

Field name	Data type / Occurrence / Range	Description
additionalData	xsd:string / 0 .. 1 /	Additional data for an error.
additionalInfo	xsd:string / 0 .. 1 /	Additional info for an error in user language.
errorClass	xsd:string / 0 .. 1 /	The class which creates the error.
errorCode	xsd:string / 0 .. 1 /	The error code.
fullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in user language.
language	xsd:string / 0 .. 1 /	The language for messages.
message	xsd:string / 0 .. 1 /	The complete message text for the error in user language.
shortMessage	xsd:string / 0 .. 1 /	The short info text for the error in user language.
systemFullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in system language.
systemMessage	xsd:string / 0 .. 1 /	The complete message text for the error in system language.
systemShortMessage	xsd:string / 0 .. 1 /	The short info text for the error in system language.

3.2.1.4 Complex Type: FindCitiesResponseType

Description

The result data for method findCities.

Field name	Data type / Occurrence / Range	Description
city	types:CityType / 0 .. * /	A list of cities as result of the search. The list is sorted by city name.

3.2.1.5 Complex Type: FindCitiesType

Description

The search parameters for method findCities.

Field name	Data type / Occurrence / Range	Description
country	types:CountryType / 1 .. 1 /	The country for the search in ISO-A2 or ISO-Num3.
zipCode	xsd:string / 0 .. 1 / maxLength: 9	The zip code of the search origin.
city	xsd:string / 0 .. 1 / maxLength: 100	The city of the search origin.
limit	xsd:int / 1 .. 1 / minInclusive: 1 maxInclusive: 100	Max. number of parcel shop records in the response.
order	xsd:string / 1 .. 1 / ZIP_CODE NAME	The field (combination) for sorting the result: Zip code or city/town name.

3.2.1.6 Complex Type: FindParcelShopsByGeoDataResponseType

Description

The parcel shops as result of the search.

Field name	Data type / Occurrence / Range	Description
parcelShop	types:ParcelShopType / 0 .. * /	A list of parcel shop data containing the search result.

3.2.1.7 Complex Type: FindParcelShopsByGeoDataType

Description

The request parameters with coordinates in form of longitude and latitude.

Field name	Data type / Occurrence / Range	Description
longitude	xsd:double / 1 .. 1 /	The longitude of the search origin.
latitude	xsd:double / 1 .. 1 /	The latitude of the search origin.
limit	xsd:int / 1 .. 1 / minInclusive: 1 maxInclusive: 100	Max. number of parcel shop records in the response.
availabilityDate	xsd:string / 0 .. 1 / minLength: 10 maxLength: 16	If this is set the request searches for shop holidays from the given availability date. Otherwise holidays will be ignored. The expected format is yyyy-MM-dd. If it is wished to regard the opening hours too, this field must contain a time and the format is yyyy-MM-dd HH:mm.
hideClosed	xsd:boolean / 0 .. 1 /	If set to true, parcel shops, which are closed on the availabilityDate are not returned. Requires that the availabilityDate contains a valid time.
searchCountry	types:CountryType / 0 .. * /	The country/countries of the search origin. It contains domestic country or cross border countries for the search.
services	types:ServicesRequest Type / 0 .. 1 /	Contains the search parameters for parcel shop services.

3.2.1.8 Complex Type: FindParcelShopsResponseType

Description

The parcel shops as result of the search.

Field name	Data type / Occurrence / Range	Description
parcelShop	types:ParcelShopType / 0 .. * /	A list of parcel shops containing the search result.

3.2.1.9 Complex Type: FindParcelShopsType

Description

The request parameters for the search.

Field name	Data type / Occurrence / Range	Description
country	types:CountryType / 1 .. 1 /	The country for the search in ISO-A2 or ISO-Num3.

zipCode	xsd:string / 0 .. 1 / maxLength: 9	The zip code of the search origin.
city	xsd:string / 0 .. 1 / maxLength: 100	The city of the search origin.
street	xsd:string / 0 .. 1 / maxLength: 100	The street name of the search origin.
houseNo	xsd:string / 0 .. 1 / maxLength: 10	The house number of the search origin.
limit	xsd:int / 1 .. 1 / minInclusive: 1 maxInclusive: 100	Max. number of parcel shop records in the response.
availabilityDate	xsd:string / 0 .. 1 / minLength: 10 maxLength: 16	If this is set the request searches for shop holidays from the given availability date. Otherwise holidays will be ignored. The expected format is yyyy-MM-dd. If it is wished to regard the opening hours too, this field must contain a time and the format is yyyy-MM-dd HH:mm.
hideClosed	xsd:boolean / 0 .. 1 /	If set to true, parcel shops, which are closed on the availabilityDate are not returned. Requires that the availabilityDate contains a valid time.
searchCountry	types:CountryType / 0 .. * /	The country/countries of the search origin. It contains domestic country or cross border countries for the search.
services	types:ServicesRequest Type / 0 .. 1 /	Contains the search parameters for parcel shop services.

3.2.1.10 Complex Type: *GetAvailableServicesResponseType*

Description

The result data for method getAvailableServices.

Field name	Data type / Occurrence / Range	Description
services	types:ServicesType / 0 .. 1 /	Contains the result data for supported parcel shop services.

3.2.1.11 Complex Type: *GetAvailableServicesType*

Description

The search parameters for method `getAvailableServices`.

Field name	Data type / Occurrence / Range	Description
Complex Type contains no elements		

3.2.1.12 Complex Type: *HolidayType*

Description

A Data container for holiday.

Field name	Data type / Occurrence / Range	Description
holidayStart	xsd:string / 0 .. 1 /	The start date of holiday in date format "yyyy-MM-dd HH:mm:ss".
holidayEnd	xsd:string / 0 .. 1 /	The end date of holiday in date format "yyyy-MM-dd HH:mm:ss".

3.2.1.13 Complex Type: *OpeningHoursType*

Description

The data structure for "opening hour" data.

Field name	Data type / Occurrence / Range	Description
weekday	xsd:string / 0 .. 1 /	The localized name of the weekday.
openMorning	xsd:string / 0 .. 1 /	The time of opening of the shop in the morning.
closeMorning	xsd:string / 0 .. 1 /	The time of closing of the shop at the morning.
closeAfternoon	xsd:string / 0 .. 1 /	The time of closing of the shop at the afternoon.
openAfternoon	xsd:string / 0 .. 1 /	The time of opening of the shop at the afternoon.

3.2.1.14 Complex Type: ParcelShopType

Description

The data structure for parcel shop data, as the result of the search.

Field name	Data type / Occurrence / Range	Description
parcelShopId	xsd:long / 0 .. 1 / minInclusive: 1 maxInclusive: 999999999 999999999	The ID of the parcel shop.
pudold	xsd:string / 0 .. 1 / maxLength: 7	The pудо ID of the parcel shop.
company	xsd:string / 0 .. 1 / maxLength: 100	The company name of the shop.
street	xsd:string / 0 .. 1 / maxLength: 100	The street, sometimes combined with house number.
houseNo	xsd:string / 0 .. 1 / maxLength: 8	The house number of the shop.
country	xsd:string / 0 .. 1 / maxLength: 2	The country code in ISO-A2 format.
countryNum	xsd:int / 0 .. 1 / minInclusive: 0 maxInclusive: 999	Country number in ISO-Num3 format (e.g. Germany = 276).
state	xsd:string / 0 .. 1 / maxLength: 22	The name of the state.
zipCode	xsd:string / 0 .. 1 / maxLength: 10	The zip code of the shop address.
city	xsd:string / 0 .. 1 / maxLength: 100	The city of the shop address.
town	xsd:string / 0 .. 1 / maxLength: 35	The town (part of the city) of the shop.
phone	xsd:string / 0 .. 1 / maxLength: 50	The phone number of the shop.
fax	xsd:string / 0 .. 1 / maxLength: 50	The fax number of the shop.

Email	xsd:string / 0 .. 1 / maxLength: 100	The email address of the shop.
homepage	xsd:string / 0 .. 1 / maxLength: 100	The homepage of the shop.
latitude	xsd:double / 0 .. 1 /	The latitude of the shop position.
longitude	xsd:double / 0 .. 1 /	The longitude of the shop position.
coordinateX	xsd:double / 0 .. 1 /	The X geo coordinate of the shop position.
coordinateY	xsd:double / 0 .. 1 /	The Y geo coordinate of the shop position.
coordinateZ	xsd:double / 0 .. 1 /	The Z geo coordinate of the shop position.
distance	xsd:double / 0 .. 1 /	The distance to the origin of the search.
expressPickupTime	xsd:string / 0 .. 1 / maxLength: 5	The time when express deliveries are picked up.
extralInfo	xsd:string / 0 .. 1 / maxLength: 300	Additional information to find the parcel shop (e.g. if the shop is in a mall).
openingHours	types:OpeningHoursType / 0 .. * /	A List of opening hours.
holiday	types:HolidayType / 0 .. * /	A List of shop vacation times.
services	types:ServicesType / 0 .. 1 /	Contains the result data for supported parcel shop services.

3.2.1.15 Complex Type: ServiceDetailRequestType

Description

The search parameters for parcel shop service details.

Field name	Data type / Occurrence / Range	Description
code	types:CodeType / 1 .. 1 /	The service code.

3.2.1.16 Complex Type: ServiceDetailType

Description

A data container for parcel shop service details.

Field name	Data type / Occurrence / Range	Description
code	types:CodeType / 1 .. 1 /	The service code.
description	types:DescriptionType / 0 .. 1 /	The description of service code detail.

3.2.1.17 Complex Type: ServiceRequestType

Description

The search parameters for parcel shop services with service details.

Field name	Data type / Occurrence / Range	Description
code	types:CodeType / 1 .. 1 /	The service code.
available	xsd:boolean / 1 .. 1 /	This field specifies whether allowed for searching of parcel shops or not.
serviceDetail	types:ServiceDetailRequestType / 0 .. * /	A list of service detail information.

3.2.1.18 Complex Type: ServiceType

Description

The result data for supported parcel shop services with service details.

Field name	Data type / Occurrence / Range	Description
code	types:CodeType / 1 .. 1 /	The service code.
available	xsd:boolean / 1 .. 1 /	This field specifies whether allowed for searching of parcel shops or not.
description	types:DescriptionType / 0 .. 1 /	The description of service code.
serviceDetail	types:ServiceDetailType / 0 .. * /	A list of service detail information.

3.2.1.19 Complex Type: ServicesRequestType

Description

Contains the search parameters for parcel shop services.

Field name	Data type / Occurrence / Range	Description
service	types:ServiceRequest Type / 0 .. * /	A List of services.

3.2.1.20 Complex Type: ServicesType

Description

Contains the result data for supported parcel shop services.

Field name	Data type / Occurrence / Range	Description
service	types:ServiceType / 0 .. * /	A List of supported services.

3.2.1.21 Complex Type: SystemFaultType

Description

Is thrown if an internal system fault occurs.

Field name	Data type / Occurrence / Range	Description
additionalData	xsd:string / 0 .. 1 /	Additional data for an error.
additionalInfo	xsd:string / 0 .. 1 /	Additional info for an error in user language.
errorClass	xsd:string / 0 .. 1 /	The class which creates the error.
errorCode	xsd:string / 0 .. 1 /	The error code.

Field name	Data type / Occurrence / Range	Description
fullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in user language.
language	xsd:string / 0 .. 1 /	The language for messages.
message	xsd:string / 0 .. 1 /	The complete message text for the error in user language.
shortMessage	xsd:string / 0 .. 1 /	The short info text for the error in user language.

systemFullMessage	xsd:string / 0 .. 1 /	The complete message text for the error with additional info in system language.
systemMessage	xsd:string / 0 .. 1 /	The complete message text for the error in system language.
systemShortMessage	xsd:string / 0 .. 1 /	The short info text for the error in system language.

3.3 Namespace: types_authentication

URL: <http://dpd.com/common/service/types/Authentication/2.0>

3.3.1 Types_authentication : anonymous

3.3.1.1 Anonymous Complex Type of Element: authentication

Description

The data structure for authentication data.

Field name	Data type / Occurrence / Range	Description
delisId	xsd:string / 1 .. 1 / minLength: 8 maxLength: 10	The delis user id for authentication.
authToken	xsd:string / 1 .. 1 / minLength: 0 maxLength: 64	The token for authentication. Field authToken of Login, as a result of Method "getAuth" of LoginService.
messageLanguage	xsd:string / 1 .. 1 / minLength: 5 maxLength: 5	The language (Java format) for messages. "de_DE" for german messages. "en_US" for english messages.

3.3.1.2 Anonymous Complex Type of Element: authenticationFault

Description

The data structure for authentication errors.

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 /	The error code. Valid codes: NoAccessAuthenticationFault=-1 NoRightsAuthenticationFault=-2
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	The error message text.

3.4 Simple Types

3.4.1 Namespace: types

URL: <http://dpd.com/common/service/types/ParcelShopFinderService/5.0>

3.4.1.1 Simple Type: CodeType

Field name	Data type / Occurrence / Range	Description
CodeType	xsd:string / / length: 3	The code of service and service detail.

3.4.1.2 Simple Type: CountryType

Field name	Data type / Occurrence / Range	Description
CountryType	xsd:string / / minLength: 1 maxLength: 3	Country ISO-A2 code (e.g. Germany = DE) or country number in ISO-Num3 format (e.g. Germany = 276).

3.4.1.3 Simple Type: DescriptionType

Field name	Data type / Occurrence / Range	Description
DescriptionType	xsd:string / / maxLength: 500	The description of service and service detail.

3.4.1.4 Simple Type: MessageLanguageType

Field name	Data type / Occurrence / Range	Description
MessageLanguageType	xsd:string / / minLength: 5 maxLength: 5	The language (Java format) for messages. - "de_DE" for german messages. - "en_US" for english messages.

3.5 Elements

3.5.1 Namespace: types

URL: <http://dpd.com/common/service/types/ParcelShopFinderService/5.0>

3.5.1.1 Element: dataFault

Field name	Data type / Occurrence / Range	Description
dataFault	types:DataFaultType / 1 .. 1 /	Thrown if a data fault occurs.

3.5.1.2 Element: findCities

Field name	Data type / Occurrence / Range	Description
findCities	types:FindCitiesType / 1 .. 1 /	Input parameters of findCities.

3.5.1.3 Element: findCitiesResponse

Field name	Data type / Occurrence / Range	Description
findCitiesResponse	types:FindCitiesResponseType / 1 .. 1 /	Output parameters of findCitiesResponse.

3.5.1.4 Element: findParcelShops

Field name	Data type / Occurrence / Range	Description
findParcelShops	types:FindParcelShopsType / 1 .. 1 /	Input parameters of findParcelShops.

3.5.1.5 Element: findParcelShopsByGeoData

Field name	Data type / Occurrence / Range	Description
findParcelShopsByGeoData	types:FindParcelShopsByGeoDataType / 1 .. 1 /	Input parameters of findParcelShopsByGeoData.

3.5.1.6 Element: findParcelShopsByGeoDataResponse

Field name	Data type / Occurrence / Range	Description
findParcelShopsByGeoDataResponse	types:FindParcelShopsByGeoDataResponseType / 1 .. 1 /	Output parameters of findParcelShopsByGeoDataResponse.

3.5.1.7 Element: *findParcelShopsResponse*

Field name	Data type / Occurrence / Range	Description
findParcelShops Response	types:FindParcelShopsResponseType / 1 .. 1 /	Output parameters of findParcelShops.

3.5.1.8 Element: *getAvailableServices*

Field name	Data type / Occurrence / Range	Description
getAvailableServices	types:GetAvailableServicesType / 1 .. 1 /	Input parameters of getAvailableServices.

3.5.1.9 Element: *getAvailableServicesResponse*

Field name	Data type / Occurrence / Range	Description
getAvailableServices Response	types:GetAvailableServicesResponseType / 1 .. 1 /	Output parameters of getAvailableServicesResponse.

3.5.1.10 Element: *systemFault*

Field name	Data type / Occurrence / Range	Description
systemFault	types:SystemFaultType / 1 .. 1 /	Thrown if an internal system fault occurs.

3.5.2 Namespace: *types_authentication*

URL: <http://dpd.com/common/service/types/Authentication/2.0>

3.5.2.1 Element: *authentication*

Anonymous Complex Type *types_authentication:authentication*

3.5.2.2 Element: *authenticationFault*

Anonymous Complex Type *types_authentication:authenticationFault*

4. ShipmentService

4.1 History

4.1.1 Version 3.3.0

- New fields and descriptions in the address type
- New fault code in the faultCodeType
- Changed description of the mpsWeight in the GeneralShipmentData type
- New and deleted fields in the international type
- New complex type internationalLine
- Changed description of weight and new fields in the parcel type

4.1.2 Version 3.2.0

- New field extraPickup in pickup
- New field startPosition in printOptions

4.1.3 Version 3.1.0

- New field tyres in productAndServiceData
- Field customerNumber in sender/recipient address and field invoicePosition in international must be numeric.
- Parcelblock is optional, for only one parcel with no additional data.
- Return parcel label in format A7 support.
- Return parcel label receipt support.
- Variable sender address support.
- Parcel with 0kg weight handle as classic parcel.

4.1.4 Version 3.0.0

- New field countrySpecificService in productAndServiceData
- Removed field paperFormat in storeOrders
- Introduced new element printOptions in storeOrders with paperformat, printertype and printer options

4.2 Methods

4.2.1 Following methods are available for the service ShipmentService-Public_3_3

Method name	Description
storeOrders	Stores up to 30 shipments and creates corresponding shipment documents.

4.2.2 Method: storeOrders

4.2.2.1 Description

Stores up to 30 shipments and creates corresponding shipment documents.

Sample request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0" xmlns:ns1="http://dpd.com/common/service/types/ShipmentService/3.3">
  <soapenv:Header>
    <ns:authentication>
      <delisId><your delisId></delisId>
```

```

    <authToken><your authToken></authToken>
    <messageLanguage>de_DE</messageLanguage>
  </ns:authentication>
</soapenv:Header>
<soapenv:Body>
  <ns1:storeOrders>
    <printOptions>
      <printerLanguage>PDF</printerLanguage>
      <paperFormat>A4</paperFormat>
    </printOptions>
    <order>
      <generalShipmentData>
        <identificationNumber>77777</identificationNumber>
        <sendingDepot>0163</sendingDepot>
        <product>CL</product>
        <mpsCompleteDelivery>0</mpsCompleteDelivery>
        <sender>
          <name1>Abholer1</name1>
          <street>Strasse 1</street>
          <country>DE</country>
          <zipCode>11111</zipCode>
          <city>Ort1</city>
          <customerNumber>12345679</customerNumber>
          <type>P</type>
        </sender>
        <recipient>
          <name1>Test-Empfaenger</name1>
          <street>Test-Strasse</street>
          <state>BY</state>
          <country>DE</country>
          <zipCode>63741</zipCode>
          <city>Aschaffenburg</city>
          <type>P</type>
        </recipient>
      </generalShipmentData>
      <parcels>
        <parcelLabelNumber></parcelLabelNumber>
      </parcels>
      <productAndServiceData>
        <orderType>consignment</orderType>
      </productAndServiceData>
    </order>
  </ns1:storeOrders>
</soapenv:Body>
</soapenv:Envelope>

```

Action

<http://dpd.com/common/service/ShipmentService/3.3/storeOrders>

4.2.2.2 Input

The input for method storeOrders is as follows:

Name	Element
authentication *	types_authentication:authentication
parameters	types:storeOrders

* header parameter

4.2.2.3 Output

The output for method storeOrders is as follows:

Name	Element
return	types:storeOrdersResponse

4.2.2.4 Faults

The defined faults for method storeOrders are as follows:

Name	Element
AuthenticationFault	types_authentication:authenticationFault

4.3 Complex Types

4.3.1 Namespace: types

URL: <http://dpd.com/common/service/types/ShipmentService/3.3>

4.3.2 Complex Type: address

Field name	Data type / Occurrence / Range	Description
name1	xs:string / 1 .. 1 / maxLength: 50	Name of address owner. For dangerous goods the maximum length is 50, otherwise always 35.
name2	xs:string / 0 .. 1 / maxLength: 35	Name 2 of address owner.
street	xs:string / 1 .. 1 / maxLength: 50	Street of address owner. For dangerous goods the maximum length is 50, otherwise always 35.
houseNo	xs:string / 0 .. 1 / maxLength: 8	House number of address owner. Maximum length is 8.
state	xs:string / 0 .. 1 / length: 2	State of address owner in ISO 3166-2 code (e.g. BY = Bayern).
country	xs:string / 1 .. 1 / length: 2	Country of address owner in ISO 3166-1 alpha-2 format (e.g. 'DE').

zipCode	xs:string / 1 .. 1 / maxLength: 9	Zip code of address owner.
city	xs:string / 1 .. 1 / maxLength: 50	City/town of address owner. For dangerous goods the maximum length is 50, otherwise always 35.
gln	xs:long / 0 .. 1 / maxInclusive: 99999999 99999	International location number of address owner.
customerNumber	xs:string / 0 .. 1 / maxLength: 17	Customer number of address owner. Maximum length is 17 for consignment and pickup information, 11 for collection request. Mandatory for sender's address.
type	xs:string / 1..1 / maxLength: 1	B = business P = private
contact	xs:string / 0 .. 1 / maxLength: 35	Contact person of address owner.
phone	xs:string / 0 .. 1 / maxLength: 30	Phone number of address owner. Mandatory if phone is the chosen notification channel. No required data format. Either phone or email is mandatory in the complex type international.
fax	xs:string / 0 .. 1 / maxLength: 30	Fax number of address owner. No required data format.
email	xs:string / 0 .. 1 / maxLength: 50	Email address of address owner. For collection requests the maximum length is 40, otherwise always 50. Either phone or email is mandatory in the complex type international.
comment	xs:string / 0 .. 1 / maxLength: 70	Comment on address owner.
iaccount	xs:string / 0 .. 1 / maxLength: 50	Account allocation or cost center (for VTG) of invoice data for consignments.

eoriNumber	xs:string / 0 .. 1 / maxLength: 20	If type = B: either eori or vatNumber is mandatory
vatNumber	xs:string / 0 .. 1 / maxLength: 20	If type = B: either eori or vatNumber is mandatory
idDocType	xs:string / 0 .. 1 / maxLength: 1	Type of identification document: P: PASSPORT Number C : CNPJ (ou CPF) Number (Brésil) if type = B this field becomes mandatory (dependency of the custom export regime) applicable or not according export custom clearance of origin country
idDocNumber	xs:string / 0 .. 1 / maxLength: 20	Identification document number
webSite	xs:string / 0 .. 1 / maxLength: 255	For sender: Selling website, it becomes mandatory if field type is B and the sender is a Ecommercant

Namespace: types

4.3.3 Complex Type: cod

Description

Bundles cash on delivery data.

Field name	Data type / Occurrence / Range	Description
amount	xs:long / 1 .. 1 / maxInclusive: 99999999 99	COD amount in the currency of the destination country. The amount is specified as integer value e.g. 300.00 becomes 30000. Please note the national ceilings for cod in the corresponding relations.
currency	xs:string / 1 .. 1 / length: 3	ISO 4217 alpha-3 currency code (destination country).
inkasso	xs:int / 1 .. 1 / 0 1	Defines collection type. Possible values are: = cash = crossed cheque
purpose	xs:string / 0 .. 1 / maxLength: 14	Purpose of use.
bankCode	xs:string / 0 .. 1 / maxLength: 25	Bank code. If this value is filled, parameters for bankName, bankAccountNumber and bankAccountHolder must also be filled.
bankName	xs:string / 0 .. 1 / maxLength: 27	Bank name. If this value is filled, parameters for bankCode, bankAccountNumber and bankAccountHolder must also be filled.

bankAccountNumber	xs:string / 0 .. 1 / maxLength: 25	Bank account number. If this value is filled, parameters for bankCode, bankName and bankAccountHolder must also be filled.
bankAccountHolder	xs:string / 0 .. 1 / maxLength: 30	Account holder. If this value is filled, parameters for bankCode, bankName and bankAccountNumber must also be filled.
iban	xs:string / 0 .. 1 / maxLength: 50	International bank account number . If this value is filled, parameters for bankCode, bankName, bankAccountNumber and bankAccountHolder must also be filled.
bic	xs:string / 0 .. 1 / maxLength: 50	Bank identifier code. If this value is filled, parameters for bankCode, bankName, bankAccountNumber and bankAccountHolder must also be filled.

Namespace: types

4.3.4 Complex Type: delivery

Description

Bundles delivery data.

Field name	Data type / Occurrence / Range	Description
day	xs:string / 0 .. 1 / maxLength: 20	Allowed delivery days in the week (0 = Sunday, 1 = Monday etc.). Comma separated list of possible delivery days (e.g. "2,3,4,5").
dateFrom	xs:int / 0 .. 1 / maxInclusive: 99999999	Fixed delivery from date - format YYYYMMDD, e.g. 20080213.
dateTo	xs:int / 0 .. 1 / maxInclusive: 99999999	Fixed delivery to date - format YYYYMMDD, e.g. 20080213.
timeFrom	xs:int / 0 .. 1 / maxInclusive: 2400	Time from which the consignee is available - format hhmm (local time recipient country), e.g. 1400 or 0830.
timeTo	xs:int / 0 .. 1 / maxInclusive: 2400	Time until the consignee is available - format hhmm (local time recipient country), e.g. 1600 or 0930.

Namespace: types

4.3.5 Complex Type: faultCodeType

Field name	Data type / Occurrence / Range	Description
faultCode	xs:string / 1 .. 1 /	<p>DATABASE_8 - delete fails on database level during rollback after some other error</p> <p>DATABASE_21 - order is to be inserted in database, but has already a database id</p> <p>DATABASE_22 - order is to be committed after inserting, but there is no database id for update given</p> <p>DATABASE_24 - deletion is necessary because of some other error, but no database id for record is given</p> <p>MPSEXP_1 - parcel label number is already in use</p> <p>ROUTING_1 - no route could be found for depot and feature</p> <p>ROUTING_2 - invalid input data</p> <p>ROUTING_3 - the connection to the database failed</p> <p>ROUTING_4 - an internal failure occurs</p> <p>ROUTING_5 - no depot could be found</p> <p>ROUTING_6 - no pickup date was given</p> <p>ROUTING_7 - internal extension rule error</p> <p>ROUTING_8 - malformed zip code</p> <p>ROUTING_9 - missing country code</p> <p>ROUTING_10 - missing routing place</p> <p>ROUTING_11 - missing service code</p> <p>ROUTING_12 - routing place error</p> <p>ROUTING_13 - sending date error</p> <p>ROUTING_14 - service code error</p> <p>ROUTING_15 - invalid relation of service codes</p> <p>ROUTING_16 - unknown country code</p> <p>ROUTING_17 - unknown destination depot</p> <p>ROUTING_18 - unknown zip code</p> <p>ROUTING_19 - unknown routing place</p> <p>ROUTING_20 - unknown service code</p> <p>ROUTING_21 - invalid service code</p> <p>ROUTING_21 - parcel is labelled as return parcel but returns is not possible</p> <p>ROUTING_22 - missing allow</p> <p>ROUTING_23 - missing routing place or destination depot</p> <p>CUSTOMS_1 – customs data are not complete</p>
message	xs:string / 1 .. 1 /	Message with detailed information for the fault (e.g. incorrect field).

Namespace: types

4.3.6 Complex Type: generalShipmentData

Description

Bundles general shipment data.

Field name	Data type / Occurrence / Range	Description
mpsId	xs:string / 0..1 / length: 25	The shipment number for consignment data. The shipment number is only accepted if the parcel label number is allocated by customer. It starts with one of "MPS", "EXP" or "B2C", the last eight ciphers are the date in format yyyyMMdd.
cUser	xs:string / 0..1 / maxLength: 10	User ID of the person who made the entry.
mpsCustomerReferenceNumber1	xs:string / 0..1 / maxLength: 50	Consignment customer reference number 1 (maximal length 35), also customer reference number for collection request orders (maximal length 50).
mpsCustomerReferenceNumber2	xs:string / 0..1 / maxLength: 35	Consignment customer reference number 2.
mpsCustomerReferenceNumber3	xs:string / 0..1 / maxLength: 35	Consignment customer reference number 3.
mpsCustomerReferenceNumber4	xs:string / 0..1 / maxLength: 35	Consignment customer reference number 4.
identificationNumber	xs:string / 0..1 / maxLength: 999	Serves as unique alphanumeric key of the shipment used by customer. Can obtain up to 999 ciphers.
sendingDepot	xs:string / 1..1 / length: 4	Sending depot for consignment, ordering depot for collection request, customer's depot for pickup information or creating/sending depot for dangerous goods. 4 alphanumeric places, including leading zeros, e.g. 0163.
product	xs:string / 1..1 / CL E830 E10 E12 E18 IE2 PL PL+ MAIL	Selection of product, exactly one per shipment, mandatory for consignment data. Possible values are: CL = DPD CLASSIC E830 = DPD 8:30 E10 = DPD 10:00 E12 = DPD 12:00 E18 = DPD 18:00 IE2 = DPD EXPRESS PL = DPD PARCELletter PL+ = DPD PARCELletterPlus MAIL = DPD International Mail
mpsCompleteDelivery	xs:boolean / 0..1 /	Defines if the shipment should be sent as complete delivery. Mandatory for COD. Default value is false.
mpsCompleteDeliveryLabel	xs:boolean / 0..1 /	Defines if the label for complete delivery is printed for pickup. Default value is false.

mpsVolume	xs:long / 0 .. 1 / maxInclusive: 999999999	Volume per consignment in cm3 (without decimal places).
mpsWeight	xs:long / 0 .. 1 / maxInclusive: 999999999	Shipment weight in grams units without decimal delimiter (e.g. 3000 equals 3kg). Sum of weight (complex type parcel) equal mpsWeight Sum of grossWeight (complex type international) equal mpsWeight
mpsExpectedSending Date	xs:string / 0 .. 1 / pattern: [2][0-9]{3}([0][0-9][1][0-2])(0[1-9][12][0-9] 3[01])	Date when the shipment is expected to be transferred to the system. Format is YYYYMMDD.
mpsExpectedSending Time	xs:string / 0 .. 1 / pattern: (([1][0-9][2][0-3] [0][0-9])([0-5][0-9])([0-5][0-9])) ([0-9]([0-5][0-9])([0-5][0-9]))	Time when the shipment is expected to be transferred to the system. Format is HHMMSS.
sender	types:address / 1 .. 1 /	Consignment sender's address, collection request customer's address or pickup information customer's address.
recipient	types:address / 1 .. 1 /	Address of the recipient. For parcel shop delivery address of the real recipient.

Namespace: types

4.3.7 Complex Type: hazardous

Description

Bundles hazardous materials data.

Field name	Data type / Occurrence / Range	Description
identificationUnNo	xs:string / 1 .. 1 / length: 4	Defines UN number of hazardous substance. UN numbers range between 0004 and 9004.
identificationClass	xs:string / 1 .. 1 / maxLength: 6	Defines class of hazardous substance. Possible values range from 1 to 9. Subclasses are specified as position after decimal point (e.g. class 1 is "explosives", class 1.6 is "extremely insensitive explosives").
classificationCode	xs:string / 0 .. 1 / maxLength: 5	Defines classification code of hazardous substance.

packingGroup	xs:string / 0 .. 1 / maxLength: 5	Defines packing group of hazardous substance. Common values are "I", "II" or "III".
packingCode	xs:string / 1 .. 1 / 0A 0A1 0A2	Defines packing code. Possible values are: 0A = thin sheet packing 0A1 = thin sheet packing with non removable head 0A2 = thin sheet packing with removable head 1A = steel barrel
	1A 1A1 1A2 1B 1B1 1B2 1H 1H1 1H2 3A 3A1 3A2 3B 3B1 3B2 3H 3H1 3H2 4A 4B 4D 4G 4H 4H1 4H2 5H 5M 6H	1A1 = steel barrel with non removable head 1A2 = steel barrel with removable head 1B = aluminium barrel 1B1 = aluminium barrel with non removable head 1B2 = aluminium barrel with removable head 1H = plastics barrel 1H1 = plastics barrel with non removable head 1H2 = plastics barrel with removable head 3A = steel canister 3A1 = steel canister with non removable head 3A2 = steel canister with removable head 3B = aluminium canister, 3B1 = aluminium canister with non removable head 3B2 = aluminium canister with removable head 3H = plastics canister, 3H1 = plastics canister with non removable head 3H2 = plastics canister with removable head 4A = steel crates 4B = aluminium crate 4D = plywood crate 4G = cardboard crate 4H = plastics crate 4H1 = plastics crate plastics expanded 4H2 = plastics crate plastics solid 5H = plastics bags 5M = paper bags 6H = combination packings
description	xs:string / 1 .. 1 / maxLength: 160	Description of hazardous substance (redundant).
subsidiaryRisk	xs:string / 0 .. 1 / maxLength: 10	Subsidiary risk of hazardous substance (redundant).
tunnelRestrictionCode	xs:string / 0 .. 1 / A B C D E	Tunnel restriction code of hazardous substance (redundant). Possible values are "A", "B", "C", "D" or "E".
hazardousWeight	xs:decimal / 1 .. 1 / totalDigits: 6 fractionDigits: 2	Weight of hazardous substance with 4 places before decimal point and 2 decimal places (with separator).
netWeight	xs:decimal / 0 .. 1 / totalDigits: 6 fractionDigits: 2	Net weight of hazardous substance with 4 places before decimal point and 2 decimal places (with separator).

factor	xs:int / 1 .. 1 / maxInclusive: 999	Factor of hazardous substance (redundant). 999 means unlimited.
notOtherwiseSpecified	xs:string / 0 .. 1 / maxLength: 150	A not otherwise specified text which is mandatory for some substances.

Namespace: types

4.3.8 Complex Type: higherInsurance

Field name	Data type / Occurrence / Range	Description
amount	xs:long / 1 .. 1 / maxInclusive: 99999999 99	Increased insurance value with 2 decimal point positions without separators.
currency	xs:string / 1 .. 1 / length: 3	Currency code for increased insurance in format ISO 4217 alpha 3.

Namespace: types

4.3.9 Complex Type: international

Field name	Data type / Occurrence / Range	Description
parcelType	xs:boolean / 1 .. 1 /	Defines if the type of parcel is "documents" (allowed for air based destination only). Default value is false.
customsAmount	xs:long / 1 .. 1 / maxInclusive: 99999999 9999	Defines the customs amount in total without decimal separator (e.g. 14.000 = 14000). For documents it is 0.
customsCurrency	xs:string / 1 .. 1 / length: 3	Currency code in ISO 4217 alpha-3 format.
customsAmountEx	xs:long / 1 .. 1 / maxInclusive: 99999999 9999	Defines the converted customs amount in total without decimal separator (e.g. 14.000 = 14000). For documents it is 0.
customsCurrencyEx	xs:string / 1 .. 1 / length: 3	Currency code in ISO 4217 alpha-3 format. Currency for conversion (destination country).

clearanceCleared	xs:string / 1 .. 1 / length: 1	(N)o, (F)ree, (E)xport Cleared, (T)ransit cleared, (I)mport Cleared, (H)ybrid.
prealertStatus	xs:string / 1 .. 1 / length: 3	S03 = Data captured by sending depot S04 = Data are available for export gateway depots S05 = Pre-alert provided (1st linehaul) S06 = Return to sending depot by export gateway S07 = Deactivated S08 =HV import customs cleared S09 = transit via road (T1) S13 = Pre-alert provided (2nd linehaul) S14 = Returned by import gateway (1st linehaul) S15 = LV import customs cleared S16 = Returns to sending country (2nd linehaul)
exportReason	xs:string / 1 .. 1 / length: 2	Reason for export : 01: sale 02: return/replacement 03: gift
customsTerms	xs:string / 1 .. 1 / 01 02 03 05 06	Declares the customs terms. Possible values are: = DAP, cleared = DDP, delivered duty paid (incl. duties and excl. Taxes) = DDP, delivered duty paid (incl duties and taxes) = ex works (EXW) = DAP
customsContent	xs:string / 1 .. 1 / maxLength: 35	Describes the content of the parcel.
customsPaper	xs:string / 0 .. 1 / maxLength: 20	Lists the accompanying documents. The values are to be combined without separator (e.g. "ABG"). Following values are defined: = commercial invoice = pro forma invoice = export declaration = EUR1 = EUR2 = ATR = delivery note = third party billing = T1 document
customsEnclosure	xs:boolean / 0 .. 1 /	Defines if the accompanying documents are at the parcel or not. Default value is false.
customsInvoice	xs:string / 1 .. 1 / maxLength: 20	Defines the invoice number.
customsInvoiceDate	xs:int / 1 .. 1 / maxInclusive: 99999999	Defines the invoice date (format YYYYMMDD).

customsAmountParcel	xs:long / 0 .. 1 / maxInclusive: 99999999 9999	Parcel value with two decimal places without separator.
linehaul	xs:string / 0 .. 1 / AI RO	Defines mode of line haul. Possible values are "AI" for air and "RO" for road.
shipMrn	xs:string / 0 .. 1 / maxLength: 20	Movement reference number of the electrical export declaration.
collectiveCustoms Clearance	xs:boolean / 0 .. 1 /	Flag for determining collective customs clearance. Default value is false.
comment1	xs:string / 0 .. 1 / maxLength: 70	Comment.
comment2	xs:string / 0 .. 1 / maxLength: 70	Second comment.
commercialInvoice ConsigneeVatNumber	xs:string / 0 .. 1 / maxLength: 20	Defines the commercial invoice consignee VAT number.
commercialInvoice Consignee	types:address / 1 .. 1 /	Contains address data of commercial invoice consignee.
commercialInvoice Consignor	types:address / 1 .. 1 /	Contains address data of commercial invoice consignor.
commercialInvoiceLine	types:internationalLine / 1 .. * /	Contains invoice line data

4.3.10 Complex Type: internationalLine

Field name	Data type / Occurrence / Range	Description
customsTarif	xs:string / 1 .. 1 / maxLength: 10	The customs tarif number.
receiverCustomsTarif	xs:string / 1 .. 1 / maxLength: 8	The customs tarif number of the destination country.
productCode	xs:string / 0 .. 1 / maxLength: 255	Internal customer product code - mandatory for destination country UK
content	xs:string / 1 .. 1 / maxLength: 200	Nature of goods
grossWeight	xs:int / 1 .. 1 / maxInclusive: 99999999	Total weight of the article including packaging, in grams. This is not the unit weight. Weight in Grams units without decimal delimiter (e.g. 3000 equals 3kg)

itemsNumber	xs:int / 1 .. 1 / maxInclusive: 9999	Quantity of items
amountLine	xs:long / 1 .. 1 / maxInclusive: 99999999 9999	The customs amount of the article without decimal separator (e.g. 14.000 = 14000).
customsOrigin	xs:string / 1 .. 1 / length: 2	Origin country in ISO 3166-1 alpha-2 format (e.g. 'DE').
invoicePosition	xs:string / 0 .. 1 / maxLength: 6	The invoice position.

Namespace: types

4.3.11 Complex Type: notification

Field name	Data type / Occurrence / Range	Description
channel	xs:int / 1 .. 1 / 1 2 3	Defines channel of notification. Possible values are: = email = telephone = SMS
value	xs:string / 1 .. 1 / maxLength: 50	Value for the chosen channel, i.e. the phone number or the e-mail address. No required data format for SMS and phone.
language	xs:string / 0 .. 1 / length: 2	Language of the notification in ISO 3166-1 alpha-2 format (e.g. 'DE').

Namespace: types

4.3.12 Complex Type: parcel

Field name	Data type / Occurrence / Range	Description
parcelLabelNumber	xs:string / 0 .. 1 / minLength: 11 maxLength: 14	Parcel label number. Number range and validity are checked. Length can be 11 or 14.
customerReference Number1	xs:string / 0 .. 1 / maxLength: 35	Parcel customer reference number 1.
customerReference Number2	xs:string / 0 .. 1 / maxLength: 35	Parcel customer reference number 2.
customerReference Number3	xs:string / 0 .. 1 / maxLength: 35	Parcel customer reference number 3.

customerReference Number4	xs:string / 0 .. 1 / maxLength: 35	Parcel customer reference number 4.
swap	xs:boolean / 0 .. 1 /	Defines if the parcel is a consignment swap parcel. Default value is false.
volume	xs:int / 0 .. 1 / maxInclusive: 999999999	Volume of the single parcel (length/width/height in format LLLWWWHH) in cm without separators.
weight	xs:int / 0 .. 1 / maxInclusive: 99999999	Parcel weight in grams units without decimal delimiter (e.g. 3000 equals 3kg). Consignement allows 8-digit values, collection requests 7-digit and dangerous goods 6-digit.
hazardousLimited Quantities	xs:boolean / 0 .. 1 /	Defines if the parcel is a limited quantities hazardous goods parcel. Default value is false.
higherInsurance	types:higherInsurance / 0 .. 1 /	Defines if the parcel has increased insurance value. Default value is false.
content	xs:string / 0 .. 1 / maxLength: 50	Content of the parcel. Mandatory for increased insurance. For collection requests maximum length is 50, for consignment it is 35.
addService	xs:int / 0 .. 1 / 1 2 3	Additional service. Possible values are: = delivery information = documents return = written permission to deposit goods by sender
messageNumber	xs:int / 0 .. 1 / maxInclusive: 99999	Message number for consignment shipper information. Default value is 1.
function	xs:string / 0 .. 1 / LOCKDZB LOCKASG LOCKEVM LOCKSHOP LOCKTV	Blockable functions, possible values are: LOCKDZB = delivery to a third party with notification of delivery LOCKASG = written permission to deposit goods LOCKEVM = delivery with non-recurring authority LOCKSHOP = parcel shop LOCKTV = appointment
parameter	xs:string / 0 .. 1 / maxLength: 300	Free text for blockable functions.
cod	types:cod / 0 .. 1 /	Contains COD data for COD consignments.
international	types:international / 0 .. 1 /	Contains data for consignments across customs frontiers.
hazardous	types:hazardous / 0 .. 4 /	Contains packing and substance data for dangerous goods.
printInfo1OnParcelLabel	xs:boolean / 0 .. 1 /	Determines if content of parameter info1 will be added on the label of a collection request parcel. Default value is false.

info1	xs:string / 0 .. 1 / maxLength: 29	Information text 1. Will be printed on the parcel label if the flag printInfo1OnParcellabel is set to true.
info2	xs:string / 0 .. 1 / maxLength: 30	Information text 2. Will not be printed on the parcel label.
returns	xs:boolean / 0 .. 1 /	Defines if parcel is a return parcel. The return parcel must always be next in order to the corresponding outbound parcel. Default value is false.
customsTransportCost	xs:long / 0 .. 1 / maxInclusive: 99999999 999999	Custom transport cost without decimal separator (e.g. 14.000 = 14000). Mandatory for the destination country UK
customsTransportCostCurrency	xs:string / 0 .. 1 / length: 3	Currency code in ISO 4217 alpha-3 format. Mandatory for the destination country UK

Namespace: types

4.3.13 Complex Type: parcelInformationType

Field name	Data type / Occurrence / Range	Description
parcelLabelNumber	xs:string / 0 .. 1 /	The parcel label number of the corresponding parcel.
dpdReference	xs:string / 0 .. 1 /	The DPD reference for this parcel.

Namespace: types

4.3.14 Complex Type: parcelShopDelivery

Field name	Data type / Occurrence / Range	Description
parcelShopId	xs:long / 1 .. 1 /	Parcel shop ID for parcel shop delivery. Can be obtained from parcel shop finder. Assumes that parameter type of personal delivery is filled with a value from 3 to 5, which means parcel shop delivery. Mandatory in this case.
parcelShopNotification	types:notification / 1 .. 1 /	Contains data for personal notification for parcel shop consignments.

Namespace: types

4.3.15 Complex Type: personalDelivery

Description

Bundles personal delivery data.

Field name	Data type / Occurrence / Range	Description
type	xs:int / 1 .. 1 / minInclusive: 1 maxInclusive: 5	Defines type of personal delivery. Possible values are: = department delivery (without personal identification) = personal delivery with personal identification (IDCheck) = personal delivery without personal identification at drop point (e.g. parcel shop) = personal delivery with personal identification at drop point (e.g. parcel shop) = personal delivery with personal identification at drop point plus ID-Check (e.g. parcel shop) For parcel shop delivery the parcel shop id must be declared in productAndServiceData. It can be obtained from parcel shop finder.
floor	xs:string / 0 .. 1 / maxLength: 30	Floor where the personal delivery shall take place. This field is only used for department delivery.
building	xs:string / 0 .. 1 / maxLength: 30	Building where the personal delivery shall take place. This field is only used for department delivery (type 1).
department	xs:string / 0 .. 1 / maxLength: 30	Department where the personal delivery shall take place. This field is only used for department delivery (type 1).
name	xs:string / 0 .. 1 / maxLength: 35	Name of the person authorised to accept the consignment. This field is only used for delivery with ID-Check (types 2 and 4).
phone	xs:string / 0 .. 1 / maxLength: 20	Telephone number of the person authorised to accept the consignment. This field is only used for delivery with ID-Check (types 2 and 4). No required data format.
personId	xs:string / 0 .. 1 / maxLength: 35	Personal identification number of the person authorised to accept the consignment. This field is only used for delivery with ID-Check (types 2 and 4).

Namespace: types

4.3.16 Complex Type: pickup

Description

Bundles pickup data.

Field name	Data type / Occurrence / Range	Description
tour	xs:int / 0 .. 1 / maxInclusive: 999	Tour number (between 000 and 999).
quantity	xs:int / 1 .. 1 / maxInclusive: 99999	Quantity of pickup parcels. Mandatory for consignment.
date	xs:int / 1 .. 1 / maxInclusive: 99999999	Pickup date in consignments and collection requests. The collection day for advice customers. The from date for pickup information. Format is YYYYMMDD.
		It can also be used as pickup date for collection requests, then format is YYMMDD. Mandatory for consignment.
day	xs:int / 1 .. 1 / maxInclusive: 6	Pickup day of week for consignments and pickup information. Allowed values are 0..6 (0=Sunday, 1=Monday, etc.). Mandatory for consignment.
fromTime1	xs:int / 0 .. 1 / maxInclusive: 2400	From time 1 for consignments and pickup information. Format is hhmm. Mandatory for consignment.
toTime1	xs:int / 0 .. 1 / maxInclusive: 2400	Until time 1 for consignments and pickup information. Format is hhmm. Mandatory for consignment.
fromTime2	xs:int / 0 .. 1 / maxInclusive: 2400	From time 2 for consignments and pickup information. Format is hhmm.
toTime2	xs:int / 0 .. 1 / maxInclusive: 2400	Until time 2 for consignments and pickup information. Format is hhmm.
extraPickup	xs:boolean / 0 .. 1 /	True if extra pickup is requested.
collectionRequest Address	types:address / 0 .. 1 /	Contains pickup address information for consignments and collection requests. Mandatory for consignment.

Namespace: types

4.3.17 Complex Type: printOptions

Description:

Options how to return the parcel labels

Field name	Data type / Occurrence / Range	Description
printerLanguage	xs:string / 1 .. 1 / PDF ZPL	The language in which the parcel labels should be returned. PDF as file output, ZPL for direct printing. In any case the output is base64 encoded.
paperFormat	xs:string / 1 .. 1 / A4 A6 A7	Declares the paper format for parcel label print, either "A4" or "A6". For direct printing the format has to be set to "A6". "A7" only for return labels, other type are not allowed.
printer	types:printer / 0 .. 1 /	Information about the printer. Enabled if direct printing is selected.
startPosition	types:startPosition / 0 .. 1 /	Start position for print on A4 paper.

4.3.18 Complex Type: printer

Description:

Information about the printer, if direct printing is selected.

Field name	Data type / Occurrence / Range	Description
manufacturer	xs:string / 0 .. 1 /	The printer's manufacturer. At the moment only for informational purposes.
model	xs:string / 0 .. 1 /	The printer's model. At the moment only for informational purposes.
revision	xs:string / 0 .. 1 /	The printer's revision / version. At the moment only for informational purposes.
offsetX	xs:decimal / 1 .. 1 /	The horizontal offset in mm for the direct printer file (Not used with normal PDF output).
offsetY	xs:decimal / 1 .. 1 /	The vertical offset in mm for the direct printer file (Not used with normal PDF output).
connectionType	xs:string / 1 .. 1 / SERIAL PARALLEL	The connection type of the printer: serial or parallel connection.
barcodeCapable2D	xs:boolean / 1 .. 1 /	If the printer can print AZTEC barcodes, set this flag to true.

4.3.19 Complex Type: proactiveNotification

Field name	Data type / Occurrence / Range	Description
channel	xs:int / 1 .. 1 / 1 2 3 6 7	Defines type of proactive notification. Possible values are: = email = telephone = SMS = FAX = postcard
value	xs:string / 1 .. 1 / maxLength: 50	Data for proactive notification, e.g. telephone number, email address, etc. The data format for the proactive message types SMS, phone and FAX is as follows: +international country number#phone number Examples: +49#1725673423 +49#01725673423
rule	xs:int / 1 .. 1 / maxInclusive: 31	Rule for which events a notification is sent. Each event has a certain value. By adding the different values it is possible to build combinations of events, e.g. notification for pick-up and delivery is 5. The different values are: = pick-up = non-delivery 4 = delivery 8 = inbound 16 = out for delivery
language	xs:string / 1 .. 1 / length: 2	Language of the proactive notification in ISO-3166-1 alpha-2 format (e.g. 'DE').

4.3.20 Complex Type: productAndServiceData

Field name	Data type / Occurrence / Range	Description
orderType	xs:string / 1 .. 1 / consignment collection request order pickup information	Defines the shipment type. Possible values are: consignment collection request order pickup information
saturdayDelivery	xs:boolean / 0 .. 1 /	Defines if saturday delivery is demanded. Only selectable for product "E12". Default value is false.
exWorksDelivery	xs:boolean / 0 .. 1 /	Defines if the recipient has to pay the consignment. Default value is false.
guarantee	xs:boolean / 0 .. 1 /	Set to true if an international shipment shall use Guarantee, only for products NP and E18 in international parcels
tyres	xs:boolean / 0 .. 1 /	Set to true if this consignment contains bulk tyres, only for product NP.

personalDelivery	types:personalDelivery / 0 .. 1 /	Contains information for personal delivery.
pickup	types:pickup / 0 .. 1 /	Contains information for pickup consignments, pickup address for collection requests or details for pickup information.
parcelShopDelivery	types:parcelShopDelivery / 0 .. 1 /	Contains the necessary information, if a parcel shop delivery is intended.
predict	types:notification / 0 .. 1 /	Contains data for interactive notification for consignments. Only channel e-mail and SMS is allowed.
personalDelivery Notification	types:notification / 0 .. 1 /	Contains data for personal delivery notification for consignments.
proactiveNotification	types:proactiveNotification / 0 .. 5 /	Contains information of proactive notification for consignments.
delivery	types:delivery / 0 .. 1 /	Contains special delivery data for consignments.
invoiceAddress	types:address / 0 .. 1 /	Contains data about invoice recipient if it differs for consignment.
countrySpecificService	xs:string / 0 .. 1 /	In some relations a specific service code can be set which overwrites the feature selection.

Namespace: types

4.3.21 ComplexType: shipmentResponse

Description

Bundles shipment response data.

Field name	Data type / Occurrence / Range	Description
identificationNumber	xs:string / 0 .. 1 /	Serves as unique alphanumeric key of the shipment used by customer.
mpslId	xs:string / 0 .. 1 / maxLength: 25	The shipment number for consignment data. If ordertype is pickup information, the shipment number is an internal database id, which is necessary for technical support requests at DPD.
parcelInformation	types:parcelInformation Type / 0 .. * /	Contains information about the single parcels.
faults	types:faultCodeType / 0 .. 3 /	Contains information about errors during shipment order processing.

Namespace: types

4.3.22 Complex Type: shipmentServiceData

Description

Bundles shipment service data.

Field name	Data type / Occurrence / Range	Description
generalShipmentData	types:generalShipmentData / 1 .. 1 /	Contains general data for a shipment.
parcels	types:parcel / 0 .. * /	Contains information for the parcels.
productAndServiceData	types:productAndServiceData / 1 .. 1 /	Contains special data for a shipment.

Namespace: types

4.3.23 Complex Type: storeOrders

Field name	Data type / Occurrence / Range	Description
printOptions types:printOptions /	printOptions types:printOptions /	printOptions types:printOptions /
order	types:shipmentServiceData / .. 30 /	Contains the whole data for the shipments.

Namespace: types

4.3.24 Complex Type: storeOrdersResponse

Field name	Data type / Occurrence / Range	Description
orderResult	types:storeOrdersResponseType / 1 .. 1 /	Contains return object.

Namespace: types

4.3.25 Complex Type: storeOrdersResponseType

Field name	Data type / Occurrence / Range	Description
parcellabelsPDF	xs:base64Binary / 0 .. 1 /	Contains parcel label PDF data in bytes.
shipmentResponses	types:shipmentResponse / 0 .. * /	Contains response data for every shipment order.

4.3.26 Namespace: types_authentication

URL: <http://dpd.com/common/service/types/Authentication/2.0>

Namespace: types_authentication

4.3.27 Anonymous Complex Type of Element: authentication

Description

The data structure for authentication data.

Field name	Data type / Occurrence / Range	Description
delisId	xsd:string / 1 .. 1 / minLength: 8 maxLength: 10	The delis user id for authentication.
authToken	xsd:string / 1 .. 1 / minLength: 0 maxLength: 64	The token for authentication. Field authToken of Login, as a result of Method "getAuth" of LoginService.
messageLanguage	xsd:string / 1 .. 1 / minLength: 5 maxLength: 5	The language (Java format) for messages. "de_DE" for german messages. "en_US" for english messages.

Namespace: types_authentication

4.3.28 Anonymous Complex Type of Element: authenticationFault

Description

The data structure for authentication errors.

Field name	Data type / Occurrence / Range	Description
errorCode	xsd:string / 1 .. 1 /	The error code. Valid codes: NoAccessAuthenticationFault=-1 NoRightsAuthenticationFault=-2
errorMessage	xsd:string / 1 .. 1 / minLength: 1 maxLength: 255	The error message text.

4.4 Simple Types

4.4.1 Namespace: types

URL: <http://dpd.com/common/service/types/ShipmentService/3.3>

Namespace: types

4.4.2 Simple Type: startPosition

Field name	Data type / Occurrence / Range	Description
startPosition	xs:string / / UPPER_LEFT UPPER_RIGHT LOWER_LEFT LOWER_RIGHT	Start positions

4.5 Elements

4.5.1 Namespace: types

URL: <http://dpd.com/common/service/types/ShipmentService/3.3>

Namespace: types

4.5.2 Element: storeOrders

Field name	Data type / Occurrence / Range	Description
storeOrders	types:storeOrders / 1 .. 1 /	See types:storeOrders

Namespace: types

4.5.3 Element: storeOrdersResponse

Field name	Data type / Occurrence / Range	Description
storeOrdersResponse	types:storeOrders Response / .. 1 /	See types:storeOrdersResponse

- *Namespace: types_authentication*

URL: <http://dpd.com/common/service/types/Authentication/2.0>

Namespace: types_authentication

- *Element: authentication*

Anonymous Complex Type types_authentication:authentication

Namespace: types_authentication

- *Element: authenticationFault*

Anonymous Complex Type types_authentication:authenticationFault

Example Request Brexit

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://dpd.com/common/service/types/Authentication/2.0"
xmlns:ns1="http://dpd.com/common/service/types/ShipmentService/3.3">
  <soapenv:Header>
    <ns:authentication>
      <delisId><your delisId></delisId>
      <authToken><your authToken></authToken>
      <messageLanguage>de_DE</messageLanguage>
    </ns:authentication>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:storeOrders>
      <printOptions>
        <printerLanguage>PDF</printerLanguage>
        <paperFormat>A6</paperFormat>
      </printOptions>
      <order>
        <generalShipmentData>
          <sendingDepot>0522</sendingDepot>
          <product>CL</product>
          <sender>
            <name1>Abholer1</name1>
            <street>Strasse</street>
            <houseNo>1</houseNo>
            <country>NL</country>
            <zipCode>1234AB</zipCode>
            <city>Ort1</city>
            <type>P</type>
          </sender>
          <recipient>
            <name1>Test-Receiver</name1>
            <street>Test-Street</street>
            <country>GB</country>
            <zipCode>NW107XF</zipCode>
            <city>London</city>
            <type>P</type>
            <contact>John Doe</contact>
            <phone>+44 123 456 7890</phone>
          </recipient>
        </generalShipmentData>
        <parcels>
          <customerReferenceNumber1>CustRef1</customerReferenceNumber1>
          <international>
            <parcelType>1</parcelType>
            <customsAmount>1500</customsAmount>
            <customsCurrency>EUR</customsCurrency>
            <customsAmountEx>1500</customsAmountEx>
            <customsCurrencyEx>EUR</customsCurrencyEx>
            <clearanceCleared>N</clearanceCleared>
            <prealertStatus>S03</prealertStatus>
            <exportReason>01</exportReason>
            <customsTerms>03</customsTerms>
            <customsContent>Content</customsContent>
            <customsPaper>A</customsPaper>
            <customsEnclosure>1</customsEnclosure>
            <customsInvoice>12345</customsInvoice>
            <customsInvoiceDate>20190327</customsInvoiceDate>
            <commercialInvoiceConsignee>
              <name1>Test-Receiver</name1>
              <street>Test-Street</street>
            </commercialInvoiceConsignee>
          </international>
        </parcels>
      </order>
    </ns1:storeOrders>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<country>GB</country>
<zipCode>NW107XF</zipCode>
<city>London</city>
<type>P</type>
<contact>John Doe</contact>
<phone>+44 123 456 7890</phone>
</commercialInvoiceConsignee>
<commercialInvoiceConsignor>
  <name1>Abholer1</name1>
  <street>Strasse</street>
  <houseNo>1</houseNo>
  <country>NL</country>
  <zipCode>1234AB</zipCode>
  <city>Ort1</city>
  <type>P</type>
  <contact>contactperson</contact>
  <phone>0123456789</phone>
</commercialInvoiceConsignor>
</international>
</parcels>
<productAndServiceData>
  <orderType>consignment</orderType>
</productAndServiceData>
</order>
</ns1:storeOrders>
</soapenv:Body>
</soapenv:Envelope>
```