



---

# **ParcelShopFinderService-Public\_5\_0** **Web Service Reference**

**Version 5.0**

19.03.2015

# Table of contents

<b>History</b> .....	<b>5</b>
<b>Methods</b> .....	<b>6</b>
findCities .....	7
findParcelShops .....	10
findParcelShopsByGeoData .....	14
getAvailableServices .....	18
<b>Complex Types</b> .....	<b>21</b>
<b>types</b> .....	<b>21</b>
CityType .....	21
DataFaultType .....	22
FaultType .....	23
FindCitiesResponseType .....	24
FindCitiesType .....	24
FindParcelShopsByGeoDataResponseType .....	25
FindParcelShopsByGeoDataType .....	26
FindParcelShopsResponseType .....	27
FindParcelShopsType .....	27
GetAvailableServicesResponseType .....	29
GetAvailableServicesType .....	29
HolidayType .....	30

OpeningHoursType .....	30
ParcelShopType .....	31
ServiceDetailRequestType .....	34
ServiceDetailType .....	34
ServiceRequestType .....	35
ServiceType .....	35
ServicesRequestType .....	36
ServicesType .....	37
SystemFaultType .....	37
<b>types_authentication .....</b>	<b>39</b>
<b>anonymous .....</b>	<b>39</b>
authentication .....	39
authenticationFault .....	40
<b>Simple Types .....</b>	<b>41</b>
<b>types .....</b>	<b>41</b>
CodeType .....	41
CountryType .....	41
DescriptionType .....	42
MessageLanguageType .....	42
<b>Elements .....</b>	<b>43</b>
<b>types .....</b>	<b>43</b>

dataFault .....	43
findCities .....	43
findCitiesResponse .....	44
findParcelShops .....	44
findParcelShopsByGeoData .....	44
findParcelShopsByGeoDataResponse .....	45
findParcelShopsResponse .....	45
getAvailableServices .....	46
getAvailableServicesResponse .....	46
systemFault .....	46
<b>types_authentication .....</b>	<b>47</b>
authentication .....	47
authenticationFault .....	47

## History

### Version 5.0

- Integration of pudo ID.
- Integration of new services structure to support new services:
  - Shop-Ident
  - Dropoff Offline
- Expansion of cross border search.

## Methods

Following methods are available for the service **ParcelShopFinderService-Public\_5\_0**

<i>Method name</i>	<i>Description</i>
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.

## 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:

<i>Name</i>	<i>Element</i>
authentication *	types_authentication:authentication
parameters	types:findCities

\* header parameter

### Output

The output for method findCities is as follows:

<i>Name</i>	<i>Element</i>
parameters	types:findCitiesResponse

### Faults

The defined faults for method findCities are as follows:

<i>Name</i>	<i>Element</i>
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

## 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:

```
{  
  "findParcelShopsResponse": {"parcelShop": [ {  
    "parcelShopId": 272195,  
    "puId": "DE16935",  
    "customerNo": "01600000000238770",  
    "companyDepotDatabaseId": 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:

<i>Name</i>	<i>Element</i>
authentication *	types_authentication:authentication
parameters	types:findParcelShops

\* header parameter

### Output

The output for method findParcelShops is as follows:

<i>Name</i>	<i>Element</i>
parameters	types:findParcelShopsResponse

### Faults

The defined faults for method findParcelShops are as follows:

<i>Name</i>	<i>Element</i>
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

## 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",
    "companyDepotDatabaseId": 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:

<i>Name</i>	<i>Element</i>
authentication *	types_authentication:authentication
parameters	types:findParcelShopsByGeoData

\* header parameter

### Output

The output for method findParcelShopsByGeoData is as follows:

<i>Name</i>	<i>Element</i>
parameters	types:findParcelShopsByGeoDataResponse

### Faults

The defined faults for method findParcelShopsByGeoData are as follows:

<i>Name</i>	<i>Element</i>
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault



<i>Name</i>	<i>Element</i>
dataFault	types:dataFault

## 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:

<i>Name</i>	<i>Element</i>
authentication *	types_authentication:authentication
parameters	types:getAvailableServices

\* header parameter

### Output

The output for method getAvailableServices is as follows:

<i>Name</i>	<i>Element</i>
parameters	types:getAvailableServicesResponse

### Faults

The defined faults for method getAvailableServices are as follows:

<i>Name</i>	<i>Element</i>
authenticationFault	types_authentication:authenticationFault
systemFault	types:systemFault
dataFault	types:dataFault

## Complex Types

### Namespace: types

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

Namespace: types

### Complex Type: CityType

#### Description

The city type with city, zip code and town.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

Namespace: types

## Complex Type: DataFaultType

### Description

Is thrown if a data fault occurs.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
systemShortMessage	xsd:string / 0..1 /	The short info text for the error in system language.

Namespace: types

### **Complex Type: FaultType**

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

Namespace: types

### Complex Type: FindCitiesResponseType

#### Description

The result data for method findCities.

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

Namespace: types

### Complex Type: FindCitiesType

#### Description

The search parameters for method findCities.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

Namespace: types

### Complex Type: FindParcelShopsByGeoDataResponseType

#### Description

The parcel shops as result of the search.

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

Namespace: types

## Complex Type: FindParcelShopsByGeoDataType

### Description

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

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.
----------	---	--

Namespace: types

### Complex Type: FindParcelShopsResponseType

#### Description

The parcel shops as result of the search.

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

Namespace: types

### Complex Type: FindParcelShopsType

#### Description

The request parameters for the search.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

Namespace: types

**Complex Type: GetAvailableServicesResponseType**

**Description**

The result data for method getAvailableServices.

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

Namespace: types

**Complex Type: GetAvailableServicesType**

**Description**

The search parameters for method getAvailableServices.

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

Namespace: types

### Complex Type: HolidayType

#### Description

A Data container for holiday.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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".

Namespace: types

### Complex Type: OpeningHoursType

#### Description

The data structure for "opening hour" data.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

Namespace: types

### Complex Type: ParcelShopType

#### Description

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

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
parcelShopId	xsd:long / 0 .. 1 / minInclusive: 1 maxInclusive: 999999999 999999999	The ID of the parcel shop.
pudold	xsd:string / 0 .. 1 / maxLength: 7	The pudo 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.
extraInfo	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.

Namespace: types

### Complex Type: ServiceDetailRequestType

#### Description

The search parameters for parcel shop service details.

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

Namespace: types

### Complex Type: ServiceDetailType

#### Description

A data container for parcel shop service details.

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

Namespace: types

### Complex Type: ServiceRequestType

#### Description

The search parameters for parcel shop services with service details.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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:ServiceDetail RequestType / 0 .. * /	A list of service detail information.

Namespace: types

### Complex Type: ServiceType

#### Description

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

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

Namespace: types

### Complex Type: ServicesRequestType

#### Description

Contains the search parameters for parcel shop services.

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

Namespace: types

### Complex Type: ServiceType

#### Description

Contains the result data for supported parcel shop services.

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

Namespace: types

### Complex Type: SystemFaultType

#### Description

Is thrown if an internal system fault occurs.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.

**Namespace: types\_authentication**

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

Namespace: types\_authentication

**Anonymous Complex Type of Element: authentication**

**Description**

The data structure for authentication data.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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

### Anonymous Complex Type of Element: authenticationFault

#### Description

The data structure for authentication errors.

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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.



## Simple Types

### Namespace: types

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

Namespace: types

### Simple Type: CodeType

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

Namespace: types

### Simple Type: CountryType

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
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).

Namespace: types

**Simple Type: DescriptionType**

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

Namespace: types

**Simple Type: MessageLanguageType**

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

## Elements

### Namespace: types

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

Namespace: types

#### Element: dataFault

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

Namespace: types

#### Element: findCities

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

Namespace: types

**Element: findCitiesResponse**

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

Namespace: types

**Element: findParcelShops**

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

Namespace: types

**Element: findParcelShopsByGeoData**

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

Namespace: types

**Element: findParcelShopsByGeoDataResponse**

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
findParcelShopsByGeoDataResponse	types:FindParcelShopsByGeoDataResponse Type / 1 .. 1 /	Output parameters of findParcelShopsByGeoDataResponse.

Namespace: types

**Element: findParcelShopsResponse**

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
findParcelShopsResponse	types:FindParcelShopsResponseType / 1 .. 1 /	Output parameters of findParcelShops.

Namespace: types

**Element: getAvailableServices**

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

Namespace: types

**Element: getAvailableServicesResponse**

<i>Field name</i>	<i>Data type / Occurrence / Range</i>	<i>Description</i>
getAvailableServicesResponse	types:GetAvailableServicesResponseType / 1 .. 1 /	Output parameters of getAvailableServicesResponse.

Namespace: types

**Element: systemFault**

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

**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*