

# Fleet API (1.24.0)

---

## Overview

---

This document defines the interface for managing terminal networks, terminals and authorization of fleet card transactions on terminals. The interface is implemented as a REST webservice with data transferred as JSON. PDF Version: [fleetAPI.pdf](#)

## Change history

Date	Version	Changed by	Description of changes
2017-09-04	1.0.3	Paweł Gamrot, Seweryn Krajewski, Łukasz Guzik	The initial version of the documentation
2017-09-19	1.0.4	Paweł Gamrot	Correction of the url access correction of names in Partners object
2017-09-28	1.0.5	Paweł Gamrot	Correction of allowed values for fields amount/quantity correction of sample (track2Data should be in hexstr)
2017-10-03	1.0.6	Paweł Gamrot, Łukasz Guzik	Format change from int32 to int64 for amount fields, Correction of maturity of fields in EodItem correction of EodTransaction.amount range
2017-10-12	1.0.7	Łukasz Guzik	Changed description of 901 response code - Advice acknowledged
2017-11-22	1.0.8	Łukasz Guzik	Partner model change, update of regular expression for authorisation code and plate number format control
2017-12-04	1.0.9	Łukasz Guzik	Change of data definition model of parameters and mandatory parameter EOD_FILE_NAME_FORMAT for DKV
2018-01-18	1.1.0	Łukasz Guzik	Customer model change. Added a new model of client's internal parameters
2018-02-07	1.2.0	Łukasz Guzik	Receipt id field added to a manual transaction endpoint and EOD operation endpoint
2018-02-28	1.3.0	Łukasz Guzik	Change of registration number format
2018-04-13	1.4.0	Maciej Wiosło	Manual transaction model added
2018-05-29	1.5.0	Maciej Wiosło	New global parameters for partners and stations independent to fleets, searching of partner by NIP
2018-08-21	1.6.0	Maciej Wiosło	System size control parameters added
2018-09-17	1.7.0	Maciej Wiosło	Product mapping method added
2018-10-23	1.7.1	Maciej Wiosło	FIX - removing parameters of removed customer
2018-11-26	1.7.2	Maciej Wiosło, Marcin Kisielewski	Acceptation of 10 digit authorisation code in DKV
2019-02-04	1.8.0	Maciej Wiosło, Ireneusz Olchawski	Product type flag in product list added Terminal authorisation added in endpoints v1/fleets and v1/terminals TIMEZONE implementation in transaction date and time Terminal authorisation added in endpoint /v1/products Global parameters added: MAX_BATCH_TRANSACTIONS, DEFAULT_RESPONSE_LANG Global parameter management endpoint added Group level added Station list endpoint change

Date	Version	Changed by	Description of changes
2019-03-26	1.9.0	Maciej Wiosło	Validation of EOD size against MAX_BATCH_TRANSACTIONS parameter
2019-04-01	1.9.1	Maciej Wiosło	Parameters getting fix
2019-06-03	1.9.2	Maciej Wiosło, Seweryn Krajewski	Translation of yaml
2019-07-04	1.9.3	Maciej Wiosło	Terminal edit fix
2019-10-16	1.10.0	Maciej Wiosło	EUROWAG fleet added
2020-01-08	1.10.1	Maciej Wiosło	FIX - checking if the terminal sends EOD transaction that already exist
2019-11-12	1.11.0	Paweł Gamrot	LOTOS and E100 fleets added
2020-01-31	1.12.0	Paweł Gamrot	Add countryCode parameter to /v1/products endpoint Add balance transaction (6-BL)
2020-04-03	1.12.1	Tomasz Michno	Removing endpoints for user modification
2020-04-14	1.12.2	Maciej Wiosło	AA response code fix - for AA amount higher than PA
2020-04-28	1.12.3	Maciej Wiosło, Tomasz Michno	Receipt id field pattern and max length 8 to 10 changed, added sorting for products endpoint
2020-06-09	1.13.0	Maciej Wiosło	Add complaintId field to /v1/stations/{stationId}/transactions endpoint
2021-01-12	1.14.0	Paweł Gamrot	Add JWT Token Security Add new endpoint /v1/terminals/{terminalId}/fleets/{fleetName}/transactions Add new endpoint for EOD where source is file
2021-01-19	1.14.1	Paweł Gamrot	Fix endpoint to get eodfile status to get eodfile status by fileName
2021-02-03	1.14.2	Paweł Gamrot	Add relatedPartnerId field to partner object
2021-02-09	1.14.3	Paweł Gamrot	Add default to fileLocalization
2021-02-09	1.14.3	Ireneusz Olchawski	EodFiles endpoint changed
2021-03-02	1.14.3	Ireneusz Olchawski	EodFiles endpoint status changed
2021-03-15	1.15.0	Paweł Gamrot	Add EKOTANK fleet
2021-03-18	1.15.1	Paweł Gamrot	Products list mandatory for PA for EKOTANK

Date	Version	Changed by	Description of changes
2021-03-31	1.15.2	Maciej Wiosło	Correction of odometer maximum value
2021-04-19	1.15.3	Maciej Wiosło	Add fleetSpecificData field to transaction object
2021-05-28	1.16.0	Maciej Wiosło	Add AVIA fleet
2021-06-30	1.17.0	Mateusz Sawczuk	Add DCI fleet, Updated FleetCardsRange Model
2021-11-22	1.18.0	Ireneusz Olchawski	Added fields to partner and station
2021-11-25	1.19.0	Mateusz Sawczuk	Added DKV restriction code to product model
2022-01-04	1.19.1	Ireneusz Olchawski	Added condition when not to send transaction to queue
2022-01-05	1.19.2	Ireneusz Olchawski	Added detailed description on auth error
2022-01-13	1.19.3	Ireneusz Olchawski	Changed date taken in monitoring controller
2022-02-10	1.20.0	Mateusz Sawczuk	Added handling for EMV transaction, created new terminal type, added CVC code to CardData
2022-03-11	1.21.0	Paweł Gamrot	Fix inline data models Remove format binary for track2 and pinData add x-enum-varnames for integer enums
2022-05-10	1.22.0	Maciej Wiosło	Add IQCARD fleet Add example of UTA transaction without CVC for terminal www Two step authorisation for indoor terminal enabled
2022-09-21	1.23.0	Seweryn Krajewski	Tokenization for AVIA card added
2022-10-21	1.24.0	Seweryn Krajewski	New response code 923 AdditionalTrack in customerData New fleets: MORGAN, DIESEL24, HOYER

## Fleets

Fleets supported by FleetAPI :

- UTA
- DKV
- EUROWAG
- E100
- LOTOS
- EKOTANK
- AVIA
- DCI
- IQCARD
- MORGAN

- DIESEL24
- HOYER

## **Fleet Parameters**

The table below presents parameters for fleets:

Fleet	Name	Description
UTA	ACTIVE	Fleet activity
UTA	CARD_ACCEPTOR_ID	Station ID (card acceptor ID, ISO8583-42)
UTA	SUPPLIER_NAME	Partner (supplier) name in UTA system (used in settlement file)
UTA	SUPPLIER_ID	Partner (supplier) ID in UTA system (used in settlement file)
DKV	ACTIVE	Fleet activity
DKV	CARD_ACCEPTOR_ID	Station ID (card acceptor ID, ISO8583-42)
DKV	SUPPLIER_NAME	Partner (supplier) name in DKV system (used in settlement file)
DKV	SUPPLIER_ID	Partner (supplier) ID in DKV system (used in settlement file)
DKV	EOD_FILE_NAME_FORMAT	Settlement file name format
EUROWAG	ACTIVE	Fleet activity
EUROWAG	CARD_ACCEPTOR_ID	Station ID (card acceptor ID, ISO8583-42)
EUROWAG	SUPPLIER_NAME	Partner (supplier) name in EUROWAG system (used in settlement file)
EUROWAG	SUPPLIER_ID	Partner (supplier) ID in EUROWAG system (used in settlement file)
E100	ACTIVE	Fleet activity
E100	CARD_ACCEPTOR_ID	Station ID (card acceptor ID, ISO8583-42)
E100	SUPPLIER_NAME	Partner (supplier) name in E100 system (used in settlement file)
E100	SUPPLIER_ID	Partner (supplier) ID in E100 system (used in settlement file)
LOTOS	ACTIVE	Fleet activity
LOTOS	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
LOTOS	SUPPLIER_NAME	Partner (supplier) name in LOTOS system (used in settlement file)
LOTOS	SUPPLIER_ID	Partner (supplier) ID in LOTOS system (used in settlement file)
EKOTANK	ACTIVE	Fleet activity
EKOTANK	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)

<b>Fleet</b>	<b>Name</b>	<b>Description</b>
EKOTANK	SUPPLIER_NAME	Partner (supplier) name in EKOTANK system (used in settlement file)
EKOTANK	SUPPLIER_ID	Partner (supplier) ID in EKOTANK system (used in settlement file)
AVIA	ACTIVE	Fleet activity
AVIA	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
AVIA	SUPPLIER_NAME	Partner (supplier) name in AVIA system (used in settlement file)
AVIA	SUPPLIER_ID	Partner (supplier) ID in AVIA system (used in settlement file)
DCI	ACTIVE	Fleet activity
DCI	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
DCI	SUPPLIER_NAME	Partner (supplier) name in DCI system (used in settlement file)
DCI	SUPPLIER_ID	Partner (supplier) ID in DCI system (used in settlement file)
IQCARD	ACTIVE	Fleet activity
IQCARD	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
IQCARD	SUPPLIER_NAME	Partner (supplier) name in IQCARD system (used in settlement file)
IQCARD	SUPPLIER_ID	Partner (supplier) ID in IQCARD system (used in settlement file)
MORGAN	ACTIVE	Fleet activity
MORGAN	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
MORGAN	SUPPLIER_NAME	Partner (supplier) name in MORGAN system (used in settlement file)
MORGAN	SUPPLIER_ID	Partner (supplier) ID in MORGAN system (used in settlement file)
DIESEL24	ACTIVE	Fleet activity
DIESEL24	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
DIESEL24	SUPPLIER_NAME	Partner (supplier) name in DIESEL24 system (used in settlement file)

Fleet	Name	Description
DIESEL24	SUPPLIER_ID	Partner (supplier) ID in DIESEL24 system (used in settlement file)
HOYER	ACTIVE	Fleet activity
HOYER	CARD_ACCEPTOR_ID	Station/Toll point ID (card acceptor ID, ISO8583-42)
HOYER	SUPPLIER_NAME	Partner (supplier) name in HOYER system (used in settlement file)
HOYER	SUPPLIER_ID	Partner (supplier) ID in HOYER system (used in settlement file)

## Global parameters of partner, station and terminal

The table below presents global parameters for partner, station and terminal:

Name	Description	Default value	Allowed values
TIMEZONE	Partner timezone	Europe/Warsaw	Europe/Warsaw Europe/Budapest Europe/Stockholm Europe/Dublin Europe/London Europe/Amsterdam Europe/Brussels Europe/Luxembourg Europe/Berlin Europe/Paris Europe/Vienna Europe/Copenhagen Europe/Madrid Europe/Ljubljana Europe/Bratislava Europe/Rome Europe/Prague Europe/Lisbon Europe/Vilnius Europe/Tallinn Europe/Belfast
MAX_BATCH_TRANSACTIONS	Maximum count of transaction in batch	100	
DEFAULT_RESPONSE_LANG	Default response language for partner	pl	pl en hu



## Products

Unified products are defined in the system (translation into products appropriate for a given fleet will take place inside the system). The list of supported products is provided by calling `GET /products`.

### Units measures of product

The following units of measurement are available:

Mark	Description
L	Liters
W	Weight (in kilograms)
Q	Quantity

## Authorization

### Levels

The user is assigned to one of the levels:

- group
- client
- partner
- station

### Access rights

Each user can have a combination of access rights on the specified level:

- create (1)
- read (2)
- write (4)
- delete (8)

### REST method access rights

Each of the access rights corresponds to the REST methods:

- create - POST
- read - GET
- write - PUT
- delete - DELETE

### Others

- The user can add a new user only to the lower level
- Each added user gets the `read` right for higher levels (in his branch), and the same rights on the same and lower levels as the adding user.
- The user can only change his password
- The user can edit (deactivate, change permissions) users only at lower levels as long as it has `write` permissions on the lower level
- Each user has the `read` permission to URI `/fleets` and `/products`

- User at the level `group`, `client` and `partner` (having the right of access `create` on these levels) can add the manual transactions registration right

## Examples

### Login using JWT

Request data:

```
{
  "username": "userName",
  "password": "strongPassword"
}
```

Response data:

```
{
  "authorizationToken": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
    "expires": "2021-01-12T08:55:28.375Z"
  },
  "refreshToken": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ8",
    "expires": "2021-01-12T08:55:28.375Z"
  },
  "loggedUserContext": {
    "userId": 1,
    "login": "userName"
  }
}
```

## Transactions

For the servicing of fleet cards, the API will support the following operations:

id	ids	name	description
1	HS	handshake	<b>Handshake operation</b> - The operation tests the availability of fleet servers
2	SO	sale online	<b>Sale online operation</b> - The operation is used to authorize sales in indoor terminals
3	PA	preauthorization	<b>Preauthorization operation</b> - The pre-authorization operation is used to block funds on the card holder's account for refueling with an outdoor terminal (optionally allowed with an indoor terminal)
4	AA	authorization advice	<b>Financial advice operation</b> - The completion operation finalizes previously approved pre-authorization by transferring the final sales amount to the server. Completion amount cannot be higher than 101% of amount approved in the pre-authorization operation. In case of offline in indoor terminal, this operation is used to report offline operations authorized via the telephone line. In this case, an authorization code obtained in the IVR system is required
5	RV	reversal	<b>Cancellation</b> - The cancellation operation is performed by the terminal in order to send information to the card authorization system about improperly completed financial operation (preauthorization / online sale) or manual cancellation of the transaction at the service station
6	BL	balance	<b>Card balance</b> - The balance operation return maximum funds and the valid products list for card
7	PC	pin change	<b>Pin change</b> - The operation change users card Pin, only available to fleets that support it (EKOTANK)
8	TC	tokenization	<b>Tokenization of card</b> - Operation generates token for card (available for avia card only)

## Retransmissions

In case of absence of a response to the query (eg. timeout), the terminal side is obliged to repeat certain operations (reversal, authorization advice) until acceptance by FleetAPI. In the case of retransmission, the field `retry` is set to 1 and the value in `transmissionDateTime` is updated, the rest of the fields are not changed.

## Request data

Required fields (M-required (mandatory), O-not required (optional), C-dependent (conditional)) of the `Transaction` object depending on the transaction type (`TransactionType`) for URI `/terminals/fleets/{fleetName}/transactions`

If the field in the table below is not listed, it means that it is a field received or for all types of transactions the maturity of this field is `O`.

Transaction type Field	1 HS	2 SO	3 PA	4 AA	5 RV	6 BL	7 PC	8 TC	Description
transactionId	M	M	M	M	M	M	M	M	
batchId	O	O	O	O	O	O	O	O	Mandatory for endpoint /terminals/fleets/{fleetName}/transactions
type	M	M	M	M	M	M	M	M	
retry	M	M	M	M	M	M	M	M	For operation <b>1, 2, 3, 6</b> and <b>8</b> always 0
fleet	M	M	M	M	M	M	M	M	
transactionDateTime	O	M	M	M	M	M	M	M	
transmissionDateTime	M	M	M	M	M	M	M	M	
amount	O	M	M	M	M	O	O	O	
cardData	O	M	M	M	M	M	M	M	
currencyCode	O	M	M	M	M	M	O	O	
pinData	O	C	M	O	O	M	M	M	For <b>2</b> not mandatory for toll point (MCC = 4784)
authorizationCode	O	O	O	M	O	O	O	O	For operation <b>4</b> in case of voice authorisation this field must be set to a value gain in phone authorisation system
customerData	O	M	M	M	O	M	O	O	
products	O	M	C	M	O	O	O	O	For <b>3</b> and for fleet <b>EKOTANK</b> field is mandatory
originalTransaction	O	O	O	C	M	O	O	O	Mandatory for operation <b>4</b> in case of preauthorisation completion
reversalReason	O	O	O	O	M	O	O	O	
gateNumber	O	C	O	O	O	O	O	O	For <b>2</b> mandatory for toll point (MCC = 4784)
newPinData	O	O	O	O	O	O	M	O	

## Response data

Fields obtained in response to the transaction (E-echo from the data sent, O-not required (optional), C-dependent (conditional)):

Transaction type Field	1 HS	2 SO	3 PA	4 AA	5 RV	6 BL	7 PC	8 TC	Description
transactionId	E	E	E	E	E	E	E	E	
transactionDateTime	O	E	E	E	E	E	E	E	
transmissionDateTime	M	M	M	M	M	M	M	M	
response	M	M	M	M	M	M	M	M	
authorizationCode	O	M	M	E	O	M	O	M	
allowedProducts	O	C	C	O	O	M	O	O	Field mandatory in response to <b>2</b> if not allowed product is present on product list in request. Mandatory in response to <b>3</b> with a list of allowed products (empty list means there are no restrictions for product code and all products defined in product table can be purchased)
amount	O	O	M	O	O	C	O	O	Present in <b>3</b> (available preauthorisation amount) in <b>6</b> field is available (card limit) if card has limit
originalAmount	O	O	M	O	O	O	O	O	Original amount of preauthorisation (from request)
requiredCustomerData	O	C	C	O	O	O	O	M	Field is present if in request <code>Transaction.customerData</code> are missing some required fields or in response to tokenization
fleetSpecificData	O	O	O	O	O	O	O	O	This field is present if fleet service returns specific data (detailed description in the <b>Fleet specific data</b> section)
newCardToken	O	O	O	O	O	O	O	M	This field is present for tokenization only

## Fleet specific data

Available fleet specific data for fleets:

- UTA - no specific data
- DKV - no specific data
- EUROWAG - no specific data
- E100 - no specific data
- LOTOS - no specific data
- EKOTANK - specific data:
  - AddData
  - AddAmount
  - Contractor
  - Limits.DayFuelQuan
  - Limits.DayOtherAmount
  - Limits.DayVisit
  - Limits.Data
  - Limits.MonthAllAmount
  - Limits.MonthFuelQuan

- Limits.MonthOtherAmount
- AVIA - no specific data
- DCI - no specific data
- IQCARD - no specific data

## Sample values of queries and responses

### 1 HS Handshake

#### Example 1

Approved handshake Request data:

```
{
  "transactionId": 12,
  "batchId": 2,
  "type": 1,
  "retry": 0,
  "fleet": "UTA",
  "transmissionDateTime": "2017-04-12T23:20:50",
}
```

Response data:

```
{
  "transactionId": 12,
  "transmissionDateTime": "2017-04-12T23:20:55",
  "response": {
    "code": 800
  }
}
```

#### Example 2

Required data is missing in the data sent Request data:

```
{
  "type": 1,
  "retry": 0
}
```

Response data:

```
{
  "code": 3,
  "message": "Błędna wartość body"
}
```

### 2 SO Sale online

## Example 1

Request data:

```
{
  "transactionId": 13,
  "batchId": 5,
  "type": 2,
  "fleet": "UTA",
  "retry": 0,
  "transactionDateTime": "2017-04-13T15:10:02",
  "transmissionDateTime": "2017-04-13T15:10:03",
  "amount": 10000,
  "cardData": {
    "track2Data":
"36333632303034353637383933313635393D31383039303030303234393030303030303030",
    "pan": "63620045678931659",
  },
  "currencyCode": "985",
  "pinData": {
    "pinBlock": "45117B1D1628FD8F",
    "pinBlockKey": "667C2E5B6C7314315B1F451E2E6E5B52"
  },
  "customerData": {
    "odometer": 12000,
    "plateNumber": "XXX1234",
  },
  "products": [
    {
      "productId": 13,
      "quantity": 1200,
      "price": 450,
      "amount": 5400,
      "internalProduct": {
        "name": "Benzyna Pb98",
        "productId": "12332"
      }
    }
  ]
}
```

Response data:

```
{
  "transactionId": 13,
  "transactionDateTime": "2017-04-13T15:10:02",
  "transmissionDateTime": "2017-04-13T15:10:03",
  "response": {
    "code": 0,
    "message": "Akceptacja"
  },
  "authorizationCode": "AQW123"
}
```

## Example 2

Request data for UTA transaction with cvc:

```
{
  "transactionId": 23,
  "batchId": 1,
  "type": 2,
  "fleet": "UTA",
  "retry": 0,
  "transactionDateTime": "2022-03-10T10:31:13.386+01:00",
  "transmissionDateTime": "2022-03-10T10:31:14.3252837+01:00",
  "amount": 400,
  "cardData": {
    "pan": "63620045678931659",
    "expirationDate": "1122",
    "cvc": "092"
  },
  "currencyCode": "985",
  "customerData": {
    "odometer": 12888
  },
  "products": [
    {
      "productId": 88,
      "quantity": 100,
      "price": 400,
      "amount": 400,
      "internalProduct": {
        "name": "Parking fee",
        "productId": "05410"
      }
    }
  ]
}
```

Response data:

```
{
  "transactionId": 82,
  "batchId": 1,
  "type": 2,
  "retry": 0,
  "fleet": "UTA",
  "transactionDateTime": "2022-03-10T10:31:13.386+01:00",
  "transmissionDateTime": "2022-03-10T10:31:14.3252837+01:00",
  "amount": 400,
  "cardData": {
    "pan": "63620045678931659",
    "expirationDate": "1122",
    "entryMode": 3,
    "cvc": "092"
  },
  "currencyCode": "985",
  "response": {
    "code": 0,
    "message": "Zaakceptowano"
  }
}
```



```

},
"authorizationCode": "307218",
"customerData": {
  "odometer": 12888
},
"products": [
  {
    "productId": 88,
    "quantity": 100,
    "price": 400,
    "amount": 400,
    "internalProduct": {
      "name": "Parking fee",
      "productId": "05410"
    }
  }
]
}

```

### Example 3

Request data for UTA transaction without cvc:

```

{
  "transactionId": 3274,
  "batchId": 1,
  "type": 2,
  "fleet": "UTA",
  "retry": 0,
  "transactionDateTime": "2022-04-15T10:03:11.569+02:00",
  "transmissionDateTime": "2022-04-15T10:03:11.569+02:00",
  "amount": 400,
  "cardData": {
    "pan": "70600045678940000",
    "expirationDate": "1122",
    "entryMode": 3
  },
  "currencyCode": "985",
  "customerData": {
    "odometer": 12888
  },
  "products": [
    {
      "productId": 88,
      "quantity": 100,
      "price": 400,
      "amount": 400,
      "internalProduct": {
        "name": "Parking fee",
        "productId": "05410"
      }
    }
  ]
}

```

Response data:



```

    },
    "products": [
      {
        "productId": 88,
        "quantity": 100,
        "price": 400,
        "amount": 400,
        "internalProduct": {
          "name": "Parking fee",
          "productId": "05410"
        }
      }
    ]
  }
}

```

Response data:

```

{
  "transactionId": 3273,
  "type": 2,
  "retry": 0,
  "fleet": "AVIA",
  "transactionDateTime": "2022-04-15T10:03:11.569+02:00",
  "transmissionDateTime": "2022-04-15T10:04:12.1462904+02:00",
  "amount": 400,
  "currencyCode": "985",
  "response": {
    "code": 0,
    "message": "Zaakceptowano"
  },
  "authorizationCode": "377923"
}

```

### Example 5

Request data for HOYER transaction with two-system card:

```

{
  "transactionId": 3275,
  "type": 2,
  "fleet": "HOYER",
  "retry": 0,
  "transactionDateTime": "2022-09-15T10:03:11.569+02:00",
  "transmissionDateTime": "2022-09-15T10:03:21.569+02:00",
  "amount": 400,
  "cardData": {
    "track2Data":
      "373030353530343536373839333136353939393D31383039303030303234393030303030303030"
  },
  "currencyCode": "985",
  "customerData": {
    {
      "odometer": 12888,
      "additionalTrack":
        "373030353530343536373839333136353939303D31383039303030303234393030303030303030"
    },
    "products": [

```

```

    {
      "productId": 88,
      "quantity": 100,
      "price": 400,
      "amount": 400,
      "internalProduct": {
        "name": "Parking fee",
        "productId": "05410"
      }
    }
  ]
}

```

Response data:

```

{
  "transactionId": 3273,
  "type": 2,
  "retry": 0,
  "fleet": "HOYER",
  "transactionDateTime": "2022-04-15T10:03:11.569+02:00",
  "transmissionDateTime": "2022-04-15T10:04:12.1462904+02:00",
  "amount": 400,
  "currencyCode": "985",
  "response": {
    "code": 0,
    "message": "Zaakceptowano"
  },
  "authorizationCode": "377923"
}

```

### 3 PA Preauthorization

#### Example 1

Approved preauthorization Request data:

```

{
  "transactionId": 3565,
  "batchId": 4313,
  "type": 3,
  "fleet": "DKV",
  "retry": 0,
  "transactionDateTime": "2022-05-10T07:57:33",
  "transmissionDateTime": "2022-05-10T07:57:33",
  "amount": 25600,
  "cardData": {
    "track2Data":
"3730343331303031313233343536373839D3232313130303930303030303030303030",
    "pan": "70431001123456789",
    "expirationDate": "1122"
  },
  "currencyCode": "985",
  "pinData": {
    "pinBlock": "C9DDBC1234ABCDE",
    "pinBlockKey": "7A4651234ABCDE1234ABCDE1234ABCDE"
  },
}

```

```

"customerData": {
  "odometer": 128512
}
}

```

Response data:

```

{
  "transactionId": 3565,
  "batchId": 4313,
  "type": 3,
  "retry": 0,
  "fleet": "DKV",
  "transactionDateTime": "2022-05-10T09:57:33.542+02:00",
  "transmissionDateTime": "2022-05-10T09:56:30.0381169+02:00",
  "amount": 25600,
  "cardData": {
    "track2Data":
"3730343331303031313233343536373839D3232313130303930303030303030303030",
    "pan": "70431001123456789",
    "expirationDate": "1122",
    "entryMode": 3
  },
  "currencyCode": "985",
  "response": {
    "code": 0,
    "message": "Zaakceptowano"
  },
  "authorizationCode": "145356",
  "customerData": {
    "odometer": 128512
  },
  "allowedProducts":
[6,9,10,12,13,17,22,20,28,30,32,33,36,38,42,48,58,64,78,80,81,87,85,90,88,95,83,
94,96,1001,1002,1004,1003,1005],
  "originalAmount": 25600
}

```

#### 4 AA Authorization advice

##### Example 1

Approved authorization advice (to previously approved pre-authorization - 3 PA - Example 1)

Request data:

```

{
  "transactionId": 3566,
  "batchId": 4313,
  "type": 4,
  "fleet": "DKV",
  "retry": 0,
  "transactionDateTime": "2022-05-10T08:49:34",
  "transmissionDateTime": "2022-05-10T08:49:34",
  "amount": 25600,
  "cardData": {
    "track2Data":
"3730343331303031313233343536373839D3232313130303930303030303030303030",

```



```

        "price": 500,
        "amount": 25600,
        "internalProduct": {
            "name": "Benzyna Pb98",
            "productId": "12332"
        }
    },
    "originalTransaction": {
        "transactionId": 3565,
        "transactionDateTime": "2022-05-10T09:57:33.542+02:00"
    }
}

```

## 6 BL Card Balance

### Example 1

Approved card balance Request data:

```

{
    "transactionId": 6428,
    "batchId": 1993,
    "type": 6,
    "fleet": "DKV",
    "retry": 0,
    "transactionDateTime": "2020-02-21T10:46:08",
    "transmissionDateTime": "2020-02-21T10:46:14",
    "cardData": {
        "track2Data":
"37303433313030313030313138373831343D3232313130303930303030303030303030",
        "pan": "70431001001187814",
        "expirationDate": "1122"
    },
    "currencyCode": "985",
    "pinData": {
        "pinBlock": "C8DDBC1593F9EAC",
        "pinBlockKey": "7A46522A486917052B0B5967520E2748"
    }
}

```

Response data:

```

{
    "transactionId": 6428,
    "batchId": 1993,
    "type": 6,
    "retry": 0,
    "fleet": "DKV",
    "transactionDateTime": "2020-02-21T10:46:08.153+01:00",
    "transmissionDateTime": "2020-02-21T10:46:14.5560346+01:00",
    "amount": 8000,
    "cardData": {
        "track2Data":
"37303433313030313030313138373831343D3232313130303930303030303030303030",
        "pan": "70431001001187814",
        "expirationDate": "1122"
    }
}

```

```

    },
    "currencyCode": "985",
    "response": {
      "code": 0,
      "message": "Approved"
    },
    "authorizationCode": "337282",
    "allowedProducts": [
      6, 9, 10, 12, 13, 16, 19, 30, 32, 33, 36, 38, 58, 64, 78, 80, 85, 88, 95,
94
    ]
  }
}

```

## Example 2

Approved card balance for UTA transaction with cvc Request data:

```

{
  "transactionId": 83,
  "batchId": 1,
  "type": 6,
  "fleet": "UTA",
  "retry": 0,
  "transactionDateTime": "2022-03-10T10:31:13.386+01:00",
  "transmissionDateTime": "2022-03-10T10:52:06.6803394+01:00",
  "cardData": {
    "pan": "63620045678931659",
    "expirationDate": "1122",
    "cvc": "092"
  },
  "currencyCode": "985"
}

```

Response data:

```

{
  "transactionId": 83,
  "batchId": 1,
  "type": 6,
  "retry": 0,
  "fleet": "UTA",
  "transactionDateTime": "2022-03-10T10:31:13.386+01:00",
  "transmissionDateTime": "2022-03-10T10:52:06.6803394+01:00",
  "amount": 923719,
  "cardData": {
    "pan": "63620045678931659",
    "expirationDate": "1122",
    "entryMode": 3,
    "cvc": "092"
  },
  "currencyCode": "985",
  "response": {
    "code": 0,
    "message": "Zaakceptowano"
  },
  "authorizationCode": "000000",
  "allowedProducts": [

```



```

17     9, 28, 10, 33, 12, 24, 35, 30, 36, 32, 38, 3, 42, 48, 6, 13, 20, 21, 94,
    ]
    }

```

## 8 TC Tokenization of card

### Example 1

Approved card tokenization Request data:

```
{
  "transactionId": 6428,
  "type": 8,
  "fleet": "AVIA",
  "retry": 0,
  "transactionDateTime": "2022-02-21T10:46:08",
  "transmissionDateTime": "2022-02-21T10:48:14",
  "cardData": {
    "pan": "70431001001187814",
    "expirationDate": "1122",
    "entryMode" = 4
  },
  "currencyCode": "985",
  "pinData": {
    "pinBlock": "C8DDBC1593F9EAC",
    "pinBlockkey": "7A46522A486917052B0B5967520E2748"
  }
}
```

Response data:

```
{
    "transactionId": 6428,
    "type": 8,
    "retry": 0,
    "fleet": "AVIA",
    "transactionDateTime": "2020-02-21T10:46:08.153+01:00",
    "transmissionDateTime": "2020-02-21T10:46:54.5560346+01:00",
    "response": {
        "code": 0,
        "message": "Approved"
    },
    "authorizationCode": "337282",
    "newCardToken": {
        "token":
            "QUMZNDIZNDU2NZIWMjQwMzMxMjM1OTU5QUJBQkFCQUJBQkFCQUJBQkFCQUJBQkFCQUJBQkFCQUJBQUFBQUFBQUJCQkJCQkJCQkI=",
        "expDate": "2024-04-30T23:59:59+00:00"
    },
    "requiredCustomerData": [
        {
            "type": 1,
            "pattern": "^^[0-9]{1,7}$"
        },
        {
            "type": 2,
```

```
    "pattern": "^[[p{L}0-9-& ]{1,15}$"
  }
]
```

## Version information

Version : 1.24.0

## URI scheme

Host : testfleetapi.dmt.com.pl

BasePath : /FleetApiTest/v1

Schemes : HTTPS

## Consumes

- `application/json`

## Produces

- `application/json`

## Paths

### Authenticates user and returns token data

POST /authentication/login

#### Parameters

Type	Name	Description	Schema
Body	<b>loginModel</b> <i>required</i>	Object with login data	<a href="#">Login</a>

#### Responses

HTTP Code	Description	Schema
200	Login Success	<a href="#">Authentication</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- Authentication

## Refreshes user's tokens

```
GET /authentication/refreshTokens
```

### Responses

HTTP Code	Description	Schema
200	Login Success	<a href="#">Authentication</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

### Tags

- Authentication

### Security

Type	Name
apiKey	<a href="#">Bearer</a>

## Returns client data

```
GET /clients/{clientId}
```

### Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Client</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

### Tags

- clients

## Updates client data

```
PUT /clients/{clientId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer
Body	<b>client</b> <i>required</i>	Client data	<a href="#">Client</a>

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- clients

## Removes client

```
DELETE /clients/{clientId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content

## Tags

- clients

## Returns client global parameters

```
GET /clients/{clientId}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- clients

## Updates client global parameters

```
PUT /clients/{clientId}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer
Body	<b>globalParameters</b> <i>required</i>	Global parameters	<a href="#">GlobalParameters</a>

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- clients

## Creates and saves partner (supplier)

```
POST /clients/{clientId}/partners
```

### Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer
Body	<b>partner</b> <i>required</i>	Partner data	<a href="#">Partner</a>

### Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Partner</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

### Tags

- clients

## Returns list of partners

```
GET /clients/{clientId}/partners
```

### Parameters

Type	Name	Description	Schema	Default
Path	<b>clientId</b> <i>required</i>	Client identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	Partner name or part of the name to search for. Format like for LIKE operator in sql (% - any string, _ - single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	
Query	<b>taxIdNumber</b> <i>optional</i>	Partner tax identification number or part of the tax identification number to be searched. Format like for LIKE operator in sql (% - any string, _ - single character, etc.).	string	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">PartnersPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- clients

## Returns list of users

```
GET /clients/{clientId}/users
```

## Parameters

Type	Name	Description	Schema	Default
Path	<b>clientId</b> <i>required</i>	Client identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The username or part of the name to search for. Format like for LIKE operator in sql (% - any string, _-single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">UsersPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- clients

## Returns user data

```
GET /clients/{clientId}/users/{userId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>clientId</b> <i>required</i>	Client identifier	integer
Path	<b>userId</b> <i>required</i>	User identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">User</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- clients

## Returns list of supported fleets

GET /fleets

## Parameters

Type	Name	Description	Schema	Default
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The name of the fleet or part of the name to be searched. Format like for LIKE operator in sql (% - any string, _-single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">FleetsPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>



## Tags

- fleets

## Returns fleet data

```
GET /fleets/{fleetName}
```

### Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Fleet</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- fleets

## Creates and saves a group

```
POST /groups
```

### Parameters

Type	Name	Description	Schema
Body	<b>group</b> <i>required</i>	Group data	<a href="#">Group</a>

### Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Group</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns list of groups

```
GET /groups
```

### Parameters

Type	Name	Description	Schema	Default
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The name of the group or part of the name to be searched. Format like for LIKE operator in sql (% - any string, _-single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">GroupsPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns group data

```
GET /groups/{groupId}
```

### Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Group</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Updates group data

```
PUT /groups/{groupId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer
Body	<b>group</b> <i>required</i>	Group data	<a href="#">Group</a>

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Removes group

```
DELETE /groups/{groupId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content

## Tags

- groups

## Creates and saves client

```
POST /groups/{groupId}/clients
```

## Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer
Body	<b>client</b> <i>required</i>	Client data	<a href="#">Client</a>

## Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Client</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns list of clients

```
GET /groups/{groupId}/clients
```

## Parameters

Type	Name	Description	Schema	Default
Path	<b>groupId</b> <i>required</i>	Group identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The customer's name or part of the name to search for. Format like for LIKE operator in sql (% - any string, _-single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">ClientsPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns global parameter

```
GET /groups/{groupId}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Updates global parameters

```
PUT /groups/{groupId}/parameters
```

### Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer
Body	<b>globalParameters</b> <i>required</i>	Global parameters	<a href="#">GlobalParameters</a>

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns list of users

```
GET /groups/{groupId}/users
```

### Parameters

Type	Name	Description	Schema	Default
Path	<b>groupId</b> <i>required</i>	Group identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The username or part of the name to search for. Format like for LIKE operator in sql (% - any string, _-single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">UsersPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns user data

```
GET /groups/{groupId}/users/{userId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>groupId</b> <i>required</i>	Group identifier	integer
Path	<b>userId</b> <i>required</i>	User identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">User</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- groups

## Returns partner data

```
GET /partners/{partnerId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Partner</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Updates partner (supplier) data

```
PUT /partners/{partnerId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer
Body	<b>partner</b> <i>required</i>	Partner data	<a href="#">Partner</a>

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Removes partner (supplier)

```
DELETE /partners/{partnerId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer



## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content

## Tags

- partners

## Returns parameters

```
GET /partners/{partnerId}/fleets/{fleetName}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Updates parameters

```
PUT /partners/{partnerId}/fleets/{fleetName}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer
Body	<b>parameters</b> <i>required</i>	Parameters	<a href="#">Parameters</a>

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Returns global parameters

```
GET /partners/{partnerId}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Updates global parameters

```
PUT /partners/{partnerId}/parameters
```

### Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer
Body	<b>globalParameters</b> <i>required</i>	Global parameters	<a href="#">GlobalParameters</a>

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

### Tags

- partners

## Creates and saves station

```
POST /partners/{partnerId}/stations
```

### Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer
Body	<b>station</b> <i>required</i>	Fuel station data	<a href="#">Station</a>

### Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Station</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Returns list of partner's stations

```
GET /partners/{partnerId}/stations
```

### Parameters

Type	Name	Description	Schema	Default
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The name of the station or part of the name to be searched. Format like for LIKE operator in sql (% - any string, _ - single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">StationsPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Returns list of user

```
GET /partners/{partnerId}/users
```

### Parameters

Type	Name	Description	Schema	Default
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The username or part of the name to search for. Format like for LIKE operator in sql (% - any string, _ - single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">UsersPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Returns user data

```
GET /partners/{partnerId}/users/{userId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>partnerId</b> <i>required</i>	Partner identifier	integer
Path	<b>userId</b> <i>required</i>	User identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">User</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- partners

## Returns list of supported products.

```
GET /products
```

## Description

The vatValue of the product is based on country code.

## Parameters

Type	Name	Description	Schema	Default
Query	<b>countryCode</b> <i>optional</i>	Numeric country code <b>ISO 3166-1</b> , if parameter is not set then default value will be 616 (Poland).	string	"616"
Query	<b>sort</b> <i>optional</i>	Sorting response rules. List of comma separated fields, each with a possible unary negative to imply descending sort order, example: sort=name,-activity. If sorting object have complex fields then there should be possible to sort by fields of complex field for example: status.date where status is field of Update object.	string	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Products</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- products

## Returns list of product mapping for fleets

```
GET /products/fleets/{fleetName}
```

## Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">ProductsMapping</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- products

## Returns fuel station data

```
GET /stations/{stationId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Station identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Station</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Updates fuel station data

```
PUT /stations/{stationId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer
Body	<b>station</b> <i>required</i>	Station data	<a href="#">Station</a>

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Removes station

```
DELETE /stations/{stationId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content

## Tags

- stations

## Returns station's parameters

```
GET /stations/{stationId}/fleets/{fleetName}/parameters
```



## Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Updates station's parameters

```
PUT /stations/{stationId}/fleets/{fleetName}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer
Body	<b>parameters</b> <i>required</i>	Parameters	<a href="#">Parameters</a>

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Returns global parameters

```
GET /stations/{stationId}/parameters
```

### Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Updates station parameters

```
PUT /stations/{stationId}/parameters
```

### Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer
Body	<b>globalParameters</b> <i>required</i>	Global parameters	<a href="#">GlobalParameters</a>

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Creates and saves terminal

```
POST /stations/{stationId}/terminals
```

### Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer
Body	<b>terminal</b> <i>required</i>	Terminal data	<a href="#">Terminal</a>

### Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Terminal</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Returns station's terminal list

```
GET /stations/{stationId}/terminals
```

### Parameters

Type	Name	Description	Schema	Default
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	
Query	<b>serialNumber</b> <i>optional</i>	The serial number of the terminal or part of the number to be searched. Such a format as for the LIKE operator in sql (% - any string, _ single sign etc.).	string	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">TerminalsPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Creates and saves manual transaction on station's level (manual transaction)

```
POST /stations/{stationId}/transactions
```

## Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer
Body	<b>transaction</b> <i>required</i>	Manual transaction data	<a href="#">ManualTransaction</a>

## Responses

HTTP Code	Description	Schema
201	OK	<a href="#">TransactionResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Returns station's user list

```
GET /stations/{stationId}/users
```

## Parameters

Type	Name	Description	Schema	Default
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer	
Query	<b>limit</b> <i>optional</i>	the maximum amount to be returned	integer	10
Query	<b>name</b> <i>optional</i>	The username or part of the name to search for. Format like for LIKE operator in sql (% - any string, _ - single character, etc.).	string	
Query	<b>offset</b> <i>optional</i>	the starting index to return	integer	

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">UsersPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Returns user data

```
GET /stations/{stationId}/users/{userId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>stationId</b> <i>required</i>	Fuel station identifier	integer
Path	<b>userId</b> <i>required</i>	User identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">User</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- stations

## Creates and saves reconciliation for terminal (EOD) - single product in position (deprecated)

POST /terminals/eods

### Parameters

Type	Name	Description	Schema
Body	<b>transaction</b> <i>required</i>	Transaction data	<a href="#">EodTransaction</a>

### Responses

HTTP Code	Description	Schema
201	OK	<a href="#">TransactionResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Creates and saves transaction

POST /terminals/fleets/{fleetName}/transactions

### Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Body	<b>transaction</b> <i>required</i>	Transaction data	<a href="#">Transaction</a>

### Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Transaction</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

**Creates and saves reconciliation for terminal (EOD) - many products in one transaction.**

```
POST /terminals/transactions/eods
```

## Parameters

Type	Name	Description	Schema
Body	<b>transaction</b> <i>required</i>	Transaction data	<a href="#">TransactionEod</a>

## Responses

HTTP Code	Description	Schema
201	OK	<a href="#">TransactionResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

**Creates a request to process a reconciliation file located on SFTP server.**

```
POST /terminals/transactions/eods/eodfiles
```

## Parameters

Type	Name	Description	Schema	Default
Query	<b>eodFileName</b> <i>required</i>	EOD file name	string	
Query	<b>fileLocalization</b> <i>optional</i>	File localizations: * 0 - main SFTP server	enum ()	0

## Responses

HTTP Code	Description	Schema
201	OK	<a href="#">EodFile</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Security

Type	Name
apiKey	<a href="#">Bearer</a>

**Get Eod files status.** `eodFileId` or `eodFileName` can be used.

```
GET /terminals/transactions/eods/eodfiles
```

## Parameters

Type	Name	Description	Schema
Query	<b>eodFileId</b> <i>optional</i>	EOD file identifier	integer
Query	<b>eodFileName</b> <i>optional</i>	EOD file name	string

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">EodFilesPagedResponse</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Security

Type	Name
apiKey	<a href="#">Bearer</a>

## Returns terminal data

```
GET /terminals/{terminalId}
```



## Parameters

Type	Name	Description	Schema
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Terminal</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Updates terminal data

```
PUT /terminals/{terminalId}
```

## Parameters

Type	Name	Description	Schema
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer
Body	<b>terminal</b> <i>required</i>	Terminal data	<a href="#">Terminal</a>

## Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Removes terminal

```
DELETE /terminals/{terminalId}
```

### Parameters

Type	Name	Description	Schema
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer

### Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Incorrect identifier	No Content

### Tags

- terminals

## Returns terminal's parameters

```
GET /terminals/{terminalId}/fleets/{fleetName}/parameters
```

### Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

### Tags

- terminals

## Updates terminal parameters

```
PUT /terminals/{terminalId}/fleets/{fleetName}/parameters
```

### Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer
Body	<b>parameters</b> <i>required</i>	Parameters	<a href="#">Parameters</a>

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

### Tags

- terminals

## Creates and saves transaction

```
POST /terminals/{terminalId}/fleets/{fleetName}/transactions
```

### Parameters

Type	Name	Description	Schema
Path	<b>fleetName</b> <i>required</i>	Fleet identifier	string
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer
Body	<b>transaction</b> <i>required</i>	Transaction data	<a href="#">Transaction</a>

## Responses

HTTP Code	Description	Schema
201	OK	<a href="#">Transaction</a>
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Security

Type	Name
apiKey	<a href="#">Bearer</a>

## Returns global parameters

```
GET /terminals/{terminalId}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

## Updates terminal parameters

```
PUT /terminals/{terminalId}/parameters
```

## Parameters

Type	Name	Description	Schema
Path	<b>terminalId</b> <i>required</i>	Terminal identifier	integer
Body	<b>globalParameters</b> <i>required</i>	Global parameters	<a href="#">GlobalParameters</a>

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">AllGlobalParameters</a>
401	Incorrect identifier	No Content
default	Error code (400, 500)	<a href="#">ResponseCode</a>

## Tags

- terminals

# Definitions

## AccessMode

User rights

Name	Schema
<b>create</b> <i>optional</i>	boolean
<b>delete</b> <i>optional</i>	boolean
<b>read</b> <i>optional</i>	boolean
<b>write</b> <i>optional</i>	boolean

## ActivityPeriod

Activity period

Name	Description	Schema
<b>dateFrom</b> <i>required</i>	Date and time activity FROM	string (dateTime)
<b>dateTo</b> <i>optional</i>	Date and time activity TO - empty value means always active	string (dateTime)

## AllGlobalParameters

Table of global parameters

Name	Schema
<b>globalParameters</b> <i>optional</i>	<a href="#">GlobalParameters</a>
<b>inheritedGlobalParameters</b> <i>optional</i>	<a href="#">GlobalParameters</a>

## AllParameters

Table of all parameters

Name	Schema
<b>inheritedParameters</b> <i>optional</i>	<a href="#">Parameters</a>
<b>parameters</b> <i>optional</i>	<a href="#">Parameters</a>

## Authentication

Authentication model - response for login action

Name	Schema
<b>authorizationToken</b> <i>optional</i>	<a href="#">Token</a>
<b>loggedUserContext</b> <i>optional</i>	<a href="#">LoggedUserContext</a>
<b>refreshToken</b> <i>optional</i>	<a href="#">Token</a>

## CardData

An object representing the data of the fleet card

Name	Description	Schema
<b>cardToken</b> <i>optional</i>	Card token	<a href="#">CardToken</a>
<b>cvc</b> <i>optional</i>	Card Verification Code <b>Maximal length</b> : 3 <b>Pattern</b> : <code>"^[0-9]{3}\$"</code>	string
<b>entryMode</b> <i>optional</i>	Entry mode type by default set to 3(magstripe)	<a href="#">EntryModeDataType</a>
<b>expirationDate</b> <i>optional</i>	card expiry date (formt <code>MMYY</code> ) <b>Maximal length</b> : 4 <b>Pattern</b> : <code>"^(0[1-9] 1[0-2])([0-9]{2})\$"</code>	string
<b>iccRelatedData</b> <i>optional</i>	Icc related data (HEXSTR)	string
<b>pan</b> <i>optional</i>	card number (PAN) <b>Maximal length</b> : 19 <b>Pattern</b> : <code>"^[0-9]{1,19}\$"</code>	string
<b>track2Data</b> <i>optional</i>	Track 2 value of a magstripe card (HEXSTR)	string

## CardToken

An object representing card token

Name	Description	Schema
<b>expDate</b> <i>optional</i>	Token expiry date in UTC	string (dateTime)
<b>token</b> <i>optional</i>	Token value - Base64 encode	string

## Client

Client

Name	Description	Schema
<b>clientId</b> <i>optional</i> <i>read-only</i>	Client identifier	integer
<b>groupId</b> <i>optional</i>	Group identifier	integer
<b>internalParameters</b> <i>optional</i>		< <a href="#">ClientInternalParameter</a> > array
<b>name</b> <i>optional</i>	Client name <b>Maximal length</b> : 50	string

## ClientInternalParameter

Internal client parameter

Name	Description	Schema
<b>fleetName</b> <i>required</i>	Fleet identifier <b>Maximal length</b> : 50	string
<b>name</b> <i>required</i>	Parameter name <b>Maximal length</b> : 50	string
<b>value</b> <i>optional</i>	Parameter value (empty value removes parameter) <b>Maximal length</b> : 255	string

## Clients

Table of clients

Name	Schema
<b>clients</b> <i>optional</i>	< <a href="#">Client</a> > array

## ClientsPagedResponse

*Polymorphism* : Composition



Name	Description	Schema
<b>clients</b> <i>optional</i>		< <a href="#">Client</a> > array
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>total</b> <i>optional</i>	Total count	integer

## CustomerData

An object representing the card holder's data

Name	Description	Schema
<b>DriverId</b> <i>optional</i>	Driver identifier within the company. Format: "[0-9]{1,12}\$" <b>Maximal length</b> : 12	string
<b>UnencryptedId</b> <i>optional</i>	Additional identifier requested by company during refuelling. Format: "[0-9]{1,10}\$" <b>Maximal length</b> : 10	string
<b>VehicleNumber</b> <i>optional</i>	Vehicle number is an identifier of a vehicle within the company (not plate number). Format: "[0-9]{1,10}\$" <b>Maximal length</b> : 10	string
<b>additionalTrack</b> <i>optional</i>	(Additional track 2 value of a magstripe card (HEXSTR)) <b>Maximal length</b> : 80	string
<b>costCenter</b> <i>optional</i>	Cost center (MPK) format: "[0-9]{1,10}\$" <b>Maximal length</b> : 50	string
<b>driverName</b> <i>optional</i>	Driver name <b>Maximal length</b> : 50	string
<b>odometer</b> <i>optional</i>	odometer reading value <b>Maximum value</b> : 9999999	integer
<b>plateNumber</b> <i>optional</i>	Car registration number (format: "[a-zA-Z0-9-& ]{1,15}\$")	string

## CustomerDataDefinition

Definition of customer data

Name	Description	Schema
<b>pattern</b> <i>optional</i>	The value of the regular expression parsing the data <b>Maximal length</b> : 128	string
<b>type</b> <i>optional</i>		<a href="#">CustomerDataType</a>

## CustomerDataType

Type of the customer data:

- 1 - odometer
- 2 - plate number
- 3 - cost center (MPK)
- 4 - driver name
- 5 - vehicle number
- 6 - driver id
- 7 - unencrypted id
- 8 - additional track

*Type* : enum (1, 2, 3, 4, 5, 6, 7, 8)

## EntryModeDataType

Type of the entry mode:

- 1 - Chip
- 2 - CTLS
- 3 - Magstripe
- 4 - Manual entry
- 5 - Magstripe fallback
- 6 - Card token

*Type* : enum (1, 2, 3, 4, 5, 6)

## EodFile

Eod file process status

Name	Description	Schema
<b>currentStatus</b> <i>optional</i>		<a href="#">EodFileStatus</a>
<b>id</b> <i>optional</i>	EOD file identifier	integer
<b>name</b> <i>optional</i>	EOD file name. File name format: TF{ <i>Client_name</i> }YYYYMMDD_SSS.json, where: * {Client_name} - client name * YYYYMMDD - file generation date in format YYYYMMDD * SSS - sequence number in current day (left-padded with zeros to 3 digits)	string

## EodFileStatus

Eod file process status

Name	Description	Schema
<b>code</b> <i>optional</i>		<a href="#">EodFileStatusCode</a>
<b>dateTime</b> <i>optional</i>		string (date-time)
<b>message</b> <i>optional</i>	Status code description (it only matters for errors)	string

## EodFileStatusCode

Processing EOD file status code:

- 0 - added
- 1 - processing
- 2 - processed with errors
- 3 - processed successfully
- 4 - partially processed

*Type* : enum (0, 1, 2, 3, 4)

## EodFiles

Table of Eod files

Name	Schema
<b>eodFiles</b> <i>optional</i>	< <a href="#">EodFile</a> > array

## EodFilesPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>eodFiles</b> <i>optional</i>		< <a href="#">EodFile</a> > array
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>total</b> <i>optional</i>	Total count	integer

## EodItem

An object representing the position to be settled

Name	Description	Schema
<b>authorizationCode</b> <i>required</i>	authorisation code, Patterns: AVIA "[a-zA-Z0-9]{6}" DCI "[a-zA-Z0-9]{6}" DKV "[a-zA-Z0-9]{6,10}" E100 "[0-9]{6}" EKOTANK "[0-9]{6}" EUROWAG "[0-9]{6}" LOTOS "[0-9]{6}" UTA "[0-9]{5,6}" IQCARD "[a-zA-Z0-9]{6}"	string
<b>cardData</b> <i>required</i>		<a href="#">CardData</a>
<b>currencyCode</b> <i>required</i>	currency code - numeric ( ISO 4217 ) <b>Pattern</b> : "[0-9]{3}"	string
<b>customerData</b> <i>optional</i>		<a href="#">CustomerData</a>
<b>fleet</b> <i>required</i>	fleet identifier	<a href="#">FleetType</a>
<b>gateNumber</b> <i>optional</i>	gate number for toll road <b>Maximal length</b> : 9 <b>Pattern</b> : "[0-9]{1,9}"	string
<b>product</b> <i>required</i>		<a href="#">TransactionProduct</a>
<b>receiptId</b> <i>required</i>	receipt number <b>Maximal length</b> : 10 <b>Pattern</b> : "[0-9]{1,10}"	string
<b>transactionDateTime</b> <i>required</i>	transaction date and time	string (dateTime)
<b>transactionId</b> <i>required</i>	transaction identifier set by terminal (STAN) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>type</b> <i>required</i>	transaction type ( <a href="#">TransactionType</a> )	<a href="#">TransactionType</a>

## EodTransaction

An object representing a transaction settlement operation (EOD)

Name	Description	Schema
<b>amount</b> <i>required</i>	sum of item amounts (for items of type 5 - reversal (withdrawal) ** RV , <b>we treat the sums when adding up as negative</b> Minimum value** : 0 <b>Maximum value</b> : 999999999999	integer (int64)
<b>batchId</b> <i>required</i>	Batch number <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>items</b> <i>optional</i>		< <a href="#">EodItem</a> > array
<b>number</b> <i>required</i>	number of transaction in the batch	integer

## Fleet

Fleet object

Name	Description	Schema
<b>name</b> <i>optional</i>	Fleet name	<a href="#">FleetType</a>
<b>ranges</b> <i>optional</i>	Fleet card number range	< <a href="#">FleetCardsRange</a> > array
<b>requiredParameters</b> <i>optional</i>	Mandatory parameters to set	< <a href="#">ParameterDefinition</a> > array

## FleetCardsRange

Fleet card range object

Name	Description	Schema
<b>from</b> <i>optional</i> <i>read-only</i>	range from <b>Maximal length</b> : 19	string
<b>length</b> <i>optional</i> <i>read-only</i>	Card PAN length	integer
<b>name</b> <i>optional</i> <i>read-only</i>	Name for sub-fleet <b>Maximal length</b> : 50	string
<b>offline</b> <i>optional</i> <i>read-only</i>	Offline transaction allowence flag	boolean
<b>to</b> <i>optional</i> <i>read-only</i>	range to <b>Maximal length</b> : 19	string

## FleetSpecificData

Object represents fleet specific data (key-value)

Type : < [FleetSpecificData](#) > array

### FleetSpecificData

Name	Description	Schema
<b>key</b> <i>optional</i>	Fleet specific data key	string
<b>value</b> <i>optional</i>	Fleet specific data value	string

## FleetType

Fleet system type

Type : string

## Fleets

Table of fleets

Name	Schema
<b>fleets</b> <i>optional</i>	< <a href="#">Fleet</a> > array

## FleetsPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>fleets</b> <i>optional</i>		< <a href="#">Fleet</a> > array
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>total</b> <i>optional</i>	Total count	integer

## GlobalParameter

The object represents a global parameter that can be set for the terminal, station, partner

Name	Description	Schema
<b>name</b> <i>required</i>	global parameter name <b>Maximal length</b> : 50	string
<b>value</b> <i>optional</i>	global parameter value <b>Maximal length</b> : 255	string

## GlobalParameterDefinition

The object represents the definition of a global parameter that can be set for terminal, station, partner, customer, group



Name	Description	Schema
<b>allowedForClient</b> <i>optional</i>	determines whether a given global parameter can be set at the client level	boolean
<b>allowedForGroup</b> <i>optional</i>	determines whether a given global parameter can be set at the group level	boolean
<b>allowedForPartner</b> <i>optional</i>	determines whether a given global parameter can be set at the partner level	boolean
<b>allowedForStation</b> <i>optional</i>	determines whether a given global parameter can be set at the station level	boolean
<b>allowedForTerminal</b> <i>optional</i>	determines whether a given global parameter can be set at the terminal level	boolean
<b>defaultValue</b> <i>optional</i>	Global parameter value <b>Maximal length</b> : 255	string
<b>description</b> <i>optional</i>	Global parameter description	string
<b>isUnique</b> <i>optional</i>	determines whether a given global parameter must be unique in the whole tree	boolean
<b>name</b> <i>optional</i>	Global parameter name <b>Maximal length</b> : 50	string
<b>pattern</b> <i>optional</i>	Regular expression for control of global parameter value	string
<b>required</b> <i>optional</i>	determines whether a given global parameter must be set	boolean

## GlobalParameters

Table of global parameters

Name	Schema
<b>globalParameters</b> <i>optional</i>	< <a href="#">GlobalParameter</a> > array

## Group

Group

Name	Description	Schema
<b>groupId</b> <i>optional</i> <i>read-only</i>	Group identifier	integer
<b>name</b> <i>optional</i>	Group name <b>Maximal length</b> : 50	string

## Groups

Table of groups

Name	Schema
<b>groups</b> <i>optional</i>	< <a href="#">Group</a> > array

## GroupsPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>groups</b> <i>optional</i>		< <a href="#">Group</a> > array
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>total</b> <i>optional</i>	Total count	integer

## InternalTransactionProduct

An object that represents product properties in the client system

Name	Description	Schema
<b>name</b> <i>optional</i>	internal product name <b>Maximal length</b> : 25	string
<b>productId</b> <i>optional</i>	internal product identifier <b>Maximal length</b> : 6 <b>Pattern</b> : <code>"^[a-zA-Z0-9_]{1,6}\$"</code>	string

## LoggedUserContext

Logged user's data

Name	Description	Schema
<b>login</b> <i>optional</i> <i>read-only</i>	User's login	string
<b>userId</b> <i>optional</i> <i>read-only</i>	User's identifier	integer

## Login

User login data

Name	Schema
<b>password</b> <i>optional</i>	string (password)
<b>username</b> <i>optional</i>	string

## ManualCardData

An object representing the data of the fleet card - manual entry

Name	Description	Schema
<b>expirationDate</b> <i>optional</i>	card expiry date (formt <code>MMYY</code> ) <b>Maximal length</b> : 4 <b>Pattern</b> : <code>"^(0[1-9]\' 1[0-2])\'([0-9]{2})\$"</code>	string
<b>pan</b> <i>optional</i>	Card number (PAN) <b>Maximal length</b> : 19 <b>Pattern</b> : <code>"^[0-9]{1,19}\$"</code>	string

## ManualTransaction

An object representing a manually entered sales transaction confirmed by an authorization code obtained by phone

Name	Description	Schema
<b>amount</b> <i>required</i>	transaction amount (two decimal digits) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999999999	integer (int64)
<b>authorizationCode</b> <i>required</i>	authorisation code, Patterns: AVIA "[a-zA-Z0-9]{6}" DCI "[a-zA-Z0-9]{6}" DKV "[a-zA-Z0-9]{6,10}" E100 "[0-9]{6}" EKOTANK "[0-9]{6}" EUROWAG "[0-9]{6}" LOTOS "[0-9]{6}" UTA "[0-9]{5,6}" IQCARD "[a-zA-Z0-9]{6}"	string
<b>complaintId</b> <i>optional</i>	complaint identifier	integer
<b>currencyCode</b> <i>required</i>	currency code - numeric ( ISO 4217 ) <b>Pattern</b> : "[0-9]{3}"	string
<b>customerData</b> <i>optional</i>		<a href="#">CustomerData</a>
<b>fleet</b> <i>required</i>	fleet identifier	<a href="#">FleetType</a>
<b>gateNumber</b> <i>optional</i>	gate number for toll road <b>Maximal length</b> : 9 <b>Pattern</b> : "[0-9]{1,9}"	string
<b>manualCardData</b> <i>required</i>		<a href="#">ManualCardData</a>
<b>products</b> <i>required</i>		< <a href="#">TransactionProduct</a> > array
<b>receiptId</b> <i>required</i>	receipt number <b>Maximal length</b> : 10 <b>Pattern</b> : "[0-9]{1,10}"	string
<b>transactionDateTime</b> <i>required</i>	transaction date and time	string (dateTime)
<b>type</b> <i>required</i>	manual transaction type ( ManualTransactionType )	<a href="#">ManualTransactionType</a>

## ManualTransactionType

An object representing possible manual transactions to be performed

- 1 - sale
- 2 - reversal

Type : enum (1, 2)

## MeasureUnit

Measure unit

Type : enum (L, W, Q)

## OriginalTransaction

An object representing base transaction data.

Name	Description	Schema
<b>transactionDateTime</b> <i>optional</i>	transaction date and time	string (dateTime)
<b>transactionId</b> <i>optional</i>	Transaction identifier set by terminal (STAN) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer

## PagedResponse

Name	Description	Schema
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>total</b> <i>optional</i>	Total count	integer

## Parameter

The object represents the fleet parameter that can be set for the terminal, station, partner

Name	Description	Schema
<b>name</b> <i>required</i>	parameter name <b>Maximal length</b> : 50	string
<b>value</b> <i>optional</i>	parameter value <b>Maximal length</b> : 255	string

## ParameterDefinition

The object represents the definition of the fleet parameter that can be set for terminal, station, partner

Name	Description	Schema
<b>allowedForPartner</b> <i>optional</i>	determines whether a given parameter can be set at the partner level	boolean
<b>allowedForStation</b> <i>optional</i>	determines whether a given parameter can be set at the station level	boolean
<b>allowedForTerminal</b> <i>optional</i>	determines whether a given parameter can be set at the terminal level	boolean
<b>defaultValue</b> <i>optional</i>	parameter value <b>Maximal length</b> : 255	string
<b>description</b> <i>optional</i>	parameter description	string
<b>isUnique</b> <i>optional</i>	determines whether a given parameter must be unique in the entire client tree	boolean
<b>name</b> <i>optional</i>	parameter name <b>Maximal length</b> : 50	string
<b>pattern</b> <i>optional</i>	regular expression for parameter value control	string
<b>required</b> <i>optional</i>	determines whether the parameter must be set	boolean

## Parameters

Table of parameters

Name	Schema
<b>parameters</b> <i>optional</i>	< <a href="#">Parameter</a> > array

## Partner

Partner

Name	Description	Schema
<b>activityPeriod</b> <i>optional</i>		<a href="#">ActivityPeriod</a>
<b>address</b> <i>optional</i>	Partner address	string
<b>city</b> <i>optional</i>	Partner city	string
<b>clientId</b> <i>optional</i>	Client identifier	integer
<b>countryCode</b> <i>optional</i>	Numeric country code of the partner <b>ISO 3166-1</b> <b>Maximal length</b> : 3	string
<b>currencyCode</b> <i>optional</i>	Numeric currency code of the partner <b>ISO 4217</b> <b>Maximal length</b> : 3	string
<b>fullName</b> <i>optional</i>	Partner full name	string
<b>name</b> <i>optional</i>	Partner name <b>Maximal length</b> : 50	string
<b>partnerId</b> <i>optional</i> <i>read-only</i>	Partner identifier	integer
<b>relatedPartnerId</b> <i>optional</i>	If partner is related to other partner (partner in the same client) then it can have the same value of SUPPLIER_ID	integer
<b>taxNumber</b> <i>optional</i>	Partner tax number	string
<b>zipCode</b> <i>optional</i>	Partner post code	string

## Partners

Table of partners

Name	Schema
<b>partners</b> <i>optional</i>	< <a href="#">Partner</a> > array

## PartnersPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>partners</b> <i>optional</i>		< <a href="#">Partner</a> > array
<b>total</b> <i>optional</i>	Total count	integer

## PinData

An object representing the data of the transmitted pinblock

Name	Description	Schema
<b>pinBlock</b> <i>optional</i>	Value of pinBlock (HEXSTR)	string
<b>pinBlockKey</b> <i>optional</i>	Random value for pin encryption key derivation (HEXSTR)	string

## Product

obiekt reprezentujący produkt

Name	Description	Schema
<b>fuelProduct</b> <i>optional</i>	Determines if the product is a fuel	boolean
<b>measureUnit</b> <i>optional</i>		<a href="#">MeasureUnit</a>
<b>name</b> <i>optional</i>	product name	string
<b>productId</b> <i>optional</i>	product identifier	integer
<b>remarks</b> <i>optional</i>	Product description <b>Maximal length</b> : 255	string
<b>restrictionCodeDKV</b> <i>optional</i>	Restriction code for DKV fleet	integer
<b>taxCode</b> <i>optional</i>	VAT group <b>Maximal length</b> : 10	string
<b>vat</b> <i>optional</i>	vat value (two decimal digits)	integer



## ProductMapping

an object representing the product identifier mapping

Name	Description	Schema
<b>fleetProductId</b> <i>optional</i>	fleet product identifier	string
<b>productId</b> <i>optional</i>	product identifier	integer

## Products

Table of products

Name	Schema
<b>products</b> <i>optional</i>	< <a href="#">Product</a> > array

## ProductsMapping

Table of product code mapping

Name	Schema
<b>products</b> <i>optional</i>	< <a href="#">ProductMapping</a> > array

## ResponseCode

Response code.

Name	Description	Schema
<b>code</b> <i>required</i>	Error code. Possible values: * 1 - internal error * 2 - incorrect request parameters (incorrect parameters in the URL) * 3 - incorrect request content (incorrect json / body object method)	integer
<b>message</b> <i>required</i>	Error description	string

## Rights

User rights (other than standard)

Name	Description	Schema
<b>complaint</b> <i>optional</i>	complaint right	boolean
<b>complaintProcess</b> <i>optional</i>	complaint process right	boolean
<b>reportAccess</b> <i>optional</i>	report access right	boolean
<b>userManagemenet</b> <i>optional</i>	user management access right	boolean

## Station

Fuel station

Name	Description	Schema
<b>activityPeriod</b> <i>optional</i>		<a href="#">ActivityPeriod</a>
<b>address</b> <i>optional</i>	Station address	string
<b>city</b> <i>optional</i>	Station city	string
<b>fullName</b> <i>optional</i>	Station full name <b>Maximal length</b> : 255	string
<b>mcc</b> <i>optional</i>	Station MCC	string
<b>mid</b> <i>optional</i>	Station MID	string
<b>name</b> <i>optional</i>	Fuel station name <b>Maximal length</b> : 255	string
<b>partnerId</b> <i>optional</i>	Fuel station's partner ID	integer
<b>stationId</b> <i>optional</i> <i>read-only</i>	Fuel station identifier	integer
<b>zipCode</b> <i>optional</i>	Station post code	string

## Stations

Table of Fuel station

Name	Schema
<b>stations</b> <i>optional</i>	< <a href="#">Station</a> > array

## StationsPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>stations</b> <i>optional</i>		< <a href="#">Station</a> > array
<b>total</b> <i>optional</i>	Total count	integer

## Terminal

Terminal in fuel station

Name	Description	Schema
<b>activityPeriod</b> <i>required</i>		<a href="#">ActivityPeriod</a>
<b>batch</b> <i>optional</i>	present batch number <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>serialNumber</b> <i>required</i>	terminal serial number <b>Maximal length</b> : 50	string
<b>stationId</b> <i>optional</i>	The identifier of the station to which the terminal is attached	integer
<b>terminalId</b> <i>optional</i> <i>read-only</i>	Terminal identifier assigned by the system	integer
<b>terminalType</b> <i>optional</i>		<a href="#">TerminalType</a>
<b>tid</b> <i>optional</i>	Terminal identifier TID <b>Maximal length</b> : 8 <b>Pattern</b> : <code>"^[0-9]{1,8}\$"</code>	string
<b>transactionId</b> <i>optional</i>	identifier of the last transaction <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>transactionPassword</b> <i>optional</i> <i>read-only</i>	The password of user which have rights to perform transactions on the terminal (value generated / transmitted when adding a terminal)	string
<b>transactionUser</b> <i>optional</i> <i>read-only</i>	The username which have rights to perform transactions on the terminal (value generated / transmitted when adding a terminal)	string

## TerminalType

typ terminala:

- 0 - desktop (desktop payment terminal - indoor)
- 1 - unattended (selfservice outdoor terminal)
- 2 - www

Type : enum (0, 1, 2)

## Terminals

Table of the terminals

Name	Schema
<b>terminals</b> <i>optional</i>	< <a href="#">Terminal</a> > array

## TerminalsPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>terminals</b> <i>optional</i>		< <a href="#">Terminal</a> > array
<b>total</b> <i>optional</i>	Total count	integer

## Token

Authorization token's data

Name	Description	Schema
<b>expires</b> <i>optional</i> <i>read-only</i>	Token's expiration date	string (date-time)
<b>token</b> <i>optional</i> <i>read-only</i>	Token	string

## Transaction

The object representing the transaction. The `response` field is completed in answers to sent data (in the request data this field is ignored)

Name	Description	Schema
<b>allowedProducts</b> <i>optional</i>	allowed product identifiers list	< integer > array
<b>amount</b> <i>optional</i>	transaction amount (two decimal digits) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999999999	integer (int64)
<b>authorizationCode</b> <i>optional</i>	authorisation code, Patterns: AVIA "[a-zA-Z0-9]{6}" DCI "[a-zA-Z0-9]{6}" DKV "[a-zA-Z0-9]{6,10}" E100 "[0-9]{6}" EKOTANK "[0-9]{6}" EUROWAG "[0-9]{6}" LOTOS "[0-9]{6}" UTA "[0-9]{5,6}" IQCARD "[a-zA-Z0-9]{6}"	string
<b>batchId</b> <i>required</i>	the number of the settlement package (batch number of the EOD operation in which current transaction will be sent) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>cardData</b> <i>optional</i>		<a href="#">CardData</a>
<b>currencyCode</b> <i>optional</i>	numeric currency code <b>ISO 4217</b> <b>Pattern</b> : "[0-9]{3}"	string
<b>customerData</b> <i>optional</i>		<a href="#">CustomerData</a>
<b>fleet</b> <i>optional</i>	fleet identifier	<a href="#">FleetType</a>
<b>fleetSpecificData</b> <i>optional</i>		<a href="#">FleetSpecificData</a>
<b>gateNumber</b> <i>optional</i>	gate number for toll road <b>Maximal length</b> : 9 <b>Pattern</b> : "[0-9]{1,9}"	string
<b>newCardToken</b> <i>optional</i>	Generated card token	<a href="#">CardToken</a>
<b>newPinData</b> <i>optional</i>		<a href="#">PinData</a>
<b>originalAmount</b> <i>optional</i>	preauthorisation amount (two decimal digits) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999999999	integer (int64)

Name	Description	Schema
<b>originalTransaction</b> <i>optional</i>		<a href="#">OriginalTransaction</a>
<b>pinData</b> <i>optional</i>		<a href="#">PinData</a>
<b>products</b> <i>optional</i>		< <a href="#">TransactionProduct</a> > array
<b>requiredCustomerData</b> <i>optional</i>		< <a href="#">CustomerDataDefinition</a> > array
<b>response</b> <i>optional</i>		<a href="#">TransactionResponse</a>
<b>retry</b> <i>optional</i>	repeat flag (0 - original transaction, 1 - repeat transkaction)	integer
<b>reversalReason</b> <i>optional</i>	Cancellation reason: * 0 - automatic reversal (communication error) * 1 - cancellation at the request of the operator of the terminal / sales system * 2 - other errors	integer
<b>transactionDateTime</b> <i>optional</i>	transaction date and time	string (dateTime)
<b>transactionId</b> <i>optional</i>	Transaction identifier set by terminal (STAN) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>transmissionDateTime</b> <i>optional</i>	transmission date and time	string (dateTime)
<b>type</b> <i>required</i>	Transaction type ( <a href="#">TransactionType</a> )	<a href="#">TransactionType</a>

## TransactionEod

An object representing a transaction settlement operation (EOD) - table of product in TransactionEodItem

Name	Description	Schema
<b>amount</b> <i>required</i>	sum of item amounts (for items of type 5 - reversal (withdrawal) ** RV , <b>we treat the sums when adding up as negative)</b> Minimum value** : 0 <b>Maximum value</b> : 999999999999	integer (int64)
<b>batchId</b> <i>required</i>	batch number <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>items</b> <i>optional</i>		< <a href="#">TransactionEodItem</a> > array
<b>number</b> <i>required</i>	number of transaction in batch	integer

## TransactionEodItem

An object representing the position to be settled (many products in one position)



Name	Description	Schema
<b>authorizationCode</b> <i>required</i>	authorisation code, Patterns: AVIA "[a-zA-Z0-9]{6}" DCI "[a-zA-Z0-9]{6}" DKV "[a-zA-Z0-9]{6,10}" E100 "[0-9]{6}" EKOTANK "[0-9]{6}" EUROWAG "[0-9]{6}" LOTOS "[0-9]{6}" UTA "[0-9]{5,6}" IQCARD "[a-zA-Z0-9]{6}"	string
<b>cardData</b> <i>required</i>		<a href="#">CardData</a>
<b>currencyCode</b> <i>required</i>	currency code - numeric ( ISO 4217 ) <b>Pattern</b> : "[0-9]{3}"	string
<b>customerData</b> <i>optional</i>		<a href="#">CustomerData</a>
<b>fleet</b> <i>required</i>	fleet identifier	<a href="#">FleetType</a>
<b>gateNumber</b> <i>optional</i>	gate number for toll road <b>Maximal length</b> : 9 <b>Pattern</b> : "[0-9]{1,9}"	string
<b>products</b> <i>optional</i>		< <a href="#">TransactionProduct</a> > array
<b>receiptId</b> <i>required</i>	receipt number <b>Maximal length</b> : 10 <b>Pattern</b> : "[0-9]{1,10}"	string
<b>transactionDateTime</b> <i>required</i>	transaction date and time	string (dateTime)
<b>transactionId</b> <i>required</i>	transaction id set by terminal (STAN) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999	integer
<b>type</b> <i>required</i>	transaction type	<a href="#">TransactionType</a>

## TransactionProduct

An object representing one product in the transaction

Name	Description	Schema
<b>amount</b> <i>optional</i>	product value (two decimal digits) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999999999	integer (int64)
<b>internalProduct</b> <i>optional</i>		<a href="#">InternalTransactionProduct</a>
<b>price</b> <i>optional</i>	unit price (two decimal digits) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999999	integer
<b>productId</b> <i>optional</i>	product identifier ( Product.productId )	integer
<b>quantity</b> <i>optional</i>	product quantity (two decimal digits) <b>Minimum value</b> : 1 <b>Maximum value</b> : 999999999	integer

## TransactionResponse

Transaction response object

Name	Description	Schema
<b>code</b> <i>optional</i>	Response code * 0 Approved * 1 Honour, with Identification * 2 Approved for partial amount * 3 Approved (VIP) * 7 Approved, update ICC * 80 Approved (liability not accepted) * 100 Do not honour * 101 Expired card * 102 Suspected fraud * 103 Card Acceptor contact acquirer * 104 Restricted card * 106 Allowable PIN Tries exceeded * 107 Refer to Card Issuer * 109 Invalid Merchant * 110 Invalid Amount * 111 Invalid Card Number * 112 PIN data required * 114 No account of type requested * 115 Requested Function not supported * 116 Not sufficient funds * 117 Incorrect PIN * 118 No card record * 119 Transaction not permitted to the customer * 120 Transaction not permitted to the terminal * 121 Exceeds withdrawal amount limit * 123 Exceeds withdrawal frequency limit * 125 Card not effective * 126 Invalid PIN block * 127 PIN length error * 128 PIN key synch error * 180 Redemption denied by Loyalty * 181 Card blocked * 182 Account blocked * 185 Product(s) not allowed * 186 Allowable PIN tries exceeded * 187 Wrong odometer reading * 188 Limit exceeded, contact FM * 189 Site / Time not allowed, contact FM * 191 Unknown POS terminal * 200 Pick-up card * 201 Expired card * 202 Suspected fraud * 203 Card acceptor contact acquirer * 204 Restricted card * 206 Allowable PIN tries exceeded * 207 Special condition * 208 Lost Card * 209 Stolen Card * 288 Too many cards on account	integer

Name	Description	Schema
	<ul style="list-style-type: none"> <li>* 289 Already migrated</li> <li>* 290 Customer registration required</li> <li>* 291 Migration has to undertaken at the callcenter</li> <li>* 292 Migration validity period expired</li> <li>* 300 PIN change required</li> <li>* 400 Reversal advice Accepted</li> <li>* 480 Reversal advice Accepted, no original transaction</li> <li>* 500 Reconciled; In balance</li> <li>* 501 Reconciled; Out of balance</li> <li>* 580 Reconciled; Out of balance do not attempt error recovery</li> <li>* 800 Accepted Transaction advice acknowledgement</li> <li>* 900 Advice acknowledged - no financial liability accepted</li> <li>* 901 Advice acknowledged</li> <li>* 902 Invalid transaction</li> <li>* 903 Re-enter transaction</li> <li>* 904 Format error</li> <li>* 906 Cutover in progress</li> <li>* 907 Card Issuer or Switch inoperative</li> <li>* 909 System malfunction</li> <li>* 912 Card issuer unavailable</li> <li>* 911 Card issuer timed out</li> <li>* 916 MAC incorrect</li> <li>* 917 MAC key synch error</li> <li>* 921 Security Software Hardware Error</li> <li>* 922 Message number out of sequence</li> <li>* 923 Request in progress</li> <li>* 930 Operation not allowed for the fleet</li> <li>* 940 Product's amount sum is different than transaction amount</li> <li>* 941 Additional data is incorrect</li> <li>* 942 Authorization code is incorrect</li> <li>* 943 Fleet is inactive</li> <li>* 944 Transaction currency is incorrect</li> <li>* 945 Transaction/batch number is incorrect</li> </ul>	
<b>message</b> <i>optional</i>	response message	string

## TransactionType

An object representing transactions supported by the FleetAPI

- 1 - handshake (connection test)
- 2 - sale online (sale on-line)
- 3 - preauthorization (preauthorization)
- 4 - authorization advice (preauthorisation completion, financial advice)
- 5 - reversal (cancellation)
- 6 - balance
- 7 - pin change
- 8 - tokenization

Type : enum (1, 2, 3, 4, 5, 6, 7, 8)

## User

User

Name	Description	Schema
<b>accessModes</b> <i>optional</i>		<a href="#">AccessMode</a>
<b>active</b> <i>optional</i>	User activity	boolean
<b>login</b> <i>optional</i>	User login <b>Maximal length</b> : 255	string
<b>password</b> <i>optional</i>	User password <b>Maximal length</b> : 255	string
<b>rights</b> <i>optional</i>		<a href="#">Rights</a>
<b>userId</b> <i>optional</i> <i>read-only</i>	User indentifier	integer

## Users

Table of users

Name	Schema
<b>users</b> <i>optional</i>	< <a href="#">User</a> > array

## UsersPagedResponse

*Polymorphism* : Composition

Name	Description	Schema
<b>limit</b> <i>optional</i>	Present limit	integer
<b>offset</b> <i>optional</i>	Offset	integer
<b>total</b> <i>optional</i>	Total count	integer
<b>users</b> <i>optional</i>		< <a href="#">User</a> > array

## Security

## BasicSecurity

Login and password based authorisation

*Type* : basic

## Bearer

*Type* : apiKey

*Name* : Authorization

*In* : HEADER