



## Alexela OCC Partners API

This document describes Alexela OCC's (online card center's) online data exchange protocol for Alexela partners.

### Table of contents

1. [API protocol](#)
  - a. [Test and production system](#)
  - b. [Authentication](#)
  - c. [Additional headers](#)
  - d. [Failed requests](#)
  - e. [Postman pre-request script](#)
2. [Payment card operations](#)
  - a. [Contract numbers request](#)
  - b. [Transactions request](#)
3. [Bonuscard operations](#)
  - a. [Define bonuscards request](#)
  - b. [Bonuscard transactions request](#)
4. [EV transactions operations](#)
  - a. [EV transactions request](#)

### API protocol

REST/JSON protocol are used for the API. Character set in all communication and data content must be in UTF-8 encoding. All dates and date times must be in UTC.

### Test and production system

Test service can be found: <https://test-partnerapi.alexela.ee/api/v1...>

Swagger: <https://test-partnerapi.alexela.ee/swagger/index.html>

Production service can be found: <https://partnerapi.alexela.ee/api/v1...>

Swagger: <https://partnerapi.alexela.ee/swagger/index.html>

### Authentication

Every request is authenticated with a calculated signature sent as a HTTP header that is based on the request, current time and pre-shared client key (PSK).

A canonical string is first created using HTTP headers containing the content-type, content MD5 sum, request URI, and the timestamp.

```
canonical_string = 'content_type,content_md5,request_uri,timestamp'
```

Described parts are joined by commas.

**content\_type:** Content Type used in request. Is always *application/json*

**content\_md5:** MD5 checksum of the request content as hexadecimal string.

**request\_uri:** Request uri for the given request starting with the */api/* root path component including the query string if applicable, e.g. */api/v1/getBalance*

**timestamp:** HTTP-date defined by RFC 2616

This string is then used to create the signature which is as Base64 encoded SHA256 HMAC calculated using the PSK.

The signature is then added as the **Authorization** HTTP header as follows

```
APIAuth-HMAC-SHA256 $client_access_id:$signature
```

On the server side, the SHA256 HMAC is computed in the same way using the request headers and the client's PSK, which is only known to the client and server but can be looked up on the server using the client's access id that is attached in the header. The access id is unique for each ERP and also identifies the calling ERP to the API.

Once generated the signature is valid for 5 minutes relative to API server time to avoid replay attacks so the client must ensure correct time is sent. The PSK must be at least 128 bytes of cryptographically secure random hex characters.

## Additional headers

In addition to the Authorization header, the **Date** header must be set with the same timestamp value that was used in the HMAC calculation.

## Failed requests

Any returned HTTP status code above or equal to 400 should be considered an error. All 4xx codes are client errors where the client must fix the request before resending it to the server. 5xx codes are server errors indicating that there is an error on the server side.

Example:

```
1 POST /api/v1/ContractTransactions HTTP/1.1
2 Content-Type: application/json
3 {
4   "contractNumbers": [
5     "string"
6   ],
7   "fromDate": "string",
8   "toDate": "string"
9 }
10
11 HTTP/1.1 400 Bad Request
12 Content-Type: application/json
13
14 {
15   "success":false,
16   "data":null,
17   "error":"Wrong Date format",
18   "message": null
19 }
```

## Postman pre-request script

```
1 var AppId = "your-api-id";
2 var APIKey = "your-api-key";
3
4 var currentDate = new Date();
5
6 var requestPath = "/" + pm.request.url.path.join("/");
7 var requestQueryString = pm.request.url.getQueryString();
8 var requestURI = requestPath + (!!requestQueryString ? '?' + requestQueryString : '');
9
10 var requestContentHexString = "";
11 if (pm.request.body) {
12     var md5 = CryptoJS.MD5(pm.request.body.toString());
13     requestContentHexString = md5.toString(CryptoJS.enc.Hex);
14 }
15
16 var signatureRawData = `application/json,${requestContentHexString},${requestURI},${currentDate.toUTCString()}`;
17 var signature = CryptoJS.enc.Utf8.parse(signatureRawData);
18 var signatureBytes = CryptoJS.HmacSHA256(signature, APIKey);
19 var requestSignatureBase64String = CryptoJS.enc.Base64.stringify(signatureBytes);
20 var hmacKey = "APIAuth-HMAC-SHA256 " + AppId + ":" + requestSignatureBase64String;
21 var hmacDate = currentDate.toUTCString();
22
23 pm.variables.set("hmacDate", hmacDate);
24 pm.variables.set("hmacKey", hmacKey);
25
26 console.log('uri', requestURI, 'string to tokenize', signatureRawData, 'hmac token', hmacKey);
```

Use variables `hmacDate` and `hmacKey` to set headers.

## Payment card operations

### Contract numbers request

GET /ContractNumbers

Operation allows to inquire contract numbers that are allowed to be seen by the specific API user.

Contract numbers – API user mappings are managed in Alexela OCC.

#### Response

| Field | Type          | Description              |
|-------|---------------|--------------------------|
|       | Array(string) | List of contract numbers |

```
1 [
2   "string"
3 ]
```

### Transactions request

POST /ContractTransactions or GET/ ContractTransactions

Operation enables to inquire payment card transactions data for the selected contracts and time period.

POST Request

| Field           | Type           | Description              |
|-----------------|----------------|--------------------------|
| contractNumbers | Array (string) | List of contract numbers |
| fromDate        | Datetime       | For example „2018-09-05“ |
| toDate          | Datetime       | For example „2018-09-06“ |

```
1 {
2   "contractNumbers": [
3     "string"
4   ],
5   "fromDate": "string",
6   "toDate": "string"
7 }
```

GET Request

*https://{baseUrl}/ContractTransactions?contractNumbers=22297,123,123&fromDate=2022-07-01&toDate=2022-07-07*

Response

| Field | Type                  | Description          |
|-------|-----------------------|----------------------|
|       | Array(transactionRow) | List of transactions |

TransactionRow:

| Field             | Type   | Description                                                              |
|-------------------|--------|--------------------------------------------------------------------------|
| transactionId     | String |                                                                          |
| terminalId        | String |                                                                          |
| stationId         | String |                                                                          |
| receiptNo         | String |                                                                          |
| authorizationCode | String | Value generated by payment terminal                                      |
| cardNumber        | String |                                                                          |
| currency          | String | EUR, SEK, PLN etc                                                        |
| occProductCode    | String |                                                                          |
| occProductName    | String |                                                                          |
| occProdyctType    | String | Balance transfer, Fuel, Insurance, Merchandise, Package, Service, Stamps |
| plateNumber       | String |                                                                          |

|                 |               |                                                |
|-----------------|---------------|------------------------------------------------|
| driverName      | String        |                                                |
| statusDate      | Datetime      |                                                |
| invoiceDate     | Datetime      | Exists after the transaction has been invoiced |
| invoiceNo       | String        | Exists after the transaction has been invoiced |
| transactionTime | Datetime      |                                                |
| quantity        | Decimal(18,4) |                                                |
| netPricePerUnit | Decimal(18,4) |                                                |
| discountPerUnit | Decimal(18,4) |                                                |
| pricePerUnit    | Decimal(18,4) |                                                |
| totalPrice      | Decimal(18,4) |                                                |
| vatRate         | Decimal(18,4) |                                                |
| vatAmount       | Decimal(18,4) |                                                |
| netAmount       | Decimal(18,4) |                                                |
| odometerReadout | Integer       |                                                |

```

1  [
2    {
3      "transactionId": "string",
4      "terminalId": "string",
5      "stationId": "string",
6      "receiptNo": "string",
7      "authorizationCode": "string",
8      "cardNumber": "string",
9      "currency": "string",
10     "occProductCode": "string",
11     "occProductName": "string",
12     "occProductType": "string"
13     "plateNumber": "string",
14     "driverName": "string",
15     "statusDate": "2020-02-26T14:02:36.994Z",
16     "invoiceDate": "2020-02-26T14:02:36.994Z",
17     "invoiceNo": "20810601-19091",
18     "transactionTime": "2020-02-26T14:02:36.994Z",
19     "quantity": 0,
20     "netPricePerUnit": 0,
21     "discountPerUnit": 0,
22     "pricePerUnit": 0,
23     "totalPrice": 0,
24     "vatRate": 0,
25     "vatAmount": 0,
26     "netAmount": 0,
27     "odometerReadout": 0
28   }
29 ]

```

## Possible error messages

- Wrong date format
- Only 60 days of search at one time is supported
- Only search over last 5 months of transactions is supported

## Bonuscard operations

### Define bonuscards request

POST /DefineBonuscards

Possible to add bonuscards to regular drivers group or premium drivers group. Based on the grouping, bonuscards will receive different discounts in Alexela stations.

Discounts for the groups are managed and controlled in Alexela OCC (discount settings, products, values, etc).

### Request

| Field             | Type           | Description                                                 |
|-------------------|----------------|-------------------------------------------------------------|
| bonuscards        | Array (string) | Define the list of regular bonuscards                       |
| premiumBonuscards | Array (string) | Define the list of premium bonuscards (with extra discount) |

```
1 {  
2   "bonuscards": [  
3     "string"  
4   ],  
5   "premiumBonuscards": [  
6     "string"  
7   ]  
8 }
```

### Response

In case of successful POST request – OK response.

### Bonuscard transactions request

POST /GetBonuscardTransactions

Operation enables to inquire transactions of partner bonuscards (the bonuscards that are previously defined with operation POST /DefineBonuscards).

### Request

| Field    | Type     | Description                    |
|----------|----------|--------------------------------|
| fromDate | Datetime | From, for example „2018-09-05“ |

|        |          |                              |
|--------|----------|------------------------------|
| toDate | Datetime | To, for example „2018-09-06“ |
|--------|----------|------------------------------|

## Response

| Field             | Type          | Description       |
|-------------------|---------------|-------------------|
| transactionId     | String        |                   |
| terminalId        | String        |                   |
| stationId         | String        |                   |
| receiptNo         | String        |                   |
| authorizationCode | String        |                   |
| cardNumber        | String        |                   |
| currency          | String        | EUR, SEK, PLN etc |
| occProductCode    | String        |                   |
| occProductName    | String        |                   |
| statusDate        | Datetime      |                   |
| transactionTime   | Datetime      |                   |
| quantity          | Decimal(18,4) |                   |
| netPricePerUnit   | Decimal(18,4) |                   |
| discountPerUnit   | Decimal(18,4) |                   |
| pricePerUnit      | Decimal(18,4) |                   |
| totalPrice        | Decimal(18,4) |                   |
| vatRate           | Decimal(18,4) |                   |
| vatAmount         | Decimal(18,4) |                   |
| netAmount         | Decimal(18,4) |                   |
| plateNumber       | String        |                   |
| driverName        | String        |                   |
| odometerReadout   | Integer?      |                   |

```

1  [
2    {
3      "transactionId": "string",
4      "terminalId": "string",
5      "stationId": "string",
6      "receiptNo": "string",
7      "authorizationCode": "string",
8      "cardNumber": "string",
9      "currency": "string",
10     "occProductCode": "string",
11     "occProductName": "string",

```

```

12     "statusDate": "2020-02-26T14:21:10.558Z",
13     "transactionTime": "2020-02-26T14:21:10.558Z",
14     "quantity": 0,
15     "netPricePerUnit": 0,
16     "discountPerUnit": 0,
17     "pricePerUnit": 0,
18     "totalPrice": 0,
19     "vatRate": 0,
20     "vatAmount": 0,
21     "netAmount": 0,
22     "plateNumber": "string",
23     "driverName": "string",
24     "odometerReadout": 0
25 }
26 ]

```

### Possible error messages

- Wrong date format
- Only 60 days of search at one time is supported
- Only search over last 5 months of transactions is supported

## EV transactions operations

### EV transactions request

GET /EvTransactions?fromDate=2023-12-01&toDate=2023-12-31

Operation enables to inquire EV transactions for specified date range. EV transactions query must be specifically enabled for a partner.

### Response

| Field | Type                    | Description             |
|-------|-------------------------|-------------------------|
|       | Array(EvTransactionRow) | List of EV transactions |

EvTransactionRow:

| Field                | Type     | Description                                                                                |
|----------------------|----------|--------------------------------------------------------------------------------------------|
| transactionId        | Integer  |                                                                                            |
| stationId            | Integer  |                                                                                            |
| chargerId            | String   | Specific machine that was used for charging                                                |
| transactionStatus    | String   | The status of the transaction.<br>Possible values: "confirmed", "unconfirmed", "cancelled" |
| transactionTime      | DateTime |                                                                                            |
| transactionStartTime | DateTime | Start of charging                                                                          |
| quantity             | Decimal  | kWh                                                                                        |



|                 |          |                                                                |
|-----------------|----------|----------------------------------------------------------------|
| connectorNumber | Integer? | The connector number of the charger that was used for charging |
|-----------------|----------|----------------------------------------------------------------|

### Possible error messages

- Wrong date format
- Only 60 days of search at one time is supported
- Only search over last 5 months of transactions is supported