FleetAPI.IntegrationAPI

Overview

This document define interface for partners to integrate with fleetAPI.

Changes history

Date	Version	Changer	Changes
2020-03-03	1.0.0	Paweł Gamrot	Initial version
2020-03-31	1.0.1	Ireneusz Olchawski	Status endpoint excluded from authorization
2020-04-01	1.1.0	Paweł Gamrot	User must have special right: IntegrationAccess to use this API Data for status endpoint is cached
2020-04-01	1.1.0	Ireneusz Olchawski	Added IntegrationAccess to enpoints
2020-04-01	1.1.0	Ireneusz Olchawski	Added caching for status endpoint
2020-05-11	1.2.0	Paweł Gamrot	Add mode parameter to endpoint `/terminals/accesscryptogram`
2020-05-19	1.3.0	Paweł Gamrot	Add `terminalId` field to `AccessCryptogram` object
2020-06-01	1.4.0	Ireneusz Olchawski	Add multi hsm access
2020-06-08	1.4.0	Maciej Wiosło	Add `includeGitInfo` field to status endpoint
2020-12-17	1.5.0	Paweł Gamrot	Add `fleets` field to `AccessCryptogram` object

HTTP status codes

The HTTP standard provides over 70 status codes to describe the return values. We don’t need them all, but there should be used at least a mount of 10.

- 200 – OK – Eyerything is working
- 201 – OK – New resource has been created
- 204 – OK – The resource was successfully deleted
- 304 – Not Modified – The client can use cached data

- 400 – Bad Request – The request was invalid or cannot be served. The exact error should be explained in the error payload. E.g. „The JSON is not valid“
- 401 – Unauthorized – The request requires an user authentication
- 403 – Forbidden – The server understood the request, but is refusing it or the access is not allowed.
- 404 – Not found – There is no resource behind the URI.
- 422 – Unprocessable Entity – Should be used if the server cannot process the enitity, e.g. if an image cannot be formatted or mandatory fields are missing in the payload.
- 500 – Internal Server Error – API developers should avoid this error. If an error occurs in the global catch blog, the stracktrace should be logged and not returned as response.

Version information

Version : 1.5.0

Paths

Authenticates user and returns token data

```
POST /v1/authentication/login
```

Description

Allowed for:

- Everyone

Parameters

Type	Name	Description	Schema
Body	**loginModel** *optional*	Object containing authentication data	Login

Responses

HTTP Code	Description	Schema
200	Success	Authentication
401	Unauthorized	No Content
500	Internal server error	No Content

Consumes

- `application/json-patch+json`
- `application/json`
- `text/json`
- `application/*+json`

Produces

- `application/json`

Tags

- Authentication

Refreshes user's tokens

```
GET /v1/authentication/refreshTokens
```

Description

Allowed for:

- Everyone

Responses

HTTP Code	Description	Schema
200	Success	Authentication
400	Bad request	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not found	No Content
500	Internal server error	No Content

Produces

- `application/json`

Tags

- Authentication

Security

Type	Name
apiKey	**Bearer**

Return terminal access cryptogram

```
GET /v1/terminals/accesscryptogram
```

Parameters

Type	Name	Description	Schema
Query	**mode** *optional*	Result type mode. Available values: * 1 - only AccessCryptogram.cryptogram field is returned * 2 - only AccessCryptogram.userCryptogram and AccessCryptogram.passwordCryptogram fields are returned Parameter not set means default value which is 1.	enum (1, 2)
Query	**tid** *required*	Terminal identifier assigned by the partner (TID)	string

Responses

HTTP Code	Description	Schema
200	OK	AccessCryptogram
400	Bad request	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not found	No Content
500	Internal server error	No Content

Produces

- `application/json`

Tags

- Terminals

Security

Type	Name
apiKey	**Bearer**

Returns application modules status info

```
GET /v1/utilities/status
```

Parameters

Type	Name	Description	Schema	Default
Query	**includeGitInfo** *optional*	Git repository info (default is False)	boolean	`"false"`

Responses

HTTP Code	Description	Schema
200	Success	ApiStatusModel
400	Bad request	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not found	No Content
500	Internal server error	No Content

Produces

- `application/json`

Tags

- Utilities

Security

Type	Name
apiKey	**Bearer**

Definitions

AccessCryptogram

Terminal access cryptogram data. Based on mode parameter from endpoint `/terminals/accesscryptogram` can be returned in two formats: * 1 - only cryptogram field * 2 - userCryptogram and passwordCryptogram fields

Name	Description	Schema

Name	Description	Schema
cryptogram *optional*	Encrypted user and terminal password (encrypted access string in base64 format). Access string is concatenation of terminal transactionUser(filled to 40 characters with 0x00) and terminal transactionPassword(filled to 40 characters with 0x00). Encryption is done using 3DES in CBC mode with secret key(double lenght 3DES key)	string
fleets *optional*	Array includes list of active fleets for terminal	< string > array
passwordCryptogram *optional*	Encrypted terminal password (encrypted password access string in base64 format). Password access string is terminal transactionPassword(filled to 40 characters with 0x00). Encryption is done using 3DES in CBC mode with secret key(double lenght 3DES key)	string
terminalId *optional*	Terminal identifier assigned by the fleetAPI	integer (int32)
userCryptogram *optional*	Encrypted terminal user (encrypted user access string in base64 format). User access string is terminal transactionUser(filled to 40 characters with 0x00). Encryption is done using 3DES in CBC mode with secret key(double lenght 3DES key)	string

ApiStatusModel

Name	Schema
componentsStatuses *optional*	< ComponentStatusModel > array
git *optional*	GitInfoModel
startupDate *optional*	string (date-time)
version *optional*	string

Authentication

Authentication model - response for login action

Name	Description	Schema

Name	Description	Schema
authorizationToken *optional*	Gets or Sets AuthorizationToken	Token
loggedUserContext *optional*	Gets or Sets LoggedUserContext	LoggedUserContext
refreshToken *optional*	Gets or Sets RefreshToken	Token

ComponentStatusModel

Name	Schema
name *optional*	string
requestDate *optional*	string (date-time)
responseTime *optional*	string
status *optional*	string
type *optional*	string

GitInfoModel

Name	Schema
branch *optional*	string
hash *optional*	string

LoggedUserContext

Logged user's data

Name	Description	Schema
login *optional*	User's login	string
userId *optional*	User's identifier	integer (int32)

Login

User login data

Name	Description	Schema
password *optional*	Gets or Sets Password	string
username *optional*	Gets or Sets Username	string

Token

Authorization token's data

Name	Description	Schema
expires *optional*	Token's expiration date	string (date-time)
token *optional*	Token	string

Security

Bearer

Type : apiKey
Name : Authorization
In : HEADER