Communication Protocol Between Pedelec Station and Central Management System

Site:
github.com/RWTH-i5-IDSG/ps-cms-protocol

Authors:
RWTH - i5

Version:
0.0.23

Date:
21.12.2015

Version History

Version	Date	Description
0.0.1	01.07.2014	Initial draft
0.0.2	03.07.2014	Added Update Firmware (by CMS) and Firmware Status Notification (by PS) Added Change & Get Pedelec Configuration (by CMS)
0.0.3	08.07.2014	Deleted maxBatteryRange from Charging Status Notification (by PS)
0.0.4	10.07.2014	Added Upload Logs (by CMS) and Logs Status Notification (by PS) Added more parameters to Charging Status Notification (by CMS)
0.0.5	18.08.2014	Updated station pedelec URLs in section 3
0.0.6	25.08.2014	Added ChangeState Type
0.0.7	06.10.2014	Added the operations Reserve Now and Cancel Reservation
0.0.8	15.10.2014	Changed in "Start Transaction" parameter from 'userId' to 'cardId'
0.0.9	24.10.2014	Send 'cardId' instead of 'userId' after Authorize Request
0.0.10	03.11.2014	Add Card Activation API
0.0.11	27.11.2014	Add "Get Available Pedelecs" API and update "Card Activation" API
0.0.12	12.01.2015	Change request param in "Remote Authorize"; replacing "remainingTrials" in "Authorize" with error 403
0.0.14	04.03.2015	In 2.7 & 2.8 renamed to "cardPin", added types-column, changed "userId" to "cardId" in 3.14; "pedelecManufacturerId" in 2.1 and "errorcode"/"errorinfo" in 2.2/2.3 are now optional; removed "slotManufacturerId" in 2.4
0.0.15	06.03.2015	Added EncryptionKeys in 2.1 BootNotification and checksum in 3.12 Update Firmware
0.0.16	10.03.2015	Renamed values in chapter 4 - Types and errorCode + errorInfo are optional
0.0.17	24.04.2015	Added stationURL in BootNotification and accountState in Authorize (Chapter 2)
0.0.18	24.06.2015	Simplified return object of get available pedelecs
0.0.19	02.07.2015	Added CardId Param to *2.11 Get Available Pedelecs* for filtering user-reservations

Version	Date	Description
0.0.20	21.07.2015	Added charging state NOT_CHARGING
0.0.21	28.07.2015	Changed BootNotification response structure to contain multiple read/write keys
0.0.22	14.10.2015	Removed HTTP error codes (errors are always HTTP-400 with error message, *(4.1)*, successful responses are always *200 OK*; *2.7* added *actualRentedCount* and *canRentCount*; 3.14 & 3.15 exchanged reservationId with pedelecId
0.0.23	21.12.2015	Removed WriteKey in Bootnotification Response. Added WriteKey, ReadKey and ApplicationKey into ActivateCard Response (subsection 2.8). Added Card Activation Notification as indicator for a (un-)successful activation (subsection 2.9)

Contents

1 Introduction **7**

1.1 Use Cases 7

1.2 Technology 9

2 Operations Initiated by Pedelec Station **10**

2.1 Boot Notification 10

2.2 Station Status Notification 10

2.3 Pedelec Status Notification 11

2.4 Charging Status Notification 11

2.5 Firmware Status Notification 12

2.6 Logs Status Notification 12

2.7 Authorize 12

2.8 Activate Card 13

2.9 Card Activation Notification 14

2.10 Start Transaction 14

2.11 Stop Transaction 14

2.12 Get Available Pedelecs 15

2.13 Heartbeat 15

3 Operations Initiated by Central Management System **16**

3.1 Change Station Operation State 16

3.2 Change Pedelec Operation State 16

3.3 Change Pedelec Configuration 16

3.4 Change Station Configuration 17

3.5 Get Station Configuration 17

3.6 Get Charging Status 18

3.7 Get Pedelec Configuration 18

3.8 Remote Authorize 18

3.9 Cancel Authorize 19

3.10 Reboot 19

3.11 Unlock Slot 19

3.12 Update Firmware 19

3.13 Upload Logs 20

3.14 Reserve Now 20

3.15 Cancel Reservation 21

4 Types **22**

4.1 Error Message Template 22

4.2 Operation State 22

4.3 Charging State 22

4.4 Firmware Update State 22

4.5 Logs Update State 22

4.6 Configuration Error Reason 23

Acronyms

PS Pedelec Station

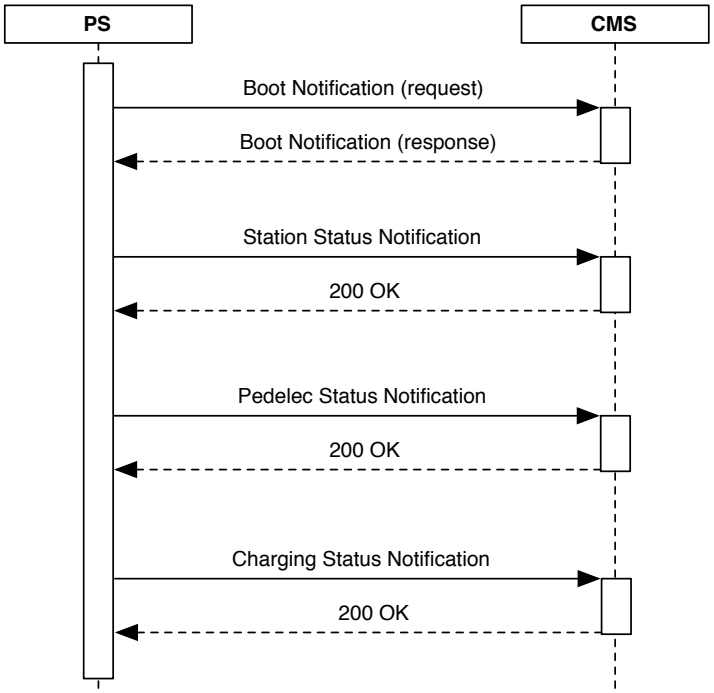
CMS Central Management System

1 Introduction

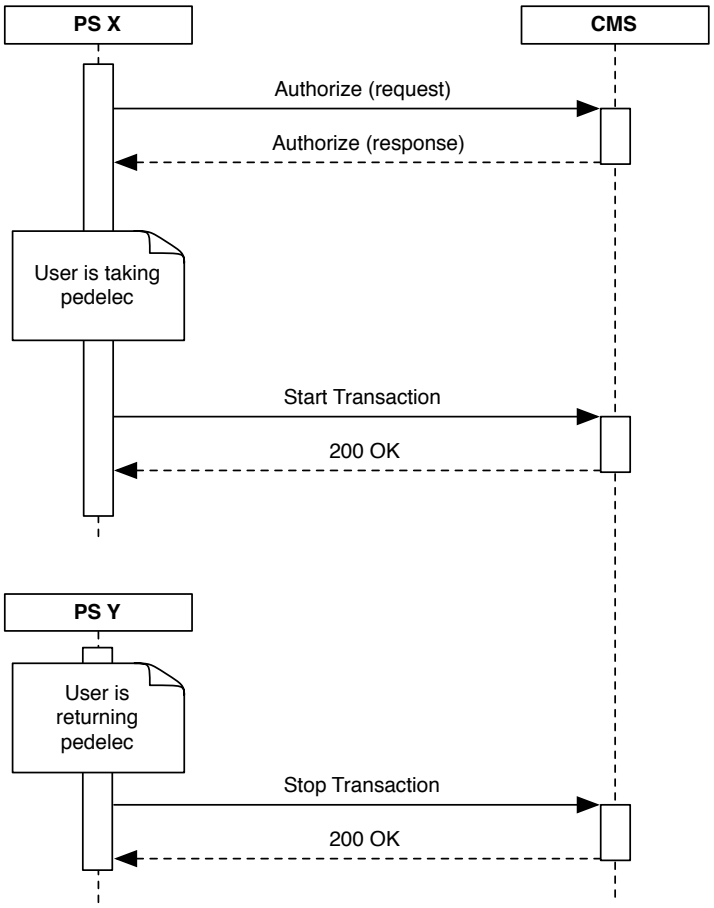
// TODO

1.1 Use Cases

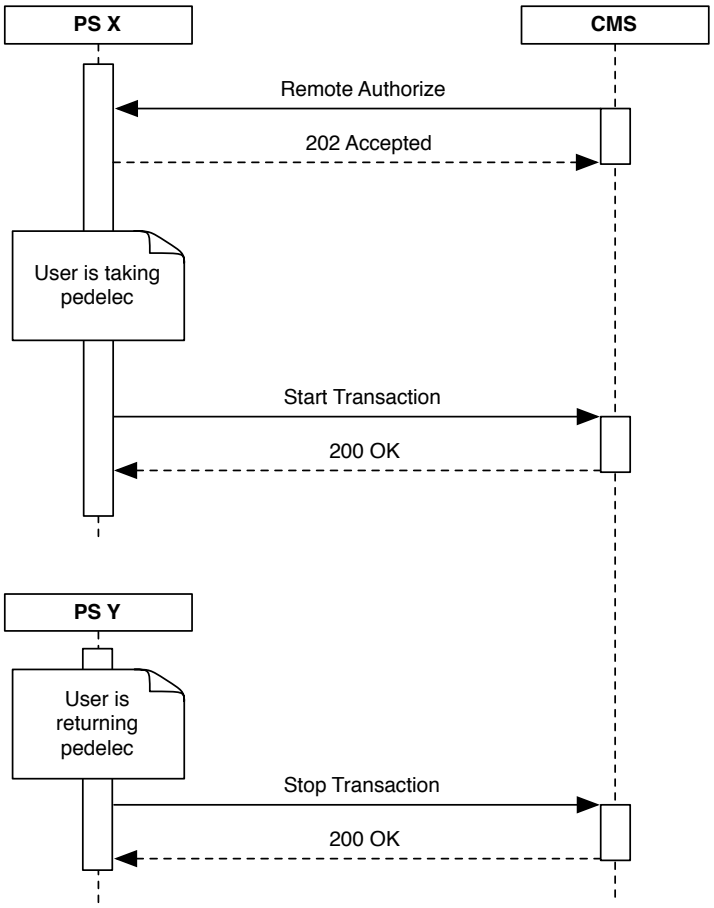
1.1.1 Station Boot



1.1.2 Rent Bike with Card



1.1.3 Rent Bike with App



1.2 Technology

The protocol is designed to be implemented as a RESTful Webservice with HTTP as the underlying data transfer protocol. The resources are represented in JSON data format.

We require that all communications are done encrypted, e.g. using SSL/TLS or VPN.

2 Operations Initiated by Pedelec Station

2.1 Boot Notification

Description: After start-up of a PS, the PS sends a notification to the CMS with information about its configuration (e.g., manufacturer id, connected station slots and pedelecs). CMS will accept only registered stations.

After each reboot, the Boot Notification is sent.

The CMS sends a response with the acceptable status, including current time and heartbeat interval if accepted.

The PS repeats the Boot Notification (in an appropriated interval) until the CMS accepts the PS. The PS requests nothing else, until CMS accepts it.

URL: `[BASE_CMS_URI]/boot`

Method: POST

Request:

Field	Type	Description
stationManufacturerId	String	This value identifies the PS by its hardware serial
firmwareVersion	String	Firmware version of PS
slots/slotManufacturerId	String	This value identifies the slot by its hardware serial
slots/slotPosition	Integer	The sequence position of the connected slot
slots/pedelecManufacturerId	String (optional)	This value identifies the locked pedelec by its hardware serial related to the slot
stationURL	String	URL to communicate with PS initiated by CMS

Response: **200 OK**

Field	Type	Description
timestamp	Long	Unix timestamp (seconds since epoch)
heartbeatInterval	Integer	In seconds
cardKeys/name	String	Name of the key pair for identification
cardKeys/readKey	String	Encryption key to read out user's cardId

Errors: **NOT_REGISTERED - The station ({id}) is not registered**

5xx Server Error

2.2 Station Status Notification

Description: A PS sends a notification to the CMS to inform the CMS about its status or error condition within the PS including the connected station slots. A PS shall send a Station Status Notification when it becomes unavailable as a result of an error condition or other external events.

URL: `[BASE_CMS_URI]/status/station`

Method: POST

Request:

Field	Type	Description
stationManufacturerId	String	This value identifies the PS by its hardware serial
stationErrorCode	String	Required when stationState is INOPERATIVE
stationErrorInfo	String	Required when stationState is INOPERATIVE
stationState	String	See Section 4.2
timestamp	Long	Unix timestamp (seconds since epoch)
slots/slotManufacturerId	String	This value identifies the slot by its hardware serial
slots/slotErrorCode	String	Required when slotState is INOPERATIVE
slots/slotErrorInfo	String	Required when slotState is INOPERATIVE
slots/slotState	String	See Section 4.2

Response: **200 OK**

Errors: **5xx Server Error**

2.3 Pedelec Status Notification

Description: A PS sends a notification to the CMS to inform the CMS about the status or error condition of connected pedelecs. A PS shall send an Pedelec Status Notification when a pedelec becomes unavailable as a result of an error condition or other external events.

URL: `[BASE_CMS_URI]/status/pedelec`

Method: POST

Request:

Field	Type	Description
pedelecManufacturerId	String	This value identifies the Pedelec by its hardware serial
pedelecErrorCode	String	Required when pedelecState is INOPERATIVE
pedelecErrorInfo	String	Required when pedelecState is INOPERATIVE
pedelecState	String	See Section 4.2
timestamp	Long	Unix timestamp (seconds since epoch)

Response: **200 OK**

Errors: **5xx Server Error**

2.4 Charging Status Notification

Description: The PS informs the CMS at regular intervals about the charging status (in

time intervals or when fully charged) of all connected pedelecs of the PS.

The message contains the current timestamp, the meter value (Wh), the charging state (e.g., charging, completed), the and PedelecID.

URL: `[BASE_CMS_URI]/status/charging`

Method: POST

Request:

Field	Type	Description
pedelecManufacturerId	String	
timestamp	Long	Unix timestamp (seconds since epoch)
chargingState	String	See Section 4.3
meterValue	Double	
battery/soc	Double	percentage points
battery/temperature	Double	
battery/cycleCount	Integer	
battery/voltage	Double	
battery/current	Double	

Response: **200 OK**

Errors: **5xx Server Error**

2.5 Firmware Status Notification

Description: A PS notifies CMS about the success/failure of the firmware update.

URL: `[BASE_CMS_URI]/status/firmware`

Method: POST

Request:

Field	Type	Description
status	String	Progress status of the firmware update; see Section 4.4

Response: **200 OK**

Errors: **5xx Server Error**

2.6 Logs Status Notification

Description: The PS informs the CMS about the status of requested uploading of logs.

URL: `[BASE_CMS_URI]/status/logs`

Method: POST

Request:

Field	Type	Description
status	String	Upload status of the logs; see Section 4.5

Response: **200 OK**

Errors: **5xx Server Error**

2.7 Authorize

Description: Before a user can choose and unlock a pedelec with his CustomerCard (e.g., Bluecard), the PS needs to be able to authorize the operation. Only after

authorization the PS will be able to unlock the pedelec. For this purpose the PS needs user's Card-ID and PIN for authorization.

The response shall indicate, whether or not the Card-ID and PIN combination is accepted by the CMS and how many pedelecs are currently rented and can be rented by the customer.

URL: `[BASE_CMS_URI]/authorize`

Method: POST

Request:

Field	Type	Description
cardId	String	Card specific number
cardPin	String	User's secret PIN

Response: **200 OK** (If credentials are accepted)

Field	Type	Description
cardId	String	Card specific number
actualRentedCount	Integer	Number of pedelecs rented by customer
canRentCount	Integer	Number of pedelecs which can be rented

Errors: **AUTH_ATTEMPTS_EXCEEDED - no trials remaining and account gets disabled**

NOT_REGISTERED - card account is unknown

CONSTRAINT_FAILED - card account is disabled

CONSTRAINT_FAILED - wrong pin

5xx Server Error

2.8 Activate Card

Description: Before a user can use his card (e.g., Bluecard) to rent a bike, he has to activate it on the PS terminal. For this purpose, the PS sends Activation-Key and PIN to the CMS.

The response shall indicate, whether or not the Activation-Key and PIN are accepted by the CMS and responses with the CardId.

URL: `[BASE_CMS_URI]/activate-card`

Method: POST

Request:

Field	Type	Description
activationKey	String	Key to start activation process to initial user's card
cardPin	String	PIN for customer's card

Response: **200 OK** (If credentials are accepted)

Errors: **CONSTRAINT_FAILED - Credentials are not accepted**

5xx Server Error

Field	Type	Description
cardId	String	Card specific number
applicationKey	String	Encryption key to write read and write keys
readKey	String	Encryption key to read user's cardId
writeKey	String	Encryption key to write user's cardId

2.9 Card Activation Notification

Description: The PS informs the CMS about a (un-)successful card activation related to subsection 2.8.

URL: `[BASE_CMS_URI]/activate-card`

Method: POST

Request:

Field	Type	Description
successfulActivation	Boolean	Indicated the state of the activation process
cardId	String	Card specific number

Response: **200 OK** (If credentials are accepted)

Errors: **5xx Server Error**

2.10 Start Transaction

Description: When the rental is authenticated, the station slot is unlocked and the user took the pedelec out of the slot, the PS needs to inform the CMS about this.

As response the CMS sends an acknowledgment.

URL: `[BASE_CMS_URI]/transaction/start`

Method: POST

Request:

Field	Type	Description
cardId	String	
pedelecManufacturerId	String	
stationManufacturerId	String	
slotManufacturerId	String	
timestamp	Long	Unix timestamp (seconds since epoch)

Response: **200 OK**

Errors: **5xx Server Error**

2.11 Stop Transaction

Description: After the PS recognizes the return of a pedelec at a station slot, it needs to inform the CMS about this.

As response the CMS sends an acknowledgment.

URL: `[BASE_CMS_URI]/transaction/stop`

Method: POST

Request:

Field	Type	Description
pedelecManufacturerId	String	
stationManufacturerId	String	
slotManufacturerId	String	
timestamp	Long	Unix timestamp (seconds since epoch)

Response: **200 OK**

Errors: **5xx Server Error**

2.12 Get Available Pedelecs

Description: PS can retrieve a list of available pedelecs ordered by a predefined priority or a specific pedelec for a user's reservation when available for this moment.

URL: `[BASE_CMS_URI]/available-pedelecs?cardId=ab34-cd56`

Method: GET

Request:

Field	Type	Description
cardId	String	User's card ID

Response: **200 OK**

Type	Description
String[]	List of manufacturer Ids for available pedelecs

Errors: **5xx Server Error**

2.13 Heartbeat

Description: To let the CMS know that a station is still connected, a PS sends heartbeats regularly in configurable time intervals.

The CMS sends a response with the current time of the CMS, which could be used to synchronize the time of the PS with the time of the CMS.

URL: `[BASE_CMS_URI]/heartbeat`

Method: GET

Request: -

Response: **200 OK**

Field	Type	Description
timestamp	Long	Unix timestamp (seconds since epoch)

Errors: **5xx Server Error**

3 Operations Initiated by Central Management System

3.1 Change Station Operation State

Description: CMS can request to change the operation state of a PS or its slots. The PS can accept or reject the process the request. When rejected, the PS must include a reason.

URL: `[BASE_PS_URI]/state`

Method: POST

Request:

Field	Type	Description
slotPosition (optional)	Integer	When present, the state of the slot with the given position will be changed. When absent, the state of whole PS will be changed.
state	See Section 4.2	

Response: **200 OK**

Errors: **CONSTRAINT_FAILED**

5xx Server Error

3.2 Change Pedelec Operation State

Description: CMS can request to change the operation state of a pedelec located at a slot of a PS. The PS can accept or reject the process the request. When rejected, the PS must include a reason.

URL: `[BASE_PS_URI]/pedelecs/<pedelecManufacturerId>/state`

Method: POST

Request:

Field	Type	Description
slotPosition	Integer	The position of the slot where the pedelec is located.
pedelecState	String	See Section 4.2

Response: **200 OK**

Errors: **CONSTRAINT_FAILED**

5xx Server Error

3.3 Change Pedelec Configuration

Description: CMS can request a PS to change specific Pedelec configuration parameters. This request contains a list of key-value pairs, where "key" is the name of the configuration setting to change and "value" contains the new setting for the configuration setting.

URL: `[BASE_PS_URI]/pedelecs/<pedelecManufacturerId>/config`

Method: POST

Request:

Field	Type	Description
maxCurrentValue	Double	..
maxBatteryLevel	Double	..

Response: **200 OK** (If all the parameter changes are accepted and done)

Errors: **400 Bad Request**

Field	Type	Description
failed	...	List of parameters that PS failed to set a new value for.
reason	String	See Section 4.6

5xx Server Error

3.4 Change Station Configuration

Description: CMS can request a PS to change configuration parameters. This request contains a list of key-value pairs, where "key" is the name of the configuration setting to change and "value" contains the new setting for the configuration setting.

URL: `[BASE_PS_URI]/config`

Method: POST

Request:

Field	Type	Description
cmsURI	String	New value for the CMS Webservice URI
heartBeatInterval	Integer	In seconds
openSlotTimeout	Integer	In seconds. How long should PS wait after unlocking a slot before locking it again.
rebootRetries	Integer	How many times should PS try to reboot before giving up.
chargingStatusInformInterval	Integer	In seconds

Response: **200 OK** (If all the parameter changes are accepted and done)

Errors: **400 Bad Request**

Field	Type	Description
failed	..	List of parameters that PS failed to set a new value for.
reason	String	See Section 4.6

5xx Server Error

3.5 Get Station Configuration

Description: CMS can retrieve the values of configuration settings. This operation requires no parameters, and PS returns all values.

URL: `[BASE_PS_URI]/config`

Method: GET

Request: -

Response: **200 OK**

JSON object with return values for elements defined in Section 3.4/Request

Errors: **5xx Server Error**

3.6 Get Charging Status

Description: Even though a PS informs the CMS about the charging status of pedelecs regularly, it is desirable to get the latest information in various cases.

URL: `[BASE_PS_URI]/charging-status`

Method: GET

Request: -

Response: **200 OK**

JSON object with return values for elements defined in Section 2.4/Request

Errors: **5xx Server Error**

3.7 Get Pedelec Configuration

Description: CMS can retrieve the values of configuration settings. This operation requires no parameters, and PS returns all values.

URL: `[BASE_PS_URI]/pedelecs/<pedelecManufacturerId>/config`

Method: GET

Request: -

Response: **200 OK**

JSON object with return values for elements defined in Section 3.3/Request

Errors: **5xx Server Error**

3.8 Remote Authorize

Description: When using the mobile app for renting a pedelec the user does not require a card to authenticate against the PS, but uses the app to authenticate directly against the CMS. In this case, CMS sends a Remote Authorize message to the PS to unlock the slot(s) for the user to take the pedelec.

After a timeout period PS checks the existence of a pedelec at the slot(s) and sends a Start Transaction message to CMS, namely the rental process proceeds usual.

URL: `[BASE_PS_URI]/authorize/remote`

Method: POST

Request:

Field	Type	Description
slotPosition	Integer	
cardId	String	

Response: **200 OK**

Errors: **CONSTRAINT_FAILED**

5xx Server Error

3.9 Cancel Authorize

Description: When using the mobile app for renting a pedelec the user can wish to cancel the rental process after Remote Authorize is initiated. In this case, CMS sends a Cancel Authorize message to PS to lock the slot(s) again.

URL: `[BASE_PS_URI]/authorize/cancel/<slotPosition>`

Method: POST

Request: -

Response: **200 OK**

Errors: **CONSTRAINT_FAILED**

5xx Server Error

3.10 Reboot

Description: CMS can request a PS to reboot. When accepted, the PS reboots after gracefully terminating running software. When rejected, the PS must include a reason.

URL: `[BASE_PS_URI]/reboot`

Method: POST

Request: -

Response: **200 OK**

Errors: **CONSTRAINT_FAILED**

5xx Server Error

3.11 Unlock Slot

Description: In cases of maintenance or technical problems CMS can request a PS to unlock a slot or all slots in order to access a pedelec.

URL: `[BASE_PS_URI]/unlock/<slotPosition>`
slotPosition is optional. When absent, the PS unlocks all slots.

Method: POST

Request: -

Response: **200 OK**

Errors: **CONSTRAINT_FAILED**

5xx Server Error

3.12 Update Firmware

Description: The CMS can send the PS a firmware update command. With a firmware update request, the CMS informs the PS about a new firmware, including the location of the firmware and the date & time when the update shall be executed.

The PS should start as soon as possible the firmware after retrieving the retrieve-update.

URL: `[BASE_PS_URI]/update-firmware`

Method: POST

Request:

Field	Type	Description
firmwareUpdateUrl	String	Location of the firmware update
executionDateTime	Long	Date and time when the PS should execute update, UNIX timestamp
checksum	String	

Response: **200 OK**

Errors: **5xx Server Error**

3.13 Upload Logs

Description: In order to diagnose errors, the CMS can request the PS to upload its locally stored log files to a remote directory. The fields oldestLogTimestamp and latestLogTimestamp define a date/time range for the requested logs. When both absent, PS uploads all logs.

URL: `[BASE_PS_URI]/upload-logs`

Method: POST

Request:

Field	Type	Description
logDirectoryUrl	String	Directory for the logs to be uploaded to
oldestLogTimestamp (optional)	Long	UNIX timestamp
latestLogTimestamp (optional)	Long	UNIX timestamp

Response: **200 OK**

A JSON array with the file names of the logs to be uploaded.

Errors: **CONSTRAINT_FAILED - Logs for the requested time window do not exist**

5xx Server Error

3.14 Reserve Now

Description: A customer can reserve a specefic pedelec for a certain time. For this purpose, the CMS sends a reservation message with a pedelecId and an expiry date to the PS.

URL: `[BASE_PS_URI]/reserve-now`

Method: POST

Request:

Field	Type	Description
pedelecId	String	Id to identify reserved pedelec on CMS and PS side
cardId	String	Customer identification
expiryDate	Long	UNIX timestamp which identifies the end of the reservation

Response: **200 OK**

A reservation was successfully added to the PS

Errors:

PEDELEC_MISSING

When pedelec is not available.

5xx Server Error

3.15 Cancel Reservation

Description: A customer or the CMS can cancel a reservation. For this purpose, the CMS sends a cancel reservation message with the pedelecId to the PS which holds the reserved pedelec.

URL: `[BASE_PS_URI]/cancel-reservation`

Method: POST

Request:

Field	Type	Description
pedelecId	String	Id to identify reserved pedelec on CMS and PS side

Response: **200 OK**

A reservation was successfully removed.

Errors:

CONSTRAINT_FAILED - pedelec not found CONSTRAINT_FAILED - reservation not found

When the reservation was not found.

5xx Server Error

4 Types

4.1 Error Message Template

In case an error occurs, send an HTTP 400 (Bad Request) which *always* responses with a JSON object containing following fields:

Field	Description
timestamp	Unix timestamp (seconds since epoch)
code	Internal, application-specific error code
message	Additional explanation

Error Codes:

- AUTH_ATTEMPTS_EXCEEDED
- CONSTRAINT_FAILED
- DATABASE_OPERATION_FAILED
- HARDWARE_MALFUNCTION
- NOT_REGISTERED
- PEDELEC_MISSING
- PEDELEC_FOUND
- UNKNOWN_SERVER_ERROR

4.2 Operation State

Value	Description
OPERATIVE	When the item is functional and working and ready to serve
INOPERATIVE	When the item is faulted and cannot be used

4.3 Charging State

Value	Description
CHARGING	When the battery is charging
COMPLETED	When the charging process is completed
NOT_CHARGING	Neither charging nor completed

4.4 Firmware Update State

Value	Description
DOWNLOAD_FAILED	PS failed to load firmware
INSTALLATION_FAILED	Installation of firmware failed
INSTALLED	Firmware is successfully installed in PS

4.5 Logs Update State

Value	Description
UPLOADED	
UPLOAD_FAILED	

4.6 Configuration Error Reason

Value	Description
NotAcceptable	If the request for some keys could not be processed. The server returns a JSON array of keys that are rejected (in this case other parameters are set)
NotFound	If some of the keys are not found/supported. The server returns a JSON array of keys that are not found as configuration parameters (in this case other parameters are set)