OCPI v2.1.1 Protocol

IOP – OCPI Interface - Implementation Guide V1.2.10

GIREVE

21/09/2026

1 INTRODUCTION 8
1.1 AIMS 8
1.2 INTENDED AUDIENCE 8
1.3 DEFINITIONS AND ABBREVIATIONS 8
1.4 CARDINALITY EXPRESSION 8
2 INTEGRATION GUIDELINES 9
2.1 TECHNICAL 9
2.1.1 Supported OCPI versions 9
2.1.2 Security 9
2.1.3 Client owned object push 9
2.1.4 Pagination 10
2.1.5 IOP HTTP headers 10
2.1.6 Anti-spam protection 11
2.2 OCPI MODULES IMPLEMENTED BY IOP 12
2.3 GIREVE MANAGEMENT OF LOCATIONS DATA 12
2.4 ROAMING 12
2.4.1 General workflow 12
2.4.2 New attribute « authorization_id » 14
2.4.3 Management of B2B tariffs 16
2.4.4 RFID Tokens 16
2.4.5 Custom OCPI flow to prevent eMSP Tokens download by CPOs 16
3 CPO SPECIFIC IMPLEMENTATION GUIDELINES 19
3.1 CPO OPERATION DEFINITION AND NAMING RULES 19
3.2 CPO OPERATION AND ROAMING OFFERS 19
3.3 USE CASES COVERED BY IOP 19
3.3.1 Technical Use cases 20
3.3.2 EVCI data 20
3.3.3 Roaming 20
3.4 USE CASES REQUIRED BY GIREVE 21
3.4.1 Always required 21
3.4.2 If the CPO implements the “Roaming” feature 22
3.4.3 If the CPO doesn’t commit and describe its tariffs in a roaming agreement 22
3.5 CONNECTION & REGISTER SPECIFICATIONS 22
3.5.1 Update Credentials FromIOP 22
3.6 LOCATIONS MODULE SPECIFICATIONS 22
3.6.1 Static and dynamic attributes 23
3.6.2 EVSE object 23
3.6.3 Connector object 23
3.6.4 Tariff_id value 23
3.6.5 Plug&Charge (P&C): Connector Object - new attribute “capabilities” 23
3.6.6 PUSH Locations ToIOP 24
3.6.7 PULL Locations ToIOP 24
3.6.8 PULL Locations FromIOP 24
3.7 TOKENS MODULE SPECIFICATIONS 25
3.7.1 GET Tokens To/FromIOP 25
3.7.2 PULL Tokens ToIOP : Get List Pagination 25
3.7.3 PULL Tokens ToIOP: Get List, Full and Delta modes 25
3.7.4 PULL Tokens: Who is the eMSP? 25
3.7.5 PULL Tokens by uid: Retrieve a unique Token 25
3.7.6 PULL Tokens: Retrieve Tokens of a single given eMSP 26
3.7.7 PULL Tokens ToIOP: Timeout Error 26
3.7.8 POST Authorize request: LocationReferences mandatory 26
3.7.9 POST Authorize request: new attribute “authorization_id” 27

3.8 COMMANDS MODULE SPECIFICATIONS 27
3.8.1 StartSession request: new attribute « authorization_id » 27
3.8.2 ReserveNow command 27
3.8.3 UnlockConnector command 27
3.9 SESSIONS MODULE SPECIFICATIONS 27
3.9.1 Session Initialisation 27
3.9.2 Usage of "authorization_id" 27
3.9.3 Minimum Interval Between Session Updates 27
3.9.4 Store and forward – PUT Sessions 28
3.10 CDRS MODULE SPECIFICATIONS 28
3.10.1 CDR sending frequency 28
3.10.2 Usage of "authorization_id" 28
3.10.3 CDR content 28
3.10.4 Store and forward – POST CDRs 29
3.10.5 Send the signed data (Calibration Law / Eichrecht) 29
3.11 TARIFFS MODULE SPECIFICATION 29
3.11.1 Tariffs flows implemented by GIREVE 29
3.11.2 Specific properties added by GIREVE 30
3.11.3 Store and forward – PUT TARIFFS 30
4 EMSP SPECIFIC IMPLEMENTATION GUIDELINES 31
4.1 USE CASES COVERED BY IOP 31
4.1.1 Technical Use cases 31
4.1.2 EVCI data 31
4.1.3 Roaming 31
4.2 USE CASES REQUIRED BY GIREVE 32
4.2.1 Always required 32
4.2.2 If the eMSP implements the "Locations static data download" feature 33
4.2.3 If the eMSP implements the "Locations dynamic data download" feature 33
4.2.4 If the eMSP implements the "Roaming" feature 33
4.3 CONNECTION & REGISTER SPECIFICATIONS 33
4.3.1 Update Credentials FromIOP 33
4.4 LOCATIONS MODULE SPECIFICATIONS 34
4.4.1 Static and dynamic attributes 34
4.4.2 PULL Locations ToIOP: Who is the CPO? 34
4.4.3 PULL Locations: Retrieve Locations of a single given CPO 34
4.4.4 PULL Locations: OCPI-2.2.1 CPO 34
4.4.5 PULL Locations ToIOP: Plug&Charge (P&C): Connector Object - new attribute "capabilities" 34
4.4.6 PULL Locations ToIOP: Get Object 35
4.4.7 PULL Locations ToIOP: Get List Pagination 35
4.4.8 PULL Locations ToIOP: Get List, Full and Delta modes 36
4.4.9 PULL Locations ToIOP: evse_id 36
4.4.10 PUSH Locations FromIOP 36
4.4.11 PULL Locations FromIOP 37
4.5 TOKENS MODULE SPECIFICATIONS 37
4.5.1 Share Tokens list 37
4.5.2 PULL Tokens FromIOP: Pagination and periodicity 37
4.5.3 POST Authorize request: LocationReferences mandatory 37
4.5.4 POST Authorize request: new attribute « authorization_id » 37
4.6 COMMANDS MODULE SPECIFICATIONS 37
4.6.1 StartSession request: new attribute « authorization_id » 38
4.6.2 ReserveNow command 38
4.6.3 UnlockConnector command 38
4.6.4 "evse_uid" mandatory in StartSession command. 38
4.7 SESSIONS MODULE SPECIFICATIONS 38
4.7.1 Usage of "authorization_id" 38

4.7.2 PULL Sessions ToIOP: Get List Pagination *38*
4.8 CDRs MODULE SPECIFICATIONS 39
4.8.1 Usage of "authorization_id" *39*
4.8.2 CDR content *39*
4.8.3 Add billing information in "Remark" field *39*
4.8.4 Get the signed data (Calibration Law / Eichrecht) *39*
4.8.5 PULL CDRs ToIOP: Get List Pagination *39*
4.8.6 PULL CDRs ToIOP: Timeout Error *40*
4.9 TARIFFS MODULE SPECIFICATIONS 40
4.9.1 Tariffs flows implemented by GIREVE *40*
4.9.2 PULL Tariffs ToIOP: Get List Pagination *40*
4.9.3 PULL Tariffs ToIOP: Get List, Full and Delta modes *41*
4.9.4 Specific properties added by GIREVE *41*

5 EXAMPLES OF OCPI REQUESTS 45

5.1 CREDENTIALS & REGISTER 45
5.1.1 ToIOP_GET_emsp_versions, FromIOP_GET_emsp_versions, ToIOP_GET_cpo_versions, FromIOP_GET_cpo_versions *45*
5.1.2 ToIOP_GET_emsp_version_detail_2.1.1, FromIOP_GET_emsp_version_detail_2.1.1, ToIOP_GET_cpo_version_detail_2.1.1, FromIOP_GET_cpo_version_detail_2.1.1 ..... *45*
5.1.3 ToIOP_POST_emsp_credentials_2.1.1, FromIOP_POST_emsp_credentials_2.1.1, ToIOP_POST_cpo_credentials_2.1.1, FromIOP_POST_cpo_credentials_2.1.1 ..... *46*
5.1.4 ToIOP_DELETE_emsp_credentials_2.1.1, FromIOP_DELETE_emsp_credentials_2.1.1, ToIOP_DELETE_cpo_credentials_2.1.1, FromIOP_DELETE_cpo_credentials_2.1.1 ..... *47*
5.2 LOCATIONS 48
5.2.1 ToIOP_GET_cpo_locations_2.1.1, FromIOP_GET_cpo_locations_2.1.1 ..... *48*
5.2.2 ToIOP_PUT_emsp_locations_2.1.1, FromIOP_ PUT_emsp_locations_2.1.1 (on Locations) *50*
5.2.3 ToIOP_PATCH_emsp_locations_2.1.1, FromIOP_ PATCH_emsp_locations_2.1.1 (on EVSE) *52*
5.3 TOKENS 52
5.3.1 ToIOP_GET_emsp_tokens_2.1.1, FromIOP_GET_emsp_tokens_2.1.1 ..... *52*
5.3.2 ToIOP_PUT_cpo_tokens_2.1.1 ..... *53*
5.3.3 ToIOP_PATCH_cpo_tokens_2.1.1 ..... *54*
5.3.4 ToIOP_POST_emsp_tokens_2.1.1, FromIOP_POST_emsp_tokens_2.1.1 ..... *55*
5.3.5 ToIOP_GET_emsp_tokens-unitary_2.1.1 ..... *56*
5.4 COMMANDS 56
5.4.1 ToIOP_POST_cpo_commands_2.1.1, FromIOP_POST_cpo_commands_2.1.1 ..... *56*
5.4.2 Callback ToIOP_POST_cpo_commands_2.1.1, FromIOP_POST_cpo_commands_2.1.1 ..... *57*
5.5 SESSIONS 58
5.5.1 ToIOP_PUT_emsp_sessions_2.1.1, FromIOP_PUT_emsp_sessions_2.1.1 ..... *58*
5.5.2 ToIOP_PATCH_emsp_sessions_2.1.1, FromIOP_PATCH_emsp_sessions_2.1.1 ..... *60*
5.6 CDRs 60
5.6.1 ToIOP_POST_emsp_cdrs_2.1.1, FromIOP_POST_emsp_cdrs_2.1.1 ..... *60*
5.6.2 ToIOP_GET_cpo_cdrs_2.1.1, FromIOP_GET_cpo_cdrs_2.1.1 ..... *62*
5.7 TARIFFS 66
5.7.1 ToIOP_PUT_emsp_tariffs_2.1.1 ..... *66*
5.7.2 ToIOP_GET_cpo_tariffs_2.1.1, FromIOP_GET_cpo_tariffs_2.1.1 ..... *68*

OCPI Protocol Version Reference

This OCPI Protocol Implementation Guide document is based on OCPI v2.1.1, as defined in document:

- OCPI 2.1.1 Open Charge Point Interface 2.1.1, document version : 2.1.1 https://github.com/ocpi

OCPI_2.1.1.pdf

Related documents

Doc reference	Content	Doc name
OCPI-2.1.1		Open Charge Point Interface 2.1.1, document version : 2.1.1

Document history

Version	Description	Date
1.0.3	• Edit of graphic design	24/09/2018
1.0.4	• Adding use cases and mandatory implementations for CPO and eMSP	05/10/2018
1.1.1	• First publication	15/02/2019
1.1.2	• Removal of http header paragraph • Add examples of all OCPI 2.1.1 flows • Minor revisions	15/01/2020
1.1.3	• Explanation about prefix value in the “Remark” field of CDRs • Explanation about IOP HTTP headers • Requirements for evse_id added • Information about “tariff_id” for CPO • Minor revisions	15/05/2020
1.1.4	• Edit of graphic design • Store and forward mechanism explained • Removal of Token.WhiteListType suggestion	20/05/2020
1.1.5	Improvements: • Update of OCPI Github links • Add information about RFID tag format New features: • Tariffs module for CPOs and eMSPs • Usage of new headers to retrieve Locations of a unique CPO or Tokens of a unique eMSP	26/02/2021
1.1.6	New features: • New OCPI flow for CPOs to get a Token by uid	22/04/2021

	• Add reception of signedData by eMSPs in CDRs (German calibration law “Eichrecht”)	
1.2.0	New features: • CPOs can send signedData in CDRs (German calibration law “Eichrecht”) • Tariffs module improvement for CPOs and eMSPs	04/06/2021
1.2.1	Improvements: • Add information about Store & Forward Process. • Add explanation about management of locations. • Add information about Intended Audience. • Add explanation about Cardinality Expression. • Add information about Management of B2B tariffs. • Add information about Connection and register. • Add information about Location module specifications. • CPO operation definition and naming rules. • CPO operation and roaming offers. • “evse_uid” mandatory in StartSession command.	20/12/2022
1.2.2	New features: • Modification of “evse_uids” property cardinality in “POST Tokens authorize request”.	03/05/2023
1.2.3	Improvements: • Add information about Store & Forward Process. • Add information about Full & Delta modes. • Add information about List Pagination Response. • Add information about Gireve’s Pulling Limit. • Url Update for GET Tokens Unitary Service. • Tariffs module improvement for CPOs and eMSPs • Update examples of all OCPI 2.1.1 flows	23/04/2024
1.2.4	New features: • Attribut “publish” set to “false” for OCPI-2.2.1 CPO’s location. • Add information about PULL Locations ToIOP: Get Object.	03/10/2024
1.2.5	Improvements: • Add information about Security.	29/11/2024
1.2.6	New features: • EVSE PnC compatibility.	19/12/2024
1.2.7	New features: • Add new attribute “capabilities” to the Connector object to ensure PnC compatibility.	26/09/2025
1.2.8	Improvements: • Add information about “evse_uids” property cardinality in “POST Tokens authorize request”.	17/11/2025
1.2.9	Improvements: • Remove the “Store & Forward” process from the Locations module and provide additional details regarding this process for the Sessions, CDRs, and Tariffs modules. • Add information regarding the retrieval (pulling) of Tokens and CDRs following a timeout error. • Add information on the required time interval to be respected between session push requests.	20/04/2026
1.2.10	Improvements:	21/09/2026

- Add information regarding Anti-spam protection

1 Introduction

1.1 Aims

This document describes guidelines to perform a proper connection with the GIREVE's platform using the OCPI protocol.

1.2 Intended Audience

This document is dedicated to technical teams (system administrators, developers, etc.) of systems connected or to be connected to the GIREVE's platform through OCPI.

This document also covers some operational requirements that must be considered when implementing the OCPI 2.1.1 protocol.

1.3 Definitions and Abbreviations

Word	Meaning
IOP	Inter-operation Platform. IOP is the acronym of the GIREVE's eMobility Services Platform.
RPC	Référentiel des Points de Charge *(in French)* = Charge Points Repository *(in English)* The RPC is a system, built around a database that contains Electric Vehicles Charge Infrastructure (EVCI) description. It is connected to IOP. IOP's interfaces (eMIP, OCPI …) are the only way to access RPC, for partners systems.
ToIOP	Referring to flows for which an operator requests GIREVE platform IOP. Partner system is client. IOP is server
FromIOP	Referring to flows for which GIREVE platform IOP requests an operator. IOP is client. Partner system is server

1.4 Cardinality expression

To designate the cardinality of fields on data structures the following symbols are used:

Symbol	Description
?	An optional object. If not set, it might be `null`, or the field might be omitted. When the field is omitted and it has a default value, the value is the default value.
1	Required object.
*	A list of zero or more objects. If empty, it might be `null`, `[]` or the field might be omitted.
+	A list of at least one object.

Note that some attributes may not be mandatory by the OCPI standard but are required by Gireve for quality reasons.

Gireve's data integration process includes quality controls that ensure the consistency of its data referential. The standard data quality level of Gireve is sustained by numerous mandatory attributes that are not all mandatory by the OCPI standard.

For further information about the cardinality of attributes by Gireve's quality standard, please refer to the document "Starting roaming operations as a CPO".

2 Integration Guidelines

2.1 Technical

2.1.1 Supported OCPI versions

IOP OCPI implementation has begun with version 2.1.1. IOP does not support previous versions.

2.1.2 Security

Exchange Authorization Tokens & HTTPS

IOP follows the security standard mechanisms of OCPI. At the connection, GIREVE provides new operator connecting with temporary Token(s) to register as CPO and/or eMSP. The Token should be used when the connection is initiated by the operator.

In the meantime, the operator can provide GIREVE with temporary Token(s) for IOP to initiate the connection. It is not mandatory as the operator is able to launch the Connection & Register process.

During the Connection & Register process, IOP and the operator exchange final Tokens to request each other, and endpoints of their respective OCPI modules. **GIREVE requires all endpoints to be in HTTPS in order to secure communication between the two backends.**

IP Filtering

IOP also uses customer IP filtering as an additional security measure. This ensures that only trusted sources can access the API, reinforcing security and protecting against unauthorized access attempts.

Calling partners must provide a list of authorised IP addresses before connecting to the platform.

SSL client certificate security

To enhance security, Gireve configures authentication via SSL client certificates in both directions.

However, **only the client certificate used by the partner to send requests to IOP, which is generated by Gireve, is mandatory**. The client certificate used by IOP to send requests to the partner, provided by the partner, is optional.

2.1.3 Client owned object push

OCPI introduces a specific use of resource identification mechanism, to manage situation where resource belongs to servers and situation where resource belongs to client. See OCPI client owned object.

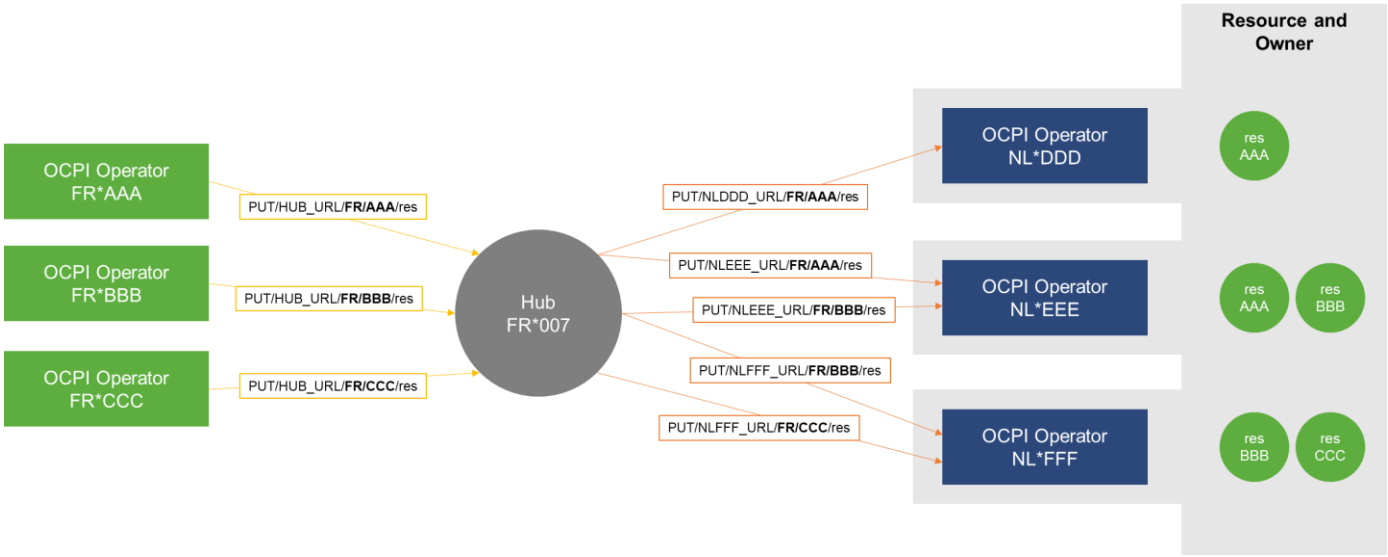
In OCPI, Objects managed through Rest protocol are owned by the CPO or the eMSP:

- Tokens are owned by the eMSP
- Locations are owned by the CPO
- Sessions are owned by the CPO
- CDRs are owned by the CPO
- Tariffs are owned by the CPO

IOP acts as a hub which routes the messages between CPOs and eMSPs, so IOP does not own objects exchanged by these two actors during their communication.

IOP is not the owner of the exchanged resources. Following this principle, IOP uses "country-code" and "partner-id" of the object owner during communication with other operators.

For example, IOP pushes Locations of a CPO to an eMSP using "country-code/party-id" of the CPO in the URL.



2.1.4 Pagination

IOP implements pagination mechanisms described by OCPI. *See OCPI pagination mechanism.*

IOP requires operators to use pagination when they pull resources requesting IOP. If the operator does not use pagination arguments in its request, IOP will force it answering with the first **X** items and a Link to the second page if any.

Finally, IOP for each module has its own max size limit per page (20 Locations, 1000 Tokens, …). These limits can change with IOP evolutions, the operator implementation might be flexible regarding these limits.

2.1.5 IOP HTTP headers

The next version of OCPI, OCPI 2.2, integrates new extra headers enabling the sharing of a single OCPI connection to multiple operators.

GIREVE has begun to deploy these extra headers in its OCPI version 2.1.1 but it is an ongoing action.

These extra headers should not be considered yet in OCPI 2.1.1, except for "*PULL Tokens: Retrieve Tokens of a single given eMSP*" (see chapter 3.7.6 page 24) and "*PULL Locations: Retrieve Locations of a single given CPO*" (see chapter 4.4.3 page 31):

- ocpi-to-country-code
- ocpi-to-party-id
- ocpi-from-country-code
- ocpi-from-party-id

2.1.6 Anti-spam protection

To protect the platform and other partners from spam, restrictions have been put in place on the use of the following web services:

2.1.6.1 Location OCPI 2.1.1

Webservice	Path	Constraint rule	Rate limit
ocpi_toiop_put_emsp_locations_2.1.1	*/ocpi/emsp/2.1.1/locations/<countrycode>/<partyid>/<locationid>*	Same country code, party id and location id	1 request / minute
ocpi_toiop_put_emsp_locations_evse_2.1.1	*/ocpi/emsp/2.1.1/locations/<countrycode>/<partyid>/<locationid>/<evseid>*	Same country code, party id, location id and evse id	2 requests / minute
ocpi_toiop_patch_emsp_locations_2.1.1	*/ocpi/emsp/2.1.1/locations/<countrycode>/<partyid>/<locationid>*	Same country code, party id and location id	1 request / minute
ocpi_toiop_patch_emsp_locations_evse_2.1.1	*/ocpi/emsp/2.1.1/locations/<countrycode>/<partyid>/<locationid>*	Same country code, party id, location id and evse id	2 requests / minute

2.1.6.2 Token OCPI 2.1.1

Webservice	Path	Constraint rule	Rate limit
ocpi_toiop_put_cpo_tokens_2.1.1	*/ocpi/cpo/2.1.1/tokens/<countrycode>/<partyid>/<tokenid>*	Same country code, party id and token id	1 request / minute
ocpi_toiop_patch_cpo_tokens_2.1.1	*/ocpi/cpo/2.1.1/tokens/<countrycode>/<partyid>/<tokenid>*	Same country code, party id and token id	1 request / minute

2.1.6.3 Session OCPI 2.1.1

Webservice	Path	Constraint rule	Rate limit
ocpi_toiop_put_emsp_sessions_2.1.1	*/ocpi/emsp/2.1.1/sessions/<countrycode>/<partyid>/<sessionid>*	Same country code, party id and session id	1 request / minute
ocpi_toiop_patch_emsp_sessions_2.1.1	*/ocpi/emsp/2.1.1/sessions/<countrycode>/<partyid>/<sessionid>*	Same country code, party id and session id	1 request / minute

2.1.6.4 Tariff OCPI 2.1.1

Webservice	Path	Constraint rule	Rate limit
ocpi_toiop_put_emsp_tariffs_2.1.1	*/ocpi/emsp/2.1.1/tariffs/<countrycode>/<partyid>/<tariffid>*	Same country code, party id and tariff id	1 request / minute

For these web services, the limit is applied to the path being called. Therefore, two calls with the same path are subject to the rate limit for the current minute. At the start of the next minute, the rate limit counter is reset to zero.

If this threshold is exceeded, any new request following the same path during that minute will be rejected with error code 429. These requests will not be processed by our system.

You will receive the following error message:
HTTP 429


```
{
  "data": {},
  "status_code": 2914,
  "status_message": "TOO_MANY_REQUESTS : Rate limit of 1pm exceeded. Please reduce your request rate."
}
```


2.2 OCPI modules implemented by IOP

All OCPI modules have been implemented in IOP.

List of modules implemented by IOP:

Module	Usage
Credentials	Initialise and update IT connection information (endpoints, OCPI versions, …)
Locations	Exchange charge infrastructure information
Tokens	Manage Tokens and local authorizations
Commands	Start a charge remotely, Send commands during charge
Sessions	Exchange information during charge
CDRs	Send the final charge report
Tariffs	Exchange Tariffs information

2.3 GIREVE management of Locations data

GIREVE and its systems distinguish two natures of Location properties:

- Static data: Locations properties which never or almost never change
- Dynamic data: Locations properties which can change frequently (e.g., EVSE status, Connector tariff_id)

All Location properties are considered as static data except for the status of the EVSE and the tariff_id attached to a connector.

GIREVE performs a specific process to first integrate static data of CPO Locations in its charge point repository then to integrate static data change like change on a Locations of a CPO or new Locations or EVSEs. This process implies data quality tests and data completion of the CPO Locations.

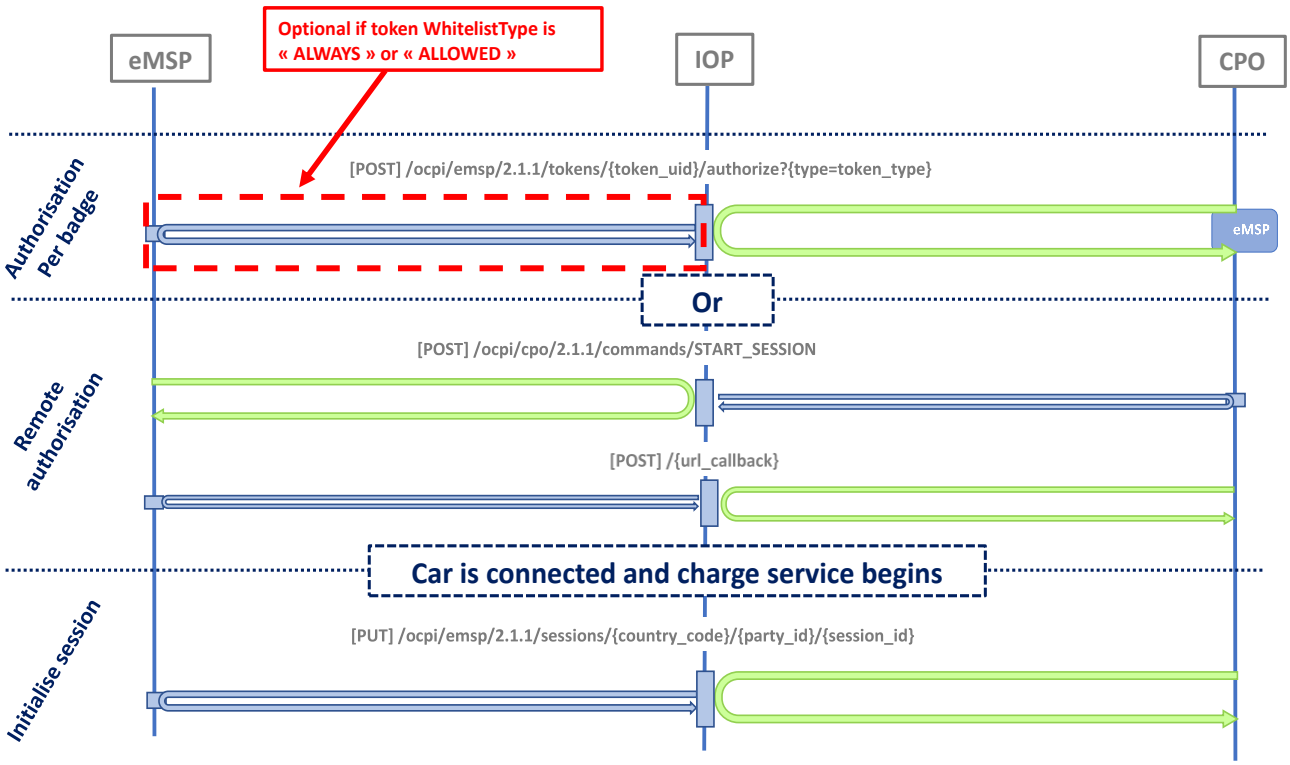
This process is asynchronous from the standard connection of the CPO with the GIREVE IOP platform, meaning that new Locations of the CPO or updates on them can be seen in the GIREVE charge point repository several days after the first PUSH from the CPO to the GIREVE IOP platform.

2.4 Roaming

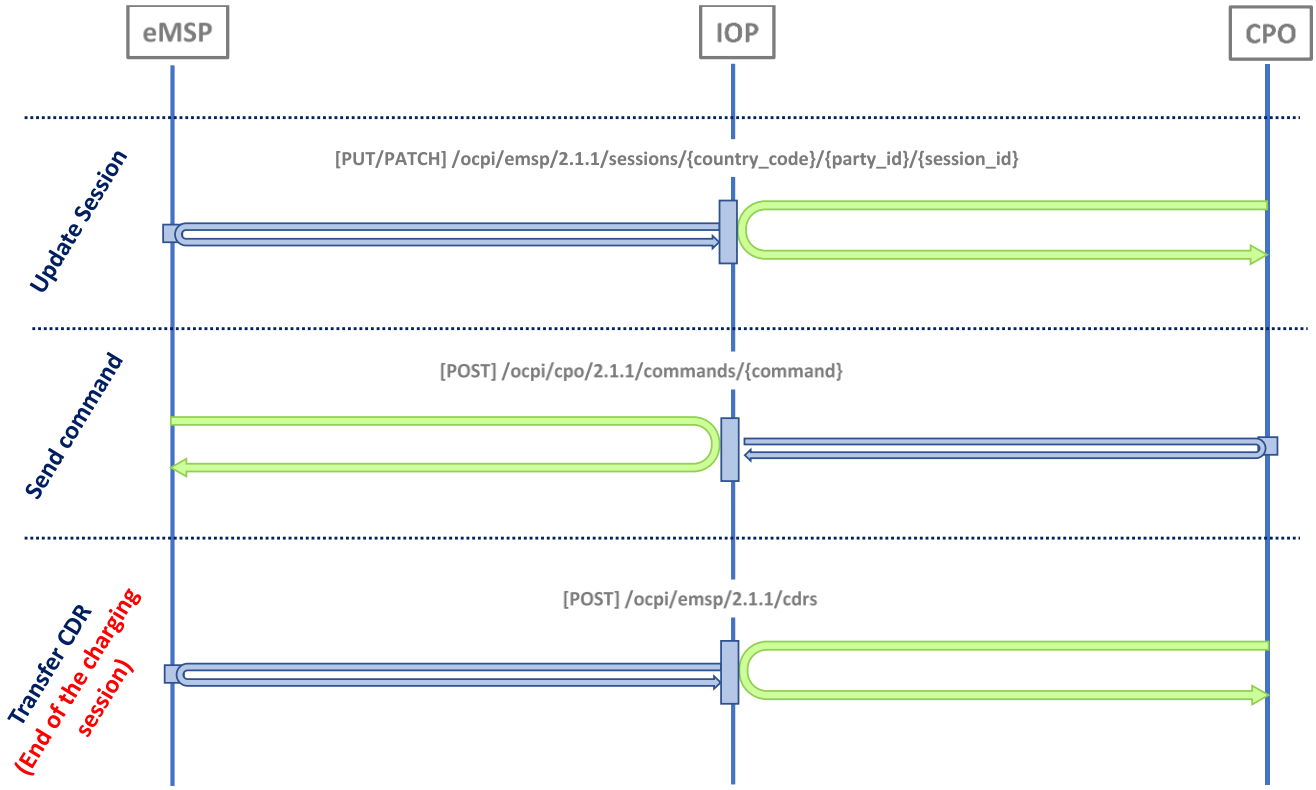
2.4.1 General workflow

An OCPI Roaming session through IOP should run the following workflow

2.4.1.1 Authorisation then initialise Session object



2.4.1.2 Session status information exchange, stop then Charge Detail Record



2.4.2 New attribute « authorization_id »

IOP uses a new attribute « authorization_id » for Session, CDR, AuthorizationInfo and StartSession objects.

This information is used to associate the « authorization » given by the eMSP at the beginning of a charging session with Sessions and CDR related to. It answers to real business cases, for example an eMSP giving 2 authorisations and receiving 3 CDRs. Which ones should it pay?

This new property must be provided by the eMSP during the authorisation process, then used by the CPO when it sends Sessions and CDRs related to the authorisation.

This property is implemented in OCPI 2.2 under the name “authorization_reference”.

a. *"authorization_id" in local authorization*

***AuthorizationInfo* Object**

Property	Type	Card.	Description
allowed	Allowed	1	Status of the Token, and if it is allowed to charge at the optionally given location.
authorization_id	string(36)	?	**IOP requirements** Unique identifier of the authorisation within the eMSPs platform
location	LocationReferences	?	Optional reference to the location if it was request in the request, and if the EV driver is allowed to charge at that location. Only the EVSEs/Connectors the EV driver is allowed to charge at are returned.
info	DisplayText	?	Optional display text, additional information to the EV driver.

2.4.2.1 *"authorization_id" in remote authorization*

***StartSession* Object**

Property	Type	Card.	Description
response_url	URL	1	URL that the CommandResponse POST should be sent to. This URL might contain an unique ID to be able to distinguish between StartSession requests.
token	Token	1	Token object the Charge Point has to use to start a new session.
authorization_id	string(36)	?	**IOP requirements** Unique identifier of the authorisation within the eMSPs platform
location_id	string(39)	1	Location.id of the Location (belonging to the CPO this request is send to) on which a session is to be started.
evse_uid	string(39)	?	Optional EVSE.uid of the EVSE of this Location on which a session is to be started.

2.4.2.2 "authorization_id" in Sessions

Session Object

Property	Type	Card.	Description
id	string(36)	1	The unique id that identifies the session in the CPO platform.
authorization_id	string(36)	?	**IOP requirements** Unique identifier of the authorisation gave by the eMSP at the authorisation step
start_datetime	DateTime	1	The time when the session became active.
end_datetime	DateTime	?	The time when the session is completed.
kwh	number	1	How many kWh are charged.
auth_id	string(36)	1	Reference to a token, identified by the auth_id field of the Token.
auth_method	AuthMethod	1	Method used for authentication.
location	Location	1	The location where this session took place, including only the relevant EVSE and connector
meter_id	string(255)	?	Optional identification of the kWh meter.
currency	string(3)	1	ISO 4217 code of the currency used for this session.
charging_periods	ChargingPeriod	*	An optional list of charging periods that can be used to calculate and verify the total cost.
total_cost	number	?	The total cost (excluding VAT) of the session in the specified currency. This is the price that the eMSP will have to pay to the CPO. A total_cost of 0.00 means free of charge. When omitted, no price information is given in the Session object, this does not have to mean it is free of charge.
status	SessionStatus	1	The status of the session.
last_updated	DateTime	1	Timestamp when this Session was last updated (or created).

2.4.2.3 "authorization_id" in CDRs

CDR Object

The *CDR* object describes the Charging Session and its costs. How these costs are build up etc.

Property	Type	Card.	Description
id	CiString(36)	1	Uniquely identifies the CDR within the CPOs platform (and suboperator platforms). **IOP requirements** Session and CDR objects are the same. So CDR.id MUST be equal to Session.id.
authorization_id	string(36)	?	**IOP requirements** Unique identifier of the authorisation gave by the eMSP at the authorisation step
start_date_time	DateTime	1	Start timestamp of the charging session.
stop_date_time	DateTime	1	Stop timestamp of the charging session.
auth_id	string(36)	1	Reference to a token, identified by the auth_id field of the Token.
auth_method	AuthMethod	1	Method used for authentication.
location	Location	1	Location where the charging session took place, including only the relevant EVSE and Connector.
meter_id	string(255)	?	Identification of the Meter inside the Charge Point.
currency	string(3)	1	Currency of the CDR in ISO 4217 Code.
tariffs	Tariff	*	List of relevant tariff elements, see: Tariffs. When relevant, a "Free of Charge" tariff should also be in this list, and point to a defined "Free of Charge" tariff.
charging_periods	ChargingPeriod	+	List of charging periods that make up this charging session. A session consists of 1 or more periods, where each period has a different relevant Tariff.
total_cost	number	1	Total cost (excluding VAT) of this transaction.
total_energy	number	1	Total energy charged, in kWh.
total_time	number	1	total duration of this session (including the duration of charging and not charging), in hours.
total_parking_time	number	?	Total duration during this session that the EV is not being charged (no energy being transfered between EVSE and EV), in hours.
remark	string(255)	?	Optional remark, can be used to provide addition human readable information to the CDR, for example: reason why a transaction was stopped.
last_updated	DateTime	1	Timestamp when this CDR was last updated (or created).

2.4.3 Management of B2B tariffs

The CPO connected to GIREVE through OCPI have two options to manage and "publish" B2B tariffs:

B2B tariffs are described in roaming agreements: The description and the commitment related to tariffs are contained in roaming agreements signed on the GIREVE connect place:

- Tariffs applied for a contract between an eMSP and a CPO are defined and negotiated, using the GIREVE connect place, before signature of the roaming agreement by both parts.
- In case of tariff updates, the CPO and the eMSP must sign an amendment whose management is fully automated.

In this case (tariffs defined in roaming agreements), the CPO should not use the OCPI Tariffs module.

B2B tariffs are not described in roaming agreements: Tariffs are not described in roaming agreements and the CPO transfers them through the OCPI Tariffs module.

- In this case, the CPO manages its tariffs on its own and eMSPs cannot give their insights or validation about applied tariffs.
- Moreover, the CPO is not able to differentiate its tariffs according to eMSPs. In other words, the tariff is unique whatever the eMSP.
- In case of tariff updates, the CPO and the eMSP do not need to sign an amendment.

2.4.4 RFID Tokens

The current typical situation for identification is swiping a MIFARE badge. In this case, the relevant RFID tag in such a situation is a character string that shall contain the hexadecimal representation of the 4- or 7-bytes RFID UID (sector 0). Please note that the 7 bytes UID is preferred for interoperability reason.

As an example: "1A2B3C4D5E6F70" shall be interpreted as

- Lowest address byte contains "1A" and "1" is the most significant nibble (half byte) and "A" is the least significant nibble.
- Highest address byte contains "70"
- Equivalent decimal value is "7365887390543728"

The RFID is not case sensitive. We recommend using uppercase characters. Leading zeros must be provided to reach 8 characters in case of 4 bytes UID or 14 characters in case of 7 bytes UID.

2.4.5 Custom OCPI flow to prevent eMSP Tokens download by CPOs

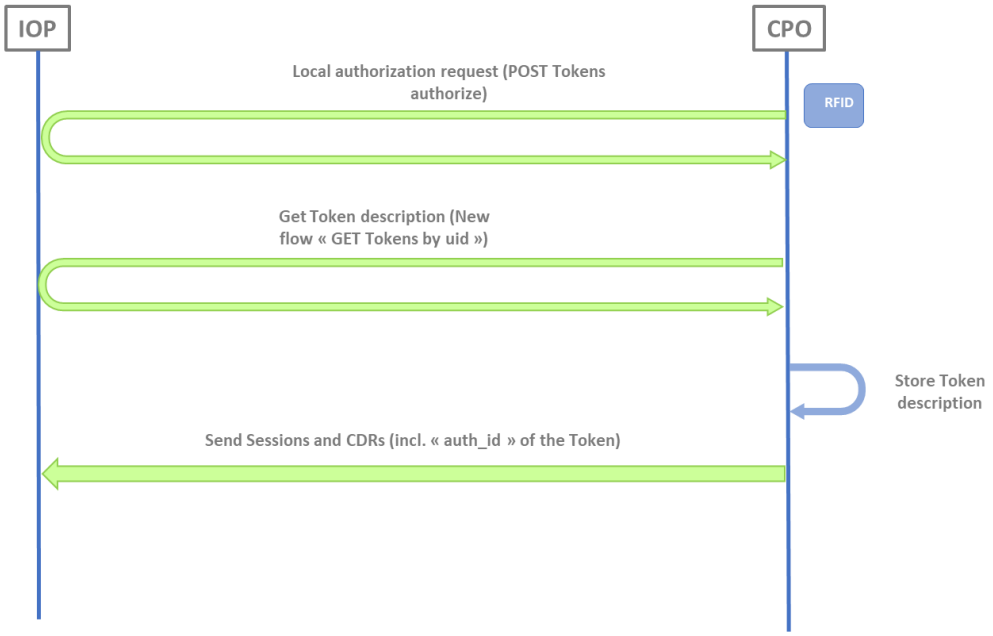
In OCPI 2.1.1, CPOs must send the "auth_id" of a Token in Sessions and CDRs flows.

The only way for CPOs to get this information is to download all Tokens of all eMSPs it is in contract with. CPOs must therefore manage millions of Tokens in their backends, sometime duplicated in case of a backend managing multiple CPOs (a single Token can be downloaded through each CPO connection).

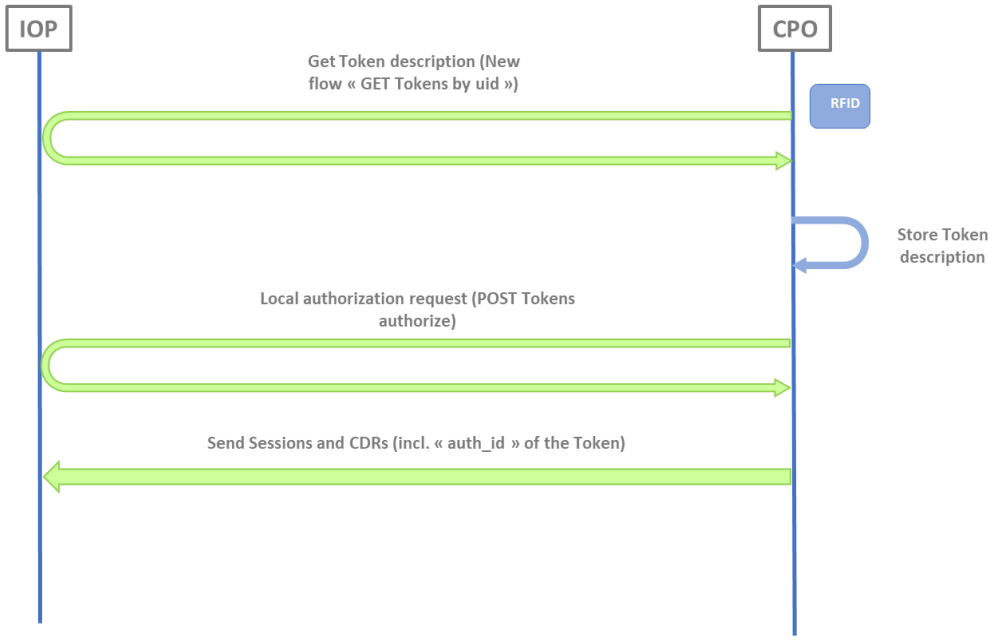
To prevent this complexity and make life easier for CPOs, IOP provides a new OCPI flow enabling a CPO to get the full description of a Token, including the “auth_id” information, by requesting IOP with the “Token’s uid”.

Using this new flow, the CPO can decide to:

- Send an authorization request to IOP without knowledge of the Token. If the authorization is accepted the CPO gets the Token, including the “auth_id”, with the new flow then send Sessions and CDRs.



- Or request this new flow to get the Token description and the confirmation that this Token is owned by an eMSP behind GIREVE, then send the authorization request to GIREVE.



This new flow can also be used as a fallback for CPOs which download all Tokens and have a local authorization for an unknown Token.

Description of this new flow in chapter 3.7.5 PULL Tokens by uid: Retrieve a unique Token.

3 CPO Specific Implementation Guidelines

3.1 CPO operation definition and naming rules

A CPO can manage one or multiple CPO operations. A CPO operation is a homogeneous group of charging points. A CPO operation is defined by its unique eMI3 operation code also called “Operator Id” which is composed by a 2 ALPHA country code plus a 3 ALPHANUMERIC Spot Operator ID.

Example: FR*AB1

eMI3 standard official documentation can be downloaded here: https://emi3group.com/documents-links/

All EVSE included in a given CPO operation must have an evse_id following the eMI3 standard naming rules defined in this specific document, page 27 :

https://emi3group.com/wp-content/uploads/sites/5/2018/12/eMI3-standard-v1.0-Part-2.pdf

Specifically, all evse_id included into a given operation must always start with the Operator Id.

Example: FR*AB1*EABCDEFG*1

It is not allowed to include in a given CPO operation evse_ids that start with a different Operator Id.

3.2 CPO operation and roaming offers

Reminder : If a CPO manages multiple operations, each operation has to go through the OCPI connection establishment process (handshake).

A CPO Operation is the smallest entity that can constitute a roaming offer.

Although Gireve’s technical platform (IOP) is built to manage bilateral communications between 1 operation (i.e. CPO) and another (i.e. eMSP), it is possible to aggregate several operations into a so-called network group on Gireve’s Market Place (Connect-Place). It allows to publish a single roaming offer that includes several operations. The main advantages of such an offer structure are:

- Reduces the number of roaming agreements (1 agreement instead of several bilateral contracts – one per operation).
- Allows to add or remove operations from the network without terminating the current agreement or issuing amendments, enabling new tenants to benefit from already active roaming agreements.

3.3 Use cases covered by IOP

OCPI features are composed by several use cases that a CPO can choose to implement or not when connecting to an operator. In case of connection to GIREVE, here is the list of use cases that a CPO can implement :

3.3.1 Technical Use cases

a. *Connection and register*

Use case	ToIOP/FromIOP	Usage
Register	**FromIOP**	IOP initialises the connection to a CPO beginning by a GET_Versions, then exchanging endpoints and Authorization-token.
Register	**ToIOP**	A CPO initialises the connection to IOP beginning by a GET_Versions, then exchanging endpoints and Authorization-token.
Update CPO credentials	**ToIOP**	A CPO already registered requests IOP to update its Credentials (endPoints, versions, authorization-token).
Unregister	**FromIOP**	IOP unregisters a CPO by requesting a DELETE Credentials
Unregister	**ToIOP**	A CPO unregisters IOP by requesting a DELETE Credentials

If a CPO manages several operations, each operation must go through the OCPI connection process (handshake).

3.3.2 EVCI data

a. *Locations upload*

Use case	ToIOP/FromIOP	Usage
Push Locations Static data	**ToIOP**	A CPO updates its Locations stored in RPC. *Only the property "connector.tariff_Id" can be updated through this flow.*
Push Locations Dynamic data (Only EVSE status)	**ToIOP**	A CPO updates status of its EVSEs stored in RPC.
Pull Locations (Static data)	**FromIOP**	IOP requests the CPO backend to retrieve Locations data.
Check Locations	**ToIOP**	A CPO requests IOP to get its Locations stored in RPC.

3.3.3 Roaming

a. *Tokens*

Use case	ToIOP/FromIOP	Usage
Pull Tokens	**ToIOP**	A CPO requests IOP to get Tokens of eMSPs, it is allowed to get (ie, depending on roaming agreements).
Pull Tokens (of a single eMSP)	**ToIOP**	A CPO requests IOP to get Tokens of a single given eMSP which is identified by "OCPI-TO" headers.
		3.5.3 PULL Tokens by uid:
Pull Token by uid	**ToIOP**	A CPO requests IOP to get single Token which is identified by the user UID.

3.3.3.1 Local authorisation

Use case	ToIOP/FromIOP	Usage
Realtime Authorisation	**ToIOP**	A CPO requests IOP to get Authorisation triggered by a local authentication (swiping of an RFID badge by an EV driver on one of its EVSE, for example).

3.3.3.2 Remote Authorisation

Use case	ToIOP/FromIOP	Usage
Start Session	**FromIOP**	IOP, triggered by a remote start received from an eMSP, requests a CPO to start remotely the charging-session.
Stop Session	**FromIOP**	IOP, triggered by a remote stop received from an eMSP, requests a CPO to stop remotely the charging-session. This use case can happen even for charging-session beginning by an authorisation per RFID badge.

3.3.3.3 Session

Use case	ToIOP/FromIOP	Usage
Push Session	**ToIOP**	A CPO requests IOP to initialise or update an OCPI Session object following an positive Authorisation.
Check Session	**ToIOP**	A CPO requests IOP to get Sessions it has previously initialised.

3.3.3.4 CDR

Use case	ToIOP/FromIOP	Usage
Push CDR	**ToIOP**	A CPO sends to IOP, a CDR related to a terminated charging-session.
Pull CDR	**FromIOP**	IOP requests a CPO to retrieve CDRs that belongs to it.
Check CDR	**ToIOP**	A CPO requests IOP to get CDRs it has previously sent.

3.3.3.5 Tariffs

Use case	ToIOP/FromIOP	Usage
Push Tariffs	**ToIOP**	A CPO sends to IOP, a Tariffs object
Pull Tariffs	**FromIOP**	IOP requests the CPO backend to retrieve Tariffs data

3.4 Use cases required by GIREVE

Some of these use cases are required when connecting to GIREVE

3.4.1 Always required

Use case		Why?
Register/FromIOP **Register/ToIOP**	**OR**	These use cases are needed to initialise connection between an operator and GIREVE

Push EVSE status/ToIOP	A CPO connected to GIREVE must transfer “in realtime” EVSE status change of its EVSEs
Pull Locations/FromIOP	GIREVE wants to be able to refresh EVCI data when needed.

3.4.2 If the CPO implements the “Roaming” feature

Use case	Why?
Realtime Authorisation/ToIOP	A CPO should be able to request eMSP through IOP when a driver uses his RFID badge to charge.
Start Session/FromIOP	Remote authorisation and start features on CPO infrastructure are required by GIREVE
Stop Session/FromIOP	Remote stop features on CPO infrastructure are required by GIREVE
Push Session/ToIOP	A CPO must be able to send information about charging-sessions through Session objects (charge started, …)
Push CDR/ToIOP	The CPO must send the CDR directly after the end of the charging-session.
Pull CDR/FromIOP	GIREVE must be able to retrieve CDRs from CPO platform when needed.
Pull Tokens/ToIOP	Pull Tokens is the only way open for a CPO to download Tokens of eMSP it is in contract with. This feature is “global” (CPO downloads all the tokens of all eMSP it is in contract with), or “unitary” (CPO downloads all the tokens of a single given eMSP)

3.4.3 If the CPO doesn’t commit and describe its tariffs in a roaming agreement

Use case	Why?
Push Tariffs/ToIOP	CPOs must inform in realtime, through IOP, eMSPs about tariff changes.
Pull Tariffs/FromIOP	GIREVE must be able to retrieve all CPO Tariffs. Useful in case of connection initialisation or after a disconnection period.

3.5 Connection & Register specifications

IOP follows the OCPI standard for Connection & Register process. *See OCPI specifications.*

3.5.1 Update Credentials FromIOP

IOP is not able to send a PUT Credentials to update its Credentials on operator backend. In order to update them, IOP sends a “DELETE Credentials” to delete the OCPI connection then starts a new Connection & Register process beginning by a POST Credentials sent by IOP or by the operator.

3.6 Locations module specifications

IOP follows the OCPI standard for Locations upload by a CPO. *See OCPI specifications*.

Nonetheless, the following attributes are mandatory for Gireve although they are optional in the OCPI standard :

Owner.name	Mandatory: this information MUST be registered in the ocpi flow under the object « owner » with the attribute « name ».
operator.name	Mandatory for Gireve - we use it to give information about the brand name
evses.connectors.tariff_id	Mandatory for Gireve
evses.capabilities	Mandatory for Gireve

3.6.1 Static and dynamic attributes

The attributes of the Location object (with EVSE and Connector) are of 2 types:

- Static attributes are data attributes that do not change frequently (address, localisation …)
- Dynamic attributes are data attributes that may change frequently (availability, occupied/free …). Only "EVSE.status" and "Connector.tariff_id" are considered as dynamic.
  These data are integrated in real-time by Gireve when CPOs send updates to IOP.
-

3.6.2 EVSE object

OCPI defines an attribute called « evse_id » which contains the eMI3 id of the EVSE.

The "evse_id" is optional for OCPI but required by GIREVE to ensure seamless compatibility with operators connected via other protocols (eMIP, …).

In addition, the eMI3 standard requires that the eMI3 Id of an EVSE begins by the eMI3 Id of the CPO. See "Part 2 v1.0" of eMI3 standard here: http://emi3group.com/documents-links/)

Ie. All "evse_id" of the CPO "FR*CPO" should start with "FR*CPO*E".

3.6.3 Connector object

IOP follows the OCPI 2.2.1 standard for Locations upload by a CPO. *See OCPI specifications*.

"tariff_id" property on connectors is required for Gireve.

3.6.4 Tariff_id value

GIREVE uses the "tariff_id" information provided by CPOs in Locations to dispatch CPO's EVSEs into separated EVSE tariff groups. Also, CPOs can refer to these tariff groups when they describe their roaming offer including tariffs via the GIREVE connect place (https://connect-place.gireve.com).

If CPOs use the OCPI Tariffs module to send their tariffs, the management of tariffs and relations to the charging infrastructure follows the OCPI standard except that in GIREVE systems, tariffs are linked to EVSEs and not to connectors (ie All connectors of a single EVSE have the same "tariff_id").

3.6.5 Plug&Charge (P&C): Connector Object - new attribute "capabilities"

The ability to enable Plug & Charge (P&C) functionality, eliminating the need for a physical badge, offers a significant improvement in the user experience for EV drivers.

In alignment with the OCPI 2.3 white paper, which recommends adding Plug & Charge capabilities at the Connector level rather than the EVSE level (as currently defined in OCPI 2.1.1 and 2.2.1), Gireve has implemented a mechanism enabling CPOs to inform eMSPs about EVSE compatibility with Plug & Charge.

To achieve this, Gireve has introduced a new attribute "capabilities", at the Connector object level. This attribute accepts the following list of values:

OCPI Field	Enum	Description
Location.evse.connector.capabilities	**ISO_15118_2_PLUG_AND_CHARGE**	Compatibility of an EVSE with PnC using ISO15118-2
Location.evse.connector.capabilities	**ISO_15118_20_PLUG_AND_CHARGE**	Compatibility of an EVSE with PnC using ISO15118-20

3.6.6 PUSH Locations ToIOP

The eMSP Interface is not fully implemented by IOP for the PUSH of static attributes. The web service is present and responds but information is not stored in RPC (Charge Point Repository).

The only data directly updated in RPC after a PUT/PATCH Locations by a CPO are:

- The dynamic attribute EVSE.status.
- The "Connector.tariff_id" information.

3.6.7 PULL Locations ToIOP

If the CPO wants to check the status of a Location, EVSE or Connector object in the IOP system, it can call these URLs:

- /ocpi/emsp/2.2.1/locations/{country_code}/{party_id}/{location_id}
- /ocpi/emsp/2.2.1/locations/{country_code}/{party_id}/{location_id}/{evse_uid}
- /ocpi/emsp/2.2.1/locations/{country_code}/{party_id}/{location_id}/{evse_uid}/{connector_id}

with:

- location_id: unique id of the Location object provided by the CPO.
- evse_uid: unique id of the EVSE object provided by the CPO.
- connector_id: unique id of the Connector object provided by the CPO.

3.6.8 PULL Locations FromIOP

IOP is able to PULL Location requesting CPO backend. In this case, IOP uses the pagination and CPO must respond with a paginated response. If the response is not paginated, it will be ignored by IOP.

The default periodicity is every day.

3.7 Tokens module specifications

IOP follows the OCPI standard for Tokens module. *See OCPI specifications*.

3.7.1 GET Tokens To/FromIOP

IOP is not able to PUSH Tokens to CPO backend. CPO should PULL IOP to get WL.

3.7.2 PULL Tokens ToIOP : Get List Pagination

If the CPO wants to retrieve a list of Tokens, it can call the URL: /ocpi/emsp/2.1.1/tokens using the paginated properties date_from, date_to, offset and limit.

Parameters « offset » and « limit » are optional but IOP always returns a paginated response (subset of objects list and link, X-Total-Count and X-limit headers).
The CPO must call **the link returned in the headers** to get the next pages.

IOP has its own max limit (**1000 Tokens**) and answers with its if the client limit is upper than IOP one or the client doesn't set its limit.

3.7.3 PULL Tokens ToIOP: Get List, Full and Delta modes

If the CPO wants to retrieve all the Tokens, it should not include « date_from » parameter in its request. But if it wants to get only changes (optimised), it should use it.

3.7.4 PULL Tokens: Who is the eMSP?

The eMSP of a Token is not a direct property of the Token object. So, a CPO pulling IOP to get Tokens does not know to which eMSP the Token refers to. That is why IOP replaces the property "Token.issuer" by the eMI3 Id of the eMSP when a CPO get a Token. Using this property, the CPO is able to know who is the eMSP of the Token.

3.7.5 PULL Tokens by uid: Retrieve a unique Token

IOP adds a new OCPI feature enabling a CPO to retrieve the full description of a Token through the Tokens.uid:

WS Name	ToIOP_GET_emsp_Tokens-unitary
HTTP Verb	GET
URL	{GIREVE_URL}/ocpi/emsp/2.1.1/tokens/{uid}?type={tokenType} *uid: Mandatory* *tokenType: Optional (default value: RFID)*
Request's body	empty
Response's body	Standard OCPI response including the Token description in data field. In case of an unknown Token or Token not visible by the CPO, the status_code "2000" is returned

If the CPO is allowed to get Tokens of the eMSP owner, the response includes the full description of the Token.

This new flow prevents CPOs to download all Tokens of all eMSPs. For more description, see *2.4.5 Custom OCPI flow to prevent eMSP Tokens download by CPOs.*

3.7.6 PULL Tokens: Retrieve Tokens of a single given eMSP

The standard OCPI 2.1.1 Tokens pulling allows CPOs to get Tokens of all eMSPs in contract with them.
In some cases, CPOs need only Tokens of a specific given eMSP. For example, when the CPO initializes data of an eMSP after signature of a new roaming agreement.
GIREVE provides a new OCPI 2.1.1 feature by allowing the CPO to get Tokens of a unique eMSP by filling two dedicated OCPI headers in their “GET Tokens” request to GIREVE :

- ocpi-to-country-code: The country code of the targeted eMSP.
- ocpi-to-party-id: The party id of the targeted eMSP.

Therefore, CPOs can request GIREVE without these headers to get Tokens of all eMSPs or including these headers to get Tokens of a unique eMSP.
For information, these headers have been included in the version 2.2 of the OCPI standard.

3.7.7 PULL Tokens ToIOP: Timeout Error

In the event of a timeout during a Get Token request (HTTP 500), the CPO may retry the request for up to one hour, leveraging a caching mechanism.

Gireve recommends waiting at least 10 minutes before retrying. Otherwise, the CPO may encounter the following error response:

```
{
   "status_code": 2914,
   "status_message": "A pulling already ran for a similar request",
   "timestamp": "2026-03-25T14:22:01Z"
}
```

In such cases, the CPO should retry the same request.

Important: For the caching mechanism to apply, all retry requests must use exactly the same parameters as the initial request (*date_from, date_to, offset, and limit*). Any variation will be treated as a new request, and the cache will not be utilized.

3.7.8 POST Authorize request: LocationReferences mandatory

In OCPI, the body of a POST Authorize request can contain a LocationReferences object.
For IOP, the LocationReferences object is mandatory.
IOP previously required that CPOs send only one “evse_uid” in the real-time authorization request. This limit is now removed, CPOs can send 0 to N “evse_uid” in the request.

Please note that a CPO can send a real-time authorization request with **0 or N evse_uid values** in the payload.

In such cases, Gireve will **randomly select an EVSE** belonging to the specified location and forward the request to the eMSP. The EVSE chosen by Gireve may **not necessarily be the one actually used** in the authorization process.

However, the CPO is **required to use the correct EVSE** in both the **Sessions** and **CDRs** objects.

3.7.9 POST Authorize request: new attribute “authorization_id”

IOP answers to a POST Token Authorize request with a new attribute of AuthorizationInfo object, the “authorization_id”. Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

The CPO must store this information to send it in Sessions and CDRs related to this Authorization.

3.8 Commands module specifications

IOP follows the OCPI standard for Commands received by a CPO. *See OCPI specifications*.

3.8.1 StartSession request: new attribute « authorization_id »

IOP sends a StartSession command to a CPO with a new attribute, the “authorization_id”. Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

The CPO must store this information to send it in Sessions and CDRs related to this Authorization.

3.8.2 ReserveNow command

The ReserveNow command is not yet implemented by IOP.

3.8.3 UnlockConnector command

The UnlockConnector is not yet implemented by IOP.

3.9 Sessions module specifications

IOP follows the OCPI standard for Sessions sent by a CPO. *See OCPI specifications*.

3.9.1 Session Initialisation

The CPO must send, after a remote or local authorization validated by the eMSP, a PUT Session with SessionStatus when the EV plugs to the EVSE.

This flow gives information to eMSP that the charge of its customer has really started.

3.9.2 Usage of “authorization_id”

When sending a Session, the CPO defines the Authorization it refers to on providing the “authorization_id” property. Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

3.9.3 Minimum Interval Between Session Updates

Gireve recommends respecting a minimum interval of **two minutes** between Push (PUT / PATCH) Session update requests.

If a CPO sends session updates at intervals shorter than **two minutes**, these requests may be considered as spam and filtered by the Gireve platform. In such cases, the updates will neither be accepted nor forwarded to the eMSP.

This recommendation applies exclusively to meter value updates (i.e., all changes except session status). Session status updates are not subject to this minimum interval constraint and may be processed without this restriction.

3.9.4 Store and forward – PUT Sessions

A Store and Forward mechanism must be implemented to ensure that no session is lost in case of connection issues. Any PUT session request that does not receive a successful response (HTTP 2XX) from the Gireve IOP platform must be stored on the CPO side, with an active retry process.

Once the connection is restored, session messages must be resent in a FIFO order.

Retries should only be performed in two specific cases: the initial session update (status **ACTIVE**) and the final session update (status **COMPLETED**). For all intermediate updates, retries are unnecessary and not recommended, as only these key transitions carry functional and contractual value according to OCPI. Additionally, retries must not be performed immediately when receiving platform error codes such as 425 (Too Early) or 429 (Too Many Requests), as these indicate that requests are being sent too early or too frequently; immediate retries would worsen the situation. In such cases, the client is expected to wait several minutes before retrying, using a progressive backoff strategy (e.g., 5 min → 10 min → 20 min → …).

1.1.1 Retries must never be executed in an uncontrolled loop or in parallel bursts. A strict retry policy should be applied: no more than one retry every defined interval (e.g., 5 min → 10 min → 20 min → …), with ideally one processing queue per flow type (Sessions, CDRs, Tokens, etc.) and sequential handling to ensure stability and compliance.

3.10 CDRs module specifications

IOP follows the OCPI standard for Sessions sent by a CPO. *See OCPI specifications*.

3.10.1 CDR sending frequency

GIREVE requires CDRs to be sent directly at the end of the charging-session.

eMSPs will so be able to display the charging-session price directly to their end customers.

3.10.2 Usage of “authorization_id”

When sending a CDR, the CPO defines the Authorization it refers to on providing the “authorization_id” property. Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

3.10.3 CDR content

The “total_time” value is the total duration of this session (including the duration of charging and not charging).

It doesn’t include the duration during which the EVSE is out of order so cannot supply the service. The out of order duration should be free of charge for eMSPs.

3.10.4 Store and forward – POST CDRs

Similarly to a PUT sessions, Store and Forward mechanism must be implemented to ensure that no CDR may be lost, in case of a connection loss. Any POST Cdrs that didn't get a correct response (HTTP code : 2XX) from the GIREVE platform IOP must be stored on CPO side and a retry process must be active.

Additionally, retries must not be performed immediately when receiving platform error codes such as 425 (Too Early) or 429 (Too Many Requests), as these indicate that requests are being sent too early or too frequently; immediate retries would worsen the situation. In such cases, the client is expected to wait several minutes before retrying, using a progressive backoff strategy (e.g., 5 min → 10 min → 20 min → …).

Retries must never be executed in an uncontrolled loop or in parallel bursts. A strict retry policy should be applied: no more than one retry every defined interval (e.g., 5 min → 10 min → 20 min → …), with ideally one processing queue per flow type (Sessions, CDRs, Tokens, etc.) and sequential handling to ensure stability and compliance.

3.10.5 Send the signed data (Calibration Law / Eichrecht)

GIREVE has extended OCPI 2.1.1 for CPOs to send signed data in CDRs, aiming to be compliant with the German calibration law (Eichrecht).

These signed data are included in CDRs following the OCPI 2.2 specifications.

To activate this feature, the CPO must validate its implementation through the standard GIREVE certification process before the deployment and activation on PRODUCTION. (Please contact connection-roaming@gireve.com).

3.11 Tariffs module specification

IOP follows the OCPI standard for Tariffs upload by a CPO. *See OCPI specifications.*

It is mandatory to patch Locations to Gireve (Locations Module) after changing the Tariff_ID of a charging point, and whenever a new Tariff_ID is added. It is also mandatory to push Tariffs (Tariff Module) after any change in tariffs.

Gireve only manages one Tariff_ID per Charging Point, thus several connectors from the same Charging Point must share the same Tariff ID. If it is not the case, the first Tariff_ID of that Charging Point received by Gireve's system will be saved.

3.11.1 Tariffs flows implemented by GIREVE

IOP only implements:

- PUT Tariffs: Used by a CPO to create and update a tariff.
- GET Tariffs: Used by IOP to pull Tariffs of a CPO.

3.11.2 Specific properties added by GIREVE

a. *TariffDimensionType "SESSION_TIME"*

Not yet available

In standard OCPI V2.1.1, the CPO can define 2 different "TariffDimensionType" related to the duration:

- **TIME**: "time charging: defined in hours, step_size multiplier: 1 second" *(description from the OCPI Github).*
- **PARKING_TIME**: "time not charging : defined in hours, step_size multiplier: 1 second" *(description from the OCPI Github).*

GIREVE adds a third "TariffDimensionType" named "SESSION_TIME" with the following description:

- **SESSION_TIME**: "time charging or not: defined in hours, step_size multiplier: 1 second".

3.11.3 Store and forward – PUT TARIFFS

For Tariffs, retries are optional and non-blocking. A retry mechanism may be implemented, but it is only recommended under controlled conditions: retries should be infrequent (in hours, not minutes) and must not follow any loop logic. This is because tariff data is relatively static and does not carry real-time criticality.

Retries must not be executed immediately when receiving platform error codes such as 425 (Too Early) or 429 (Too Many Requests). These errors indicate that requests are being sent too early or too frequently, and immediate retries would worsen the situation. In such cases, the client is expected to wait at least one hour before retrying, using a progressive backoff strategy.

A strict retry policy must be applied: retries must never be executed in an uncontrolled loop or in parallel bursts, especially when no successful response (HTTP 2XX) has been received. As a general recommendation, no more than one retry per hour should be performed. Ideally, a dedicated processing queue should be implemented per flow type (Sessions, CDRs, Tokens, Tariffs, etc.), with sequential processing to ensure system stability and compliance.

4 eMSP Specific Implementation Guidelines

4.1 Use cases covered by IOP

OCPI features are composed by several use cases that an eMSP can choose to implement or not when connecting to an operator. In case of connection to GIREVE, here is the list of use cases that a CPO can implement :

4.1.1 Technical Use cases

a. *Connection and register*

Use case	ToIOP/FromIOP	Usage
Register	**FromIOP**	IOP initialises the connection to a eMSP beginning by a GET_Versions, then exchanging endpoints and Authorization-token.
Register	**ToIOP**	An eMSP initialises the connection to IOP beginning by a GET_Versions, then exchanging endpoints and Authorization-token.
Update CPO credentials	**ToIOP**	An eMSP already registered requests IOP to update its Credentials (endPoints, versions, authorization-token)
Unregister	**FromIOP**	IOP unregisters an eMSP by requesting a DELETE Credentials
Unregister	**ToIOP**	An eMSP unregisters IOP by requesting a DELETE Credentials

4.1.2 EVCI data

a. *Locations upload*

Use case	ToIOP/FromIOP	Usage
Push Locations Static data	**FromIOP**	IOP sends update of Locations to an eMSP.
Push Locations Dynamic data (Only EVSE.status)	**FromIOP**	IOP sends EVSE status updates to an eMSP.
Pull Locations (Static data)	**ToIOP**	An eMSP requests IOP to get Locations of CPOs, it is allowed to get (ie depending on its roaming agreements).
Pull Locations **(of a single CPO)**	**ToIOP**	An eMSP requests IOP to get Locations of a single given CPO which is identified by “OCPI-TO” headers.

4.1.3 Roaming

a. *Tokens*

Use case	ToIOP/FromIOP	Usage
Push Tokens	**ToIOP**	An eMSP updates its Tokens stored in IOP.
Pull Tokens	**FromIOP**	IOP requests an eMSP to retrieve eMSP tokens.

Check Tokens	**ToIOP**	An eMSP requests IOP to get its tokens stored in IOP.

4.1.3.1 Local authorisation

Use case	ToIOP/FromIOP	Usage
Realtime Authorisation	**FromIOP**	IOP, triggered by a get authorisation request coming from a CPO, requests an eMSP to get the authorisation for a charging session.

4.1.3.2 Remote Authorisation

Use case	ToIOP/FromIOP	Usage
Start Session	**ToIOP**	An eMSP requests IOP to start a charging-session remotely on a CPO's EVSE.
Stop Session	**ToIOP**	An eMSP requests IOP to stop remotely a charging-session. This use case can happen even if the charging-session beginning with an authorisation per RFID badge.

4.1.3.3 Session

Use case	ToIOP/FromIOP	Usage
Push Session	**FromIOP**	IOP requests an eMSP to initialise or update an OCPI Session object following an authorised Authorisation.
Pull Session	**ToIOP**	An eMSP requests IOP to get Sessions that belong to it.

4.1.3.4 CDR

Use case	ToIOP/FromIOP	Usage
Push CDR	**FromIOP**	IOP sends to an eMSP a CDR related to a terminated charging-session.
Pull CDR	**ToIOP**	An eMSP requests IOP to get CDRs that belongs to it.

4.1.3.5 Tariffs

Use case	ToIOP/FromIOP	Usage
Pull Tariffs	**ToIOP**	An eMSP requests IOP to get Tariffs related to CPO it is in contract with.

4.2 Use cases required by GIREVE

Some of these use cases are required when connecting to GIREVE.

4.2.1 Always required

Use case		Why?
Register/FromIOP **Register/ToIOP**	**OR**	These use cases are needed to initialise connection between an operator and GIREVE

4.2.2 If the eMSP implements the "Locations static data download" feature

Use case	Why?
Pull Locations/ToIOP	Only way for an operator to retrieve all Locations.

4.2.3 If the eMSP implements the "Locations dynamic data download" feature

Use case	Why?
Push EVSE status/FromIOP	Only way for an operator to be alerted about EVSE status change in realtime.

4.2.4 If the eMSP implements the "Roaming" feature

Use case	Why?
Pull Tokens/FromIOP	An eMSP has no possibility to send a single request to update a list of Tokens. The Pull Tokens feature allows IOP to retrieve all Tokens in a few numbers of requests instead of receiving many single updates.
Push CDR/FromIOP OR Pull CDR/ToIOP	Getting CDR is mandatory to enable roaming

a. If the eMSP implements the "realtime authorisations" feature

Use case	Why?
Realtime Authorisation/FromIOP	Required by GIREVE if eMSP's TokenTypes are "NEVER" or "ALLOWED-*"

4.2.4.1 If the eMSP implements the "remote authorisations" feature

Use case	Why?
Start Session/ToIOP	Required to start a charging-session remotely
Stop Session/ToIOP	Required to stop a charging-session remotely

4.2.4.2 If the eMSP implements the "Sessions" feature

Use case	Why?
Push Session/FromIOP OR Pull Session/ToIOP	Required to be alerted about charging-sessions status changes.

4.3 Connection & Register specifications

IOP follows the OCPI standard for Connection & Register process. *See OCPI specifications.*

4.3.1 Update Credentials FromIOP

IOP is not able to send a PUT Credentials to update its Credentials on operator backend. In order to update them, IOP sends a "DELETE Credentials" to delete the OCPI connection then starts a new Connection & Register process beginning by a POST Credentials sent by IOP or by the operator.

4.4 Locations module specifications

IOP follows the OCPI standard for Locations downloaded by an eMSP. *See OCPI specifications*.

4.4.1 Static and dynamic attributes

The attributes of the Location object (with EVSE and Connector) are of 2 types :

- Static attributes are data attributes that do not change frequently (address, localisation …).
- Dynamic attributes are data attributes that may change frequently (availability, occupied/free …).

For the moment, all properties of a Location are Static data except EVSE.status and CONNECTOR.tariff_id, the only Dynamic data.

4.4.2 PULL Locations ToIOP: Who is the CPO?

The CPO of a Location is not a direct property of the Location object (String "name" is not strongly consistent). So, an eMSP pulling IOP to get Locations does not know to which CPO the Location refers to. That is why IOP replaces the property "Location.operator.name" by the eMI3 Id of the CPO when an eMSP get a Location. Using this property, the eMSP is able to know who the CPO of the Location is.

4.4.3 PULL Locations: Retrieve Locations of a single given CPO

The standard OCPI 2.1.1 Locations pulling allows eMSPs to get Locations of all CPOs in contract with them.

In some cases, eMSPs need only Locations of a specific CPO. For example, when the eMSP initializes data of a CPO after signature of a new roaming agreement.

GIREVE provides a new OCPI 2.1.1 feature by allowing the eMSP to get Locations of a unique CPO by filling two dedicated OCPI headers in their "GET Locations" request to GIREVE:

- ocpi-to-country-code: The country code of the CPO targeted.
- ocpi-to-party-id: The party id of the CPO targeted.

Therefore, eMSPs can request GIREVE without these headers to get Locations of all CPOs or including these headers to get Locations of a unique CPO.

For information, these headers have been included in the version 2.2 of the OCPI standard.

4.4.4 PULL Locations: OCPI-2.2.1 CPO

If the "publish" property of the OCPI-2.2.1 CPO's Location is set to "false", Gireve doesn't send Location to the eMSP.

➔ *For OCPI 2.1.1 there is no "published" property, so OCPI 2.1.1 eMSPs cannot distinguish "publish" false or true.*

➔ *By default, only "publish" true is sent.*

4.4.5 PULL Locations ToIOP: Plug&Charge (P&C): Connector Object - new attribute "capabilities"

The ability to enable Plug & Charge (P&C) functionality, eliminating the need for a physical badge, offers a significant improvement in the user experience for EV drivers.

In alignment with the OCPI 2.3 white paper, which recommends adding Plug & Charge capabilities at the *Connector* level rather than the *EVSE* level (as currently defined in OCPI 2.1.1 and 2.2.1), Gireve has implemented a mechanism enabling CPOs to inform eMSPs about EVSE compatibility with Plug & Charge.

To achieve this, Gireve has introduced a new attribute, capabilities, at the Connector object level. This attribute accepts the following list of values:

OCPI Field	Enum	Description
Location.evse.connector.capabilities	**ISO_15118_2_PLUG_AND_CHARGE**	Compatibility of an EVSE with PnC using ISO15118-2
Location.evse.connector.capabilities	**ISO_15118_20_PLUG_AND_CHARGE**	Compatibility of an EVSE with PnC using ISO15118-20

eMSPs **should accept these two new capabilities** when they are provided by Gireve.

4.4.6 PULL Locations ToIOP: Get Object

If the eMSP wants to retrieve a specific given Location, EVSE or Connector, it can call these URLs:

- /ocpi/cpo/2.2.1/locations/{location_id}
- /ocpi/cpo/2.2.1/locations/{location_id}/{evse_uid}
- /ocpi/cpo/2.2.1/locations/{location_id}/{evse_uid}/{connector_id}

With:

- location_id: unique id of the Location object provided by IOP.
- evse_uid: unique id of the EVSE object provided by IOP.
- connector_id: unique id of the Connector object provided by IOP.

Gireve strongly advises partners not to use them because we do not provide support for those flows, and we do not guarantee any service level on them.

4.4.7 PULL Locations ToIOP: Get List Pagination

If the eMSP wants to retrieve a list of Locations, it can call the URL: /ocpi/cpo/2.1.1/locations using the paginated properties date_from, date_to, offset and limit.
Parameters « offset » and « limit » are optional but IOP always returns a paginated response (subset of objects list and link, X-Total-Count and X-limit headers).
The eMSP must call **the link returned in the headers** to get the next pages.

IOP has its own max limit (**100 Locations**) and answers with its if the client limit is upper than IOP one or the client doesn't set its limit.

4.4.8 PULL Locations ToIOP: Get List, Full and Delta modes

If the eMSP wants to retrieve all the Locations, it should not include « date_from » parameter in its request. But if it wants to get only changes (optimised), it should use it.

NB: The Location and EVSE deletion logic is different if using « date_from » parameter or not. When « date_from » is present, the eMSP get EVSEs of the Location in status « REMOVED » whereas without « date_from » the deleted items are not included in the response (see table below).

For information, when eMSP PULL Locations list from IOP, IOP follows the below logic in responses provided :

	LOCATION	EVSE	CONNECTOR
CREATE	New Location is part of the response.	New EVSE is part of the response.	New Connector is part of the response.
UPDATE	Updates are parts of the response.	Updated EVSE is part of the response.	Updated Connector is part of the response.
DELETE	- Using « date_from » option, response contains all EVSEs of the Location with « REMOVED » value for « status » field. - Without « date_from » option, the response does not contain this Location.	- Using « date_from » option, response contains the EVSE with « REMOVED » value for « status » field. - Without « date_from » option, the response does not contain this EVSE.	Deleted Connectors are not part of the response.

4.4.9 PULL Locations ToIOP: evse_id

It may happen that evse_id may not be compliant with the eMI3 standard. (See “Part 2 v1.0” of eMI3 standard here: http://emi3group.com/documents-links/).

You should not reject them because of the non-compliance with the standard.

4.4.10 PUSH Locations FromIOP

When a change occurs on a Location, EVSE or Connector, IOP notifies eMSP following this logic:

	LOCATION	EVSE	CONNECTOR
CREATE	Location is pushed by IOP.	EVSE is pushed by IOP.	Connector appears in EVSE description pushed by IOP.
UPDATE	Each EVSE of the updated Location are pushed by IOP.	EVSE is pushed by IOP.	EVSE containing Connector updates is pushed by IOP.
DELETE	Each EVSE of the deleted Location are pushed by IOP with « REMOVED » value for « status » field.	EVSE containing « REMOVED » value for « status » field is pushed by IOP.	EVSE description is pushed by IOP, it doesn’t contain the Connector anymore.

When IOP pushes a notification on static attributes to eMSP, IOP only uses PUT method.

As a reminder, Gireve doesn’t send OCPI-2.2.1 CPO’s Location to the eMSP if the “publish” property of the Location is set to “false”. But the eMSP can still receive dynamic data from these

locations (patch evse status) and, as for any evci the eMSP doesn't know, simply ignore it (standard behavior).

4.4.11 PULL Locations FromIOP

IOP will never PULL eMSP to check the status of a Location, EVSE or Connector object in the eMSP system.

4.5 Tokens module specifications

IOP follows the OCPI standard for Tokens module. *See OCPI specifications*.

4.5.1 Share Tokens list

OCPI requires eMSPs to transfer their tokens list to CPOs. The eMSP can choose between pulling and pushing its tokens, IOP supports both.

4.5.2 PULL Tokens FromIOP: Pagination and periodicity

IOP is able to PULL Tokens requesting eMSP backend. In this case, IOP uses the pagination and eMSP must respond with a paginated response. If the response is not paginated, it will be ignored by IOP.
The default periodicity is every day.

4.5.3 POST Authorize request: LocationReferences mandatory

In OCPI, the body of a POST Authorize request can contain a LocationReferences object.
For IOP, the LocationReferences object is mandatory.
IOP previously required that CPOs send only one “evse_uid” in the real-time authorization request. This limit is now removed, CPOs can send 0 to N “evse_uid” in the request.

Please note that a CPO can send a real-time authorization request with **0 or N evse_uid values** in the payload.
In such cases, Gireve will **randomly select an EVSE** belonging to the specified location and forward the request to the eMSP. The EVSE chosen by Gireve may **not necessarily be the one actually used** in the authorization process.
However, the CPO is **required to use the correct EVSE** in both the **Sessions** and **CDRs** objects.

4.5.4 POST Authorize request: new attribute « authorization_id »

When an eMSP answers to a POST Token Authorize request, GIREVE highly recommends eMSPs to use the new attribute of AuthorizationInfo object, the “authorization_id”.
Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

4.6 Commands module specifications

IOP follows the OCPI standard for Commands received by a CPO. *See OCPI specifications*.

4.6.1 StartSession request: new attribute « authorization_id »

When an eMSP sends a StartSession command to a CPO, GIREVE highly recommends eMSPs to use the new attribute “authorization_id”. Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

4.6.2 ReserveNow command

The ReserveNow command is not yet implemented by IOP.

4.6.3 UnlockConnector command

The UnlockConnector is not yet implemented by IOP.

4.6.4 “evse_uid” mandatory in StartSession command.

In OCPI 2.1.1 standard, the “evse_uid” property is optional for StartSession command.
IOP requires it to do the mapping with the eMIP protocol.

4.7 Sessions module specifications

IOP follows the OCPI standard for Sessions sent by a CPO. *See OCPI specifications*.

4.7.1 Usage of “authorization_id”

If an “authorization_id” has been provided by the eMSP during the Authorisation, this information will be defined in all Sessions PUT/PATCH requests sent to the eMSP. It allows eMSPs to link the Session with its previous Authorisation.
Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

4.7.2 PULL Sessions ToIOP: Get List Pagination

If the eMSP wants to retrieve a list of Sessions, it can call the URL: /ocpi/cpo/2.1.1/sessions?date_from= using the paginated properties date_from, date_to, offset and limit.

Parameters « offset » and « limit » are optional but IOP always returns a paginated response (subset of objects list and link, X-Total-Count and X-limit headers).
The eMSP must call **the link returned in the headers** to get the next pages.

IOP has its own max limit (**20 Sessions**) and answers with its if the client limit is upper than IOP one or the client doesn’t set its limit.

The timeframe (‘date_from’ and ‘date_to’) used in the request parameters must be lower than **one month**.

4.8 CDRs module specifications

IOP follows the OCPI standard for CDRs sent by a CPO. *See OCPI specifications*.

4.8.1 Usage of "authorization_id"

If an "authorization_id" has been provided by the eMSP during the Authorisation, this information will be defined in all CDRs requests sent to the eMSP. It allows eMSPs to link the CDR with its previous Authorisation.

Please refer to paragraph *2.4.2 New attribute « authorization_id »*.

4.8.2 CDR content

The "total_time" value is the total duration of this session (including the duration of charging and not charging).

It might not include the duration during which the EVSE is out of order so cannot supply the service. The out of order duration should be free of charge for eMSPs.

4.8.3 Add billing information in "Remark" field

In some cases, the CPO is not able to send a consistent B2B price in the CDR. ("total_cost" not mandatory for CPOs connected through eMIP protocol, CPO not able to calculate the price, …)

For these reasons GIREVE has implemented a billing feature, included in its "Clearing" service, which calculates the B2B price for a given CDR and injects this information in the CDR.

IOP uses the "Remark" field in CDRs sent to the eMSP to send extra information about this billing.

The "Remark" field is prefixed by :

- [*] prefix: The total_cost provided in the CDR is calculated by GIREVE.
- [?] prefix: The total_cost provided in the CDR is not significant. The B2B price has not been sent by the CPO and/or the eMSP has not subscribed to GIREVE Clearing service.
- No prefix: The total_cost provided in the CDR has been sent by the CPO.

4.8.4 Get the signed data (Calibration Law / Eichrecht)

GIREVE has extended OCPI 2.1.1 for eMSPs to receive signed data in CDRs, aiming to be compliant with the German calibration law (Eichrecht).

eMSPs can ask GIREVE to activate the transfer of these signed data, in the case where the CPO sends them in CDRs.

These signed data are also included in CDRs following the OCPI 2.2 specifications.

4.8.5 PULL CDRs ToIOP: Get List Pagination

If the eMSP wants to retrieve a list of CDRs, it can call the URL: /ocpi/cpo/2.1.1/cdrs?date_from= using the paginated properties date_from, date_to, offset and limit.

Parameters « offset » and « limit » are optional but IOP always returns a paginated response (subset of objects list and link, X-Total-Count and X-limit headers).

The eMSP must call **the link returned in the headers** to get the next pages.

IOP has its own max limit (**20 CDRs**) and answers with its if the client limit is upper than IOP one or the client doesn't set its limit.

The timeframe ('date_from' and 'date_to') used in the request parameters must be lower than **one month**.

4.8.6 PULL CDRs ToIOP: Timeout Error

In the event of a timeout during a Get CDRs request (HTTP 500), the eMSP may retry the request for up to one hour, leveraging a caching mechanism.

Gireve recommends waiting at least 10 minutes before retrying. Otherwise, the eMSP may encounter the following error response:

```
{
   "status_code": 2914,
   "status_message": "A pulling already ran for a similar request",
   "timestamp": "2026-03-25T14:22:01Z"
}
```

In such cases, the eMSP should retry the same request.

Important: For the caching mechanism to apply, all retry requests must use exactly the same parameters as the initial request (date_from, date_to, offset, and limit). Any variation will be treated as a new request, and the cache will not be utilized.

4.9 Tariffs module specifications

IOP follows the OCPI standard for Tariffs download by an eMSP. *See OCPI specifications*

4.9.1 Tariffs flows implemented by GIREVE

IOP only implements the "GET Tariffs" used by an eMSP to retrieve tariffs of CPOs.

4.9.2 PULL Tariffs ToIOP: Get List Pagination

If the eMSP wants to retrieve a list of Tariffs, it can call the URL: /ocpi/cpo/2.1.1/tariffs using the paginated properties date_from, date_to, offset and limit.

Parameters « offset » and « limit » are optional but IOP always returns a paginated response (subset of objects list and link, X-Total-Count and X-limit headers).
The eMSP must call **the link returned in the headers** to get the next pages.

IOP has its own max limit (**100 Tariffs**) and answers with its if the client limit is upper than IOP one or the client doesn't set its limit.

4.9.3 PULL Tariffs ToIOP: Get List, Full and Delta modes

If the eMSP wants to retrieve all the Tariffs, it should not include « date_from » parameter in its request. But if it wants to get only changes (optimised), it should use it.

4.9.4 Specific properties added by GIREVE

a. *Tariffs.country_code and Tariffs.party_id (mandatory)*

In the same way as the OCPI 2.2 standard, GIREVE includes in Tariffs objects returned to eMSPs the identification of the owner of the Tariffs (the CPO).
This identification is done by adding two new properties in each tariff :

- country_code: Country code of the CPO owner of the tariff.
- party_id: Party id of the CPO owner of the tariff.

b. *TariffDimensionType "SESSION_TIME"*

In standard OCPI V2.1.1, the CPO can define 2 different "TariffDimensionType" related to the duration :

- **TIME**: "time charging: defined in hours, step_size multiplier: 1 second" *(description from the OCPI Github)*
- **PARKING_TIME**: "time not charging: defined in hours, step_size multiplier: 1 second" *(description from the OCPI Github)*

GIREVE adds a third "TariffDimensionType" named "SESSION_TIME" with the following description :

- **SESSION_TIME**: "time charging or not: defined in hours, step_size multiplier: 1 second".

4.9.4.1 Tariffs.restriction (optional)

If the CPO describes its tariffs in the roaming agreement on the GIREVE connect-place, the CPO can define different tariffs depending on the timeslots of the start of the charging session.

As an example:

- The Tariff A is applicable if the charge begins "between Monday and Friday" **and** on the timeslots "8 AM to 12 AM" or "2 PM to 6 PM".
- If the charge starts outside of these time and day slots, the Tariff B is applicable.

If a tariff described on the GIREVE connect-place depends on time and day of the charging session start, GIREVE adds these criteria in the tariff description sent to eMSPs through the OCPI Tariffs module.

As for the property “TariffElement.restrictions”, the “Tariffs.restrictions” contains several properties :

- **start_time**: Start time of the first timeslot, for example 13:30, valid from this time of the day. Must be in 24h format with leading zeros. Hour/Minute separator: ":" Regex: ([0-1][0-9]|2[0-3]):[0-5][0-9].
- **end_time**: End time of the first timeslot, for example 13:30, valid from this time of the day. Must be in 24h format with leading zeros. Hour/Minute separator: ":" Regex: ([0-1][0-9]|2[0-3]):[0-5][0-9].
- **start_time_2**: Start time of the second timeslot, for example 13:30, valid from this time of the day. Must be in 24h format with leading zeros. Hour/Minute separator: ":" Regex: ([0-1][0-9]|2[0-3]):[0-5][0-9].
- **end_time_2**: End time of the second timeslot, for example 13:30, valid from this time of the day. Must be in 24h format with leading zeros. Hour/Minute separator: ":" Regex: ([0-1][0-9]|2[0-3]):[0-5][0-9].
- **day_of_week**: Which day(s) of the week this tariff is valid.

To determine which tariff is applicable for a charging session, the rule is: **“The tariff containing a restriction has priority other tariffs without restrictions”.**

4.9.4.2 PriceComponent.price_round (mandatory)

If the CPO describes its tariffs in the roaming agreement on the GIREVE connect-place, the CPO defines the rounding of the “TariffElement”.

This rounding is applied for each “TariffElement” after the calculation of “the energy delivered in the TariffElement” or “the time spent in the TariffElement”.

This rounding has 2 properties :

- **round_granularity**: Can take values “UNIT”, “TENTH”, “HUNDRETH” or “THOUSANDTH”.
- **round rule**: Can take values “ROUND_UP”, “ROUND_DOWN” or “ROUND_NEAR”.

For tariffs coming from CPOs through the OCPI Tariffs module, the rounding is not defined and the price properties can contain until 4 decimals so the default rounding applied by GIREVE is “round_granularity: thousandth” and “round_rule: round_near”.

4.9.4.3 PriceComponent.step_round (mandatory)

If the CPO describes its tariffs in the roaming agreement on the GIREVE connect-place, the CPO defines the rounding of the number of “step_size”.

This rounding is applied for each “TariffElement” after the calculation of “the energy delivered in the TariffElement” or “the time spent in the TariffElement”.

This rounding has 2 properties :

- **round_granularity**: single option “UNIT”
- **round_rule**: Can take values “ROUND_UP”, “ROUND_DOWN” or “ROUND_NEAR”.

For tariffs coming from CPOs through the OCPI Tariffs module, the rounding is standard and take values “round_granularity: UNIT” and “round_rule: ROUND_UP”.

4.9.4.4 PriceComponent.exact_price_component (mandatory)

When a price_component cannot be exactly mapped from the GIREVE connect-place to the OCPI 2.1.1 format, GIREVE informs the eMSP by adding the property “exact_price_component” taking values :

- **true**: The price_component is equal to the one declared by the CPO.
- **false**: The original price_component cannot be mapped exactly with OCPI 2.1.1, an approximation has been made.

This new property will be consistent too for the future mapping between different versions of OCPI.

4.9.4.5 min_price (optional)

When this property is set, a Charging Session with this tariff will at least cost this amount. When the cost of a Charging Session is lower than this amount, the cost of the Session will be equal to this amount.

4.9.4.6 max_price (optional)

When this property is set, a Charging Session with this tariff will NOT cost more than this amount.

4.9.4.7 Synthesis of the GIREVE specifications

Tariff			
Property	**Type**	**Card.**	**Description**
country_code	String(2)	1	Country code of the CPO owner of the tariff
party_id	String(3)	1	Party id of the CPO owner of the tariff
restriction	Object containing up to 4 properties: • start_time • end_time • start_time_2 • end_time_2	?	Time and day the tariff is applicable. To determine which tariff is applicable for a charging session, the rule is: **“The tariff containing a restriction has**

	• day_of_week		**priority other tariffs without restrictions".**
min_price	Float	?	When the cost of a Charging Session is lower than this amount, the cost of the Session will be equal to this amount.
max_price	Float	?	The charging Session will not cost more than this amount.
Tariff.elements.price_components			
type	Add "SESSION_TIME" as a new "TariffDimensionType"	1	**TIME**: "time charging: defined in hours, step_size multiplier: 1 second" *(description from the OCPI Github)* **PARKING_TIME**: "time not charging: defined in hours, step_size multiplier: 1 second" *(description from the OCPI Github)* **SESSION_TIME**: "time charging or not: defined in hours, step_size multiplier: 1 second"
price_round	Object containing 2 properties: • round_granularity (« UNIT », « TENTH », « HUNDRETH », « THOUSANDTH »). • round_rule (« ROUND_UP », « ROUND_DOWN », « ROUND_NEAR »)	1	Rounding of the price applied for each "TariffElement" after the calculation of "the energy delivered in the TariffElement" or "the time spent in the TariffElement".
step_round	Object containing 2 properties: • round_granularity (« UNIT »). • round_rule (« ROUND_UP », « ROUND_DOWN », « ROUND_NEAR »)	1	Rounding of the number of step_size to apply after the calculation of "the energy delivered in the TariffElement" or "the time spent in the TariffElement".
exact_price_component	Boolean	1	Boolean informing if the "price_component" description corresponds exactly with the one sent/described by the CPO.

5 EXAMPLES OF OCPI REQUESTS

5.1 Credentials & Register

5.1.1 ToIOP_GET_emsp_versions, FromIOP_GET_emsp_versions, ToIOP_GET_cpo_versions, FromIOP_GET_cpo_versions

Request

VERB: GET

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}

BODY:

Response

CODE: 200

HEADERS: {Transfer-Encoding:chunked}{Date:Wed, 08 Jan 2020 10:24:52 GMT}{Connection:Close}{Content-Type:application/json}

BODY:


```
{
	"data": [
		{
			"version": "2.1.1",
			"url": "https://DomainName.com/ocpi/emsp/2.1.1"
		}
	],
	"status_code": 1000,
	"status_message": "Success",
	"timestamp": "2020-01-08T10:24:52Z"
}
```


5.1.2 ToIOP_GET_emsp_version_detail_2.1.1, FromIOP_GET_emsp_version_detail_2.1.1, ToIOP_GET_cpo_version_detail_2.1.1, FromIOP_GET_cpo_version_detail_2.1.1

Request

VERB: GET

HEADERS: {connection:close}{authorization:Token xxx-xxx-xxx}

BODY:

Response

CODE: 200

HEADERS: {Date:Thu, 16 Jan 2020 07:30:16 GMT}{Connection:close}{Content-Type:application/json}

BODY:


```
{
  "data": {
```

```
    "endpoints": [
      {
        "identifier": "credentials",
        "url": "https://DomainName.com/ocpi/emsp/2.1.1/credentials"
      },
      {
        "identifier": "locations",
        "url": "https://DomainName.com/ocpi/emsp/2.1.1/locations"
      },
      {
        "identifier": "tokens",
        "url": "https://DomainName.com/ocpi/emsp/2.1.1/tokens"
      },
      {
        "identifier": "commands",
        "url": "https://DomainName.com/ocpi/emsp/2.1.1/commands"
      },
      {
        "identifier": "sessions",
        "url": "https://DomainName.com/ocpi/emsp/2.1.1/sessions"
      },
      {
        "identifier": "cdrs",
        "url": "https://DomainName.com/ocpi/emsp/2.1.1/cdrs"
      }
    ],
    "version": "2.1.1"
  },
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-08T10:24:52Z"
}
```


5.1.3 ToIOP_POST_emsp_credentials_2.1.1, FromIOP_POST_emsp_credentials_2.1.1, ToIOP_POST_cpo_credentials_2.1.1, FromIOP_POST_cpo_credentials_2.1.1

Request

VERB: POST

HEADERS: {content-type:application/json; charset=UTF-8}{connection:close}{accept:application/json, application/*+json}{authorization:Token xxx-xxx-xxx}

BODY:


```
{
  "business_details": {
```

```
    "logo": {
      "category": "NETWORK",
      "height": 44,
      "thumbnail": "http://DomainName.com/yyyy.png",
      "width": 92,
      "type": "png",
      "url": " http://DomainName.com/logo.png"
    },
    "website": "http://DomainName.com",
    "name": "NAME"
  },
  "token": "aaa-xxx-eee",
  "party_id": "PID",
  "country_code": "FR",
  "url": "https://DomainName.com/ocpi/emsp/versions"
}
```


Response

CODE: 200

HEADERS: {Date:Thu, 16 Jan 2020 07:30:16 GMT}{Connection:close}{Content-Type:application/json}

BODY:


```
{
  "data": {
    "url": "http://DomainName.com/ocpi/versions",
    "token": "eee-fff-ddd",
    "party_id": "AAA",
    "country_code": "DE",
    "business_details": {
      "name": "NAME"
    }
  },
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-14T09:10:21Z"
}
```


5.1.4 ToIOP_DELETE_emsp_credentials_2.1.1, FromIOP_DELETE_emsp_credentials_2.1.1, ToIOP_DELETE_cpo_credentials_2.1.1, FromIOP_DELETE_cpo_credentials_2.1.1

Request

VERB: DELETE

HEADERS: {content-type:application/json; charset=UTF-8}{connection:close}{accept:application/json, application/*+json}{authorization:Token xxx-xxx-xxx}

BODY:

Response
CODE: 200
HEADERS: {Date:Thu, 16 Jan 2020 07:30:16 GMT}{Connection:close}{Content-Type:application/json}
BODY:

```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-17T09:39:42Z"
}
```


5.2 Locations

5.2.1 ToIOP_GET_cpo_locations_2.1.1, FromIOP_GET_cpo_locations_2.1.1

URL
/ocpi/cpo/2.1.1/locations?date_from=2020-01-20T14:51:43Z&limit=20&offset=10
(Date_from parameter shall be the date of the last successful request)
Request
VERB: GET
HEADERS: {connection:close}{accept:application/json, application/*+json}{authorization:Token xxx-xxx-xxx}
+ {ocpi-to-party-id:CPO}{ocpi-to-country-code:FR} (If you wish to get the locations of a specific CPO)
BODY:

Response
CODE: 200
HEADERS: {Date:Thu, 16 Jan 2020 07:30:16 GMT}{Connection:close}{Content-Type:application/json} {Link:<https://DomainName.com/ocpi/cpo/2.1.1/locations?date_from=2020-01-20T14:51:43Z&limit=20&offset=30>;rel="next"} {X-Total-Count:11682} {X-Limit:20}
BODY:

```
{
  "data": [{
    "id": "azerty",
    "type": "UNKNOWN",
    "name": "Name of the Location",
    "address": "Adress of the Location ",
    "city": "City of the Location ",
    "country": "ESP",
    "coordinates": {
      "latitude": "11.222222",
      "longitude": "11.222222"
    },
```

```
  "evses": [
   {
    “uid”: “33656”,
    "status": "AVAILABLE",
    "capabilities": [
     "REMOTE_START_STOP_CAPABLE",
     "RFID_READER"
    ],
    "connectors": [
     {
      "id": "UIOP",
      "standard": "IEC_62196_T2",
      "format": "SOCKET",
      "voltage": 230,
      "amperage": 32,
      "power_type": "AC_1_PHASE",
      "last_updated": "2019-11-21T10:43:49Z"
     }
    ],
    "coordinates": {
     "latitude": "11.222222",
     "longitude": "11.222222"
    },
    "evse_id": "ES*AAA*E11111",
    "last_updated": "2019-11-21T10:43:49Z"
   }
  ],
  "operator": {
   "name": "ES*AAA"
  },
  "postal_code": "12345",
  "related_locations": [
   {
    "latitude": "11.222222",
    "longitude": "11.222222"
   }
  ],
  "opening_times": {
   "twentyfourseven": true
  },
  "last_updated": "2019-11-21T10:43:49Z"
 }],
 "status_code": 1000,
 "status_message": "Success",
```

"timestamp": "2020-01-15T18:56:08Z"
}

5.2.2 ToIOP_PUT_emsp_locations_2.1.1, FromIOP_ PUT_emsp_locations_2.1.1 (on Locations)

URL

/ocpi/emsp/2.1.1/locations/FR/CPO/1111

(FR and CPO refer to the country_code and party_id of the CPO which owns the Location cf. 2.1.3 Client owned object push*)*

Request

VERB: PUT

HEADERS: {content-type:application/json; charset=UTF-8}{connection:close}{accept:application/json, application/*+json}{authorization:Token xxx-xxx-xxx}

BODY:

{
"id": "eeee",
"type": "UNKNOWN",
"name": "name ",
"address": "address",
"city": "city",
"postal_code": "pc",
"country": "FRA",
"coordinates": {
"latitude": "11.111111",
"longitude": "11.111111"
},
"related_locations": null,
"evses": [
{
"uid": "aaaaaa",
"evse_id": "FR*CPO*E111",
"status": "AVAILABLE",
"status_schedule": null,
"capabilities": [],
"connectors": [
{
"id": "QSDF",
"standard": "IEC_62196_T2",
"format": "SOCKET",
"power_type": "AC_3_PHASE",
"voltage": 230,
"amperage": 16,

```
        "tariff_id": "A3",
        "terms_and_conditions": null,
        "last_updated": "2020-01-09T09:45:59Z"
       }
     ],
     "floor_level": "",
     "coordinates": null,
     "physical_reference": null,
     "directions": [],
     "parking_restrictions": null,
     "images": null,
     "last_updated": "2020-01-16T13:25:20Z"
   }
 ],
 "directions": [],
 "operator": {
   "name": "FR*CPO",
   "website": "https://DomainName.com",
   "logo": {
     "url": "https://DomainName.com/logo.png",
     "thumbnail": "https://DomainName.com/img/logo_thumb.png",
     "category": "OPERATOR",
     "type": "png",
     "width": 0,
     "height": 0
   }
 },
 "suboperator": null,
 "owner": null,
 "facilities": [],
 "time_zone": null,
 "opening_times": null,
 "charging_when_closed": false,
 "images": null,
 "energy_mix": null,
 "last_updated": "2020-01-16T13:59:37Z"
}
```


Response

CODE: 200

HEADERS: {Date:Thu, 16 Jan 2020 07:30:16 GMT}{Connection:close}{Content-Type:application/json}

BODY:


```
{
 "data": {},
 "status_code": 1000,
```

"status_message": "Success",
"timestamp": "2020-01-16T14:26:27Z"
}

5.2.3 ToIOP_PATCH_emsp_locations_2.1.1, FromIOP_PATCH_emsp_locations_2.1.1 (on EVSE)

URL

/ocpi/emsp/2.1.1/locations/FR/CPO/1111/22222

(FR and CPO refer to the country_code and party_id of the CPO which owns the Location cf. 2.1.3 Client owned object push*)*

Request

VERB: PATCH

HEADERS: {content-type:application/json; charset=UTF-8}{connection:close}{accept:application/json, application/*+json}{authorization:Token xxx-xxx-xxx}

BODY:

{
"status": "AVAILABLE",
"last_updated": "2020-01-16T15:18:52Z"
}

Response

CODE: 200

HEADERS: {Date:Thu, 16 Jan 2020 07:30:16 GMT}{Connection:close}{Content-Type:application/json}

BODY:

{
"data": {},
"status_code": 1000,
"status_message": "Success",
"timestamp": "2020-01-16T14:26:27Z"
}

5.3 Tokens

5.3.1 ToIOP_GET_emsp_tokens_2.1.1, FromIOP_GET_emsp_tokens_2.1.1

URL

/2.1.1/tokens?date_from=2020-01-13T12%3A54%3A25&offset=0&limit=50

(Date_from parameter shall be the date of the last successful request)

Request

VERB: GET

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}

BODY:

Response

CODE: 200

HEADERS: {Link:<https://DomainName.com/ocpi/emsp/2.1.1/tokens?date_from=2020-01-13T12:54:25Z&offset=50&limit=50>;rel="next"}{X-Limit:50}{Content-Type:application/json}{X-Total-Count:10627}

BODY:


```
{
"data": [
        {
                "uid": "aaa",
                "valid": true,
                "issuer": "FR*AAA",
                "auth_id": "AZERTY",
                "last_updated": "2020-01-13T12:55:01Z",
                "language": "FR",
                "type": "OTHER",
                "whitelist": "ALLOWED_OFFLINE"
        },
        {
                "uid": "bbb",
                "valid": true,
                "issuer": "FR*BBB",
                "auth_id": "QSDFG",
                "last_updated": "2020-01-13T13:01:53Z",
                "type": "RFID",
                "whitelist": "NEVER",
                "visual_number": "AZSCV"
        }
],
"status_code": 1000,
"status_message": "Success",
"timestamp": "2020-01-17T09:54:25Z"
}
```


5.3.2 ToIOP_PUT_cpo_tokens_2.1.1

URL

/ocpi/cpo/2.1.1/tokens/FR/EMP/AAAAAAAA

(FR and EMP refer to the country_code and party_id of the eMSP which owns the Token cf. 2.1.3 Client owned object push*)*

Request

VERB: PUT

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}

BODY:

```
{
        "uid": "AAAAAAAA",
        "type": "RFID",
        "auth_id": "111111",
        "visual_number": "22222",
        "issuer": " FR*AAA ",
        "valid": true,
        "whitelist": "ALWAYS",
        "language": “FR”,
        "last_updated": "2020-01-21T10:42:30Z"
}
```

Response

CODE: 200

HEADERS: {Content-Type:application/json}

BODY:

```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-21T10:53:12Z"
}
```

5.3.3 ToIOP_PATCH_cpo_tokens_2.1.1

URL

/ocpi/cpo/2.1.1/tokens/FR/EMP/AAAAAAAA

(FR and EMP refer to the country_code and party_id of the eMSP which owns the Token cf. 2.1.3 Client owned object push*)*

Request

VERB: PUT

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}

BODY:

```
{
  "visual_number": "2222222222",
  "last_updated": "2020-01-20T23:34:53Z"
}
```

Response

CODE: 200

HEADERS: {Content-Type:application/json}

BODY:

```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-21T10:53:12Z"
}
```


5.3.4 ToIOP_POST_emsp_tokens_2.1.1, FromIOP_POST_emsp_tokens_2.1.1

URL

/ocpi/emsp/2.1.1/tokens/AAAAAA/authorize

Request

VERB: POST

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}

BODY:


```
{
  "location_id": "11111",
  "evse_uids": [
    "FR*CPO*E111"
  ]
}
```


Response

CODE: 200

HEADERS: {Content-Type:application/json}

BODY:


```
{
  "data": {
    "allowed": "ALLOWED",
    "location": {
      "location_id": "11111",
      "evse_uids": [
        "FR*CPO*E111"
      ]
    },
    "authorization_id": "CCCC-VVVV-BBBB"
  },
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-17T10:15:18Z"
}
```

5.3.5 ToIOP_GET_emsp_tokens-unitary_2.1.1

URL

/ocpi/emsp/2.1.1/tokens/AAAAAAAA?type=RFID

Request

VERB: GET

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}

BODY:

Response

CODE: 200

HEADERS: {Content-Type:application/json}

BODY:


```
{
  "data": {
          "uid": "AAAAAAAA",
          "type": "RFID",
          "auth_id": "111111",
          "visual_number": "22222",
          "issuer": " FR*AAA ",
          "valid": true,
          "whitelist": "ALWAYS",
          "language": “FR”,
          "last_updated": "2020-01-21T10:42:30Z"
          },
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-21T10:53:12Z"
}
```


5.4 Commands

5.4.1 ToIOP_POST_cpo_commands_2.1.1, FromIOP_POST_cpo_commands_2.1.1

URL

/ocpi/cpo/2.1.1/commands/START_SESSION

Request

VERB: POST

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}

BODY:


```
{
        "response_url":  "https://DomainName.com/ocpi/emsp/2.1.1/commands/START_SESSION/111-222",
```

```
        "token": {
                "uid": "11111",
                "type": "OTHER",
                "auth_id": "22222",
                "visual_number": "33333",
                "issuer": "issuer",
                "valid": true,
                "last_updated": "2020-01-21T06:43:13Z",
                "whitelist": "ALWAYS"
        },
        “authorization_id” : “aaa-vvv”,
         "location_id": "12345",
        "evse_uid": "54321"
}
```


Response

CODE: 200

HEADERS: {Content-Type:application/json}

BODY:


```
{
  "data": {
    "result": "ACCEPTED"
  },
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-21T08:09:31Z"
}
```


5.4.2 Callback ToIOP_POST_cpo_commands_2.1.1, FromIOP_POST_cpo_commands_2.1.1

URL

/ocpi/emsp/2.1.1/commands/START_SESSION/aaaa-fffff-ggggg

Request

VERB: POST

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}

BODY:


```
{
  "result": "ACCEPTED"
}
```


Response

CODE: 200

HEADERS: {Content-Type:application/json}
BODY:

```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-06T14:51:38Z"
}
```


5.5 Sessions

5.5.1 ToIOP_PUT_emsp_sessions_2.1.1, FromIOP_PUT_emsp_sessions_2.1.1

URL

/ocpi/emsp/2.1.1/sessions/FR/CPO/AAAAAAA

(FR and CPO refer to the country_code and party_id of the CPO which owns the Session cf. 2.1.3 Client owned object push*)*

Request

VERB: PUT
HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}
BODY:

```
{
  "kwh": 0,
  "location": {
    "type": "UNKNOWN",
    "name": "name",
    "address": "address",
    "city": "city",
    "country": "NLD",
    "coordinates": {
      "latitude": "11.111111",
      "longitude": "11.111111"
    },
    "operator": {
      "name": "name"
    },
    "suboperator": {
      "name": "suboperator"
    },
    "id": "aaaa",
    "postal_code": "11111",
    "evses": [
      {
        "uid": "aaaaa",
```

```
      "status": "CHARGING",
      "capabilities": [
        "RFID_READER",
        "REMOTE_START_STOP_CAPABLE",
        "CREDIT_CARD_PAYABLE"
      ],
      "connectors": [
        {
          "standard": "CHADEMO",
          "format": "CABLE",
          "voltage": 400,
          "amperage": 125,
          "id": "1",
          "power_type": "DC",
          "last_updated": "2020-01-17T08:54:41.000Z"
        }
      ],
      "coordinates": {
        "latitude": "11.111111",
        "longitude": "11.111111"
      },
      "evse_id": "FR*CPO*E111",
      "last_updated": "2020-01-17T08:54:41.000Z"
    }
  ],
  "time_zone": "Europe/Amsterdam",
  "opening_times": {
    "twentyfourseven": true
  },
  "last_updated": "2020-01-17T08:54:41.000Z"
},
"currency": "EUR",
"status": "ACTIVE",
"id": "AAAAAAA",
"start_datetime": "2020-01-17T09:39:29.000Z",
"auth_id": "FR*EMP*11111",
"auth_method": "AUTH_REQUEST",
"charging_periods": [],
"total_cost": 0,
"last_updated": "2020-01-17T09:39:41.356Z",
"authorization_id": "mmm-ppp"
}
```


Response

CODE: 200
HEADERS: {Content-Type:application/json}
BODY:

```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-17T09:39:42Z"
}
```


5.5.2 ToIOP_PATCH_emsp_sessions_2.1.1, FromIOP_PATCH_emsp_sessions_2.1.1

URL

/ocpi/emsp/2.1.1/sessions/FR/CPO/AAAAAAA

(FR and CPO refer to the country_code and party_id of the CPO which owns the Session cf. 2.1.3 Client owned object push*)*

Request

VERB: PUT
HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}
BODY:

```
{
  "kwh": 2,
  "id": "AAAAAAA"
}
```


Response

CODE: 200
HEADERS: {Content-Type:application/json}
BODY:

```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-17T09:39:42Z"
}
```


5.6 CDRs

5.6.1 ToIOP_POST_emsp_cdrs_2.1.1, FromIOP_POST_emsp_cdrs_2.1.1

URL

/ocpi/emsp/2.1.1/cdrs

Request

VERB: POST
HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}{Content-Type:application/json}
BODY:


```
{
  "id": "AAAAA",
  "start_date_time": "2020-01-17T07:59:07",
  "stop_date_time": "2020-01-17T07:59:08",
  "auth_id": "FR*EMP*1111",
  "authorization_id": "mmm-ttt",
  "auth_method": "AUTH_REQUEST",
  "location": {
    "id": "1",
    "type": "ON_STREET",
    "name": "name",
    "address": "address",
    "city": "city",
    "postal_code": "11111",
    "country": "FRA",
    "coordinates": {
      "latitude": "11.111111",
      "longitude": "11.111111"
    },
    "evses": [
      {
        "uid": "23",
        "evse_id": "FR*CPO*E111",
        "status": "AVAILABLE",
        "capabilities": [
          "REMOTE_START_STOP_CAPABLE",
          "RFID_READER",
          "UNLOCK_CAPABLE"
        ],
        "connectors": [
          {
            "id": "1",
            "standard": "IEC_62196_T2",
            "format": "SOCKET",
            "power_type": "AC_3_PHASE",
            "voltage": 230,
            "amperage": 32,
            "last_updated": "2019-12-02T16:17:07"
          }
        ],
```

```
      "last_updated": "2019-08-26T16:46:46"
     }
    ],
    "opening_times": {
     "twentyfourseven": true
    },
    "last_updated": "2019-08-26T13:53:05"
  },
  "currency": "EUR",
  "charging_periods": [
    {
     "start_date_time": "2020-01-17T07:59:07",
     "dimensions": [
      {
       "type": "TIME",
       "volume": 0.0002
      }
     ]
    }
  ],
  "total_cost": 0,
  "total_energy": 0,
  "total_time": 0.0002,
  "last_updated": "2020-01-17T07:59:19"
}
```


Response

CODE: 200
HEADERS: {Content-Type:application/json}
BODY:


```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2020-01-17T09:39:42Z"
}
```


5.6.2 ToIOP_GET_cpo_cdrs_2.1.1, FromIOP_GET_cpo_cdrs_2.1.1

URL

/ocpi/cpo/2.1.1/cdrs?date_from=2020-01-20T02:44:00Z&offset=40&limit=20
(Date_from parameter shall be the date of the last successful request)

Request

VERB: GET
HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}

BODY:

Response

CODE: 200

HEADERS: {X-Limit:20}{Connection:close}{Content-Type:application/json}{X-Total-Count:57}

BODY:


```
{
"data": [
        {
                "total_time": 0,
                "location": {
                        "evses": [
                                {
                                        "uid": "1",
                                        "connectors": [
                                                {
                                                        "id": "1",
                                                        "amperage": 32,
                                                        "standard": "IEC_62196_T2",
                                                        "voltage": 380,
                                                        "power_type": "AC_3_PHASE",
                                                        "last_updated": "2019-12-30T09:26:13Z",
                                                        "format": "SOCKET"
                                                }
                                        ],
                                        "status": "CHARGING",
                                        "last_updated": "2019-12-30T09:26:13Z",
                                        "capabilities": [
                                                "RFID_READER",
                                                "CREDIT_CARD_PAYABLE"
                                        ],
                                        "coordinates": {
                                                "longitude": "11.111111",
                                                "latitude": "11.111111"
                                        },
                                        "floor_level": "0",
                                        "evse_id": "FR*CPO*e1111"
                                }
                        ],
                        "id": "2",
                        "address": "address,
                        "postal_code": "11111",
                        "name": "name",
                        "opening_times": {
```

```
                        "twentyfourseven": true
                },
                "last_updated": "2019-12-30T09:26:13Z",
                "type": "UNKNOWN",
                "operator": {
                        "name": "FR*CPO"
                },
                "coordinates": {
                        "longitude": "11.111111",
                                "latitude": "11.111111"
                },
                "country": "FRA",
                "city": "city"
        },
        "remark": "[?]",
        "auth_id": "1111111",
        "auth_method": "WHITELIST",
        "stop_date_time": "2020-01-20T16:17:58Z",
        "charging_periods": [
                {
                        "dimensions": [
                                {
                                        "volume": 0.0,
                                        "type": "TIME"
                                }
                        ],
                        "start_date_time": "2020-01-20T16:17:37Z"
                }
        ],
        "start_date_time": "2020-01-20T16:17:37Z",
        "authorization_id": "aaa-eee-rrr",
        "total_energy": 0.0,
        "currency": "EUR",
        "id": "111-222",
        "total_cost": 0,
        "last_updated": "2020-01-20T17:00:22Z",
        "total_parking_time": 0
},
{
        "total_time": 0.0167,
        "location": {
                "evses": [
                        {
                                "uid": "1",
```

```
                "connectors": [
                        {
                                "id": "1",
                                "amperage": 32,
                                "standard": "IEC_62196_T2",
                                "voltage": 380,
                                "power_type": "AC_3_PHASE",
                                "last_updated": "2019-12-30T09:26:13Z",
                                "format": "SOCKET"
                        }
                ],
                "status": "CHARGING",
                "last_updated": "2019-12-30T09:26:13Z",
                "capabilities": [
                        "RFID_READER",
                        "CREDIT_CARD_PAYABLE"
                ],
                "coordinates": {
                        "longitude": "11.111111",
                        "latitude": "11.111111"
                },
                "floor_level": "0",
                "evse_id": "FR*CPO*e1111"
        }
    ],
    "id": "2",
    "address": "address",
    "postal_code": "11111",
    "name": "name",
    "opening_times": {
        "twentyfourseven": true
    },
    "last_updated": "2019-12-30T09:26:13Z",
    "type": "UNKNOWN",
    "operator": {
        "name": "FR*CPO"
    },
    "coordinates": {
        "longitude": "11.111111",
            "latitude": "11.111111"
    },
    "country": "FRA",
    "city": "city"
},
```

```
                "remark": "[?]",
                "auth_id": "2222222",
                "auth_method": "WHITELIST",
                "stop_date_time": "2020-01-20T16:19:34Z",
                "charging_periods": [
                        {
                                "dimensions": [
                                        {
                                                "volume": 0.0,
                                                "type": "TIME"
                                        }
                                ],
                                "start_date_time": "2020-01-20T16:17:57Z"
                        }
                ],
                "start_date_time": "2020-01-20T16:17:57Z",
                "authorization_id": "aaa-eee-rrr2",
                "total_energy": 0.0,
                "currency": "EUR",
                "id": "eee-rrr",
                "total_cost": 0,
                "last_updated": "2020-01-20T17:04:28Z",
                "total_parking_time": 0
        }
],
"status_code": 1000,
"status_message": "Success",
"timestamp": "2020-01-21T02:47:32Z"
}
```


5.7 Tariffs

5.7.1 ToIOP_PUT_emsp_tariffs_2.1.1

URL

/ocpi/emsp/2.1.1/tariffs/FR/CPO/Tariff-1

Request

VERB: PUT

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}

BODY:


```
{
  "id": "Tariff-1",
  "currency": "EUR",
```

```
  "elements": [
    {
      "price_components": [
        {
          "type": "TIME",
          "price": 6,
          "step_size": 1
        },
        {
          "type": "FLAT",
          "price": 1,
          "step_size": 1
        },
        {
          "type": "ENERGY",
          "price": 2,
          "step_size": 1
        }
      ]
    }
  ],
  "last_updated": "2021-02-11T16:46:37Z"
}
```


Response

CODE: 200

HEADERS: {Content-Type:application/json}

BODY:


```
{
  "data": {},
  "status_code": 1000,
  "status_message": "Success",
  "timestamp": "2021-02-12T09:39:42Z"
}
```

5.7.2 ToIOP_GET_cpo_tariffs_2.1.1, FromIOP_GET_cpo_tariffs_2.1.1

URL

/ocpi/cpo/2.1.1/tariffs?offset=0&limit=10&date_from=2021-02-10T04:00:00Z&date_to=2021-02-11T04:00:00Z

(Date_from parameter shall be the date of the last successful request)

Request

VERB: GET

HEADERS: {Authorization:Token xxx-xxx-xxx}{Connection:close}{Accept:application/json}

BODY:

Response

CODE: 200

HEADERS: {X-Limit:10}{Connection:close}{Content-Type:application/json}{X-Total-Count:2}

BODY:


```
{
  "data": [
    {
      "id": "aaa-xxxx",
      "party_id": "AAA",
      "last_updated": "2015-06-29T20:39:09Z",
      "tariff_alt_url": "https://DomainName.com/yyyyy/11",
      "country_code": "FR",
      "elements": [
        {
          "restrictions": {
            "max_duration": 200000,
            "day_of_week": [
              "MONDAY",
              "TUESDAY"
               ],
          },
          "price_components": [
            {
              "price": 1,
              "step_size": 910,
              "type": "TIME"
```

```
        },
        {
          "price": 1,
          "step_size": 900,
          "type": "PARKING_TIME"
        }
      ]
    }
  ],
  "currency": "EUR"
 }
],
"status_code": 1000,
"status_message": "Success",
"timestamp": "2020-12-22T09:50:25Z"
}
```