



**OCPP 1.6-J Errata sheet**  
v1.0 Release, 2019-12-04

## Errata sheet: for the OCPP 1.6-J specification.

---

Copyright © 2010 – 2019 Open Charge Alliance. All rights reserved.

This document is made available under the *\*Creative Commons Attribution-NoDerivatives 4.0 International Public License\** (<https://creativecommons.org/licenses/by-nd/4.0/legalcode>).

---

### Version History

VERSION	DATE	AUTHOR	DESCRIPTION
v1.0 Release	2019-12-04	Robert de Leeuw <i>IHomer</i>	Release of the OCPP-J errata sheet

# 1. Scope

This document contains errata on the OCPP 1.6-J specification (JSON) and JSON-Schema files.

## 1.1. JSON Schema files

With this release of this version of this erratasheet (v1.0) also new OCPP 1.6 JSON Schema files are released.

Version information has been added to all the files: ID field. For this version they are marked with: "urn:OCPP:1.6:2019:12"

## 1.2. Terminology and Conventions

Underlined: when needed to clarify differences, they might be underlined.

# 2. Major errata

Problems with the content/definition of the messages, class and enumerations of the protocol.

## 2.1. MeterValues.json and StopTransaction.json incorrect spelling of Celsius

The OCPP 1.6 JSON schema message definitions set, as originally published, contains a typo that renders it inconsistent with both the written OCPP 1.6 specification and the SOAP/XML Schema/WSDLs schema definitions.

In the MeterValues and StopTransaction message structure (in MeterValues.json and StopTransaction.json), the definition of Celsius as a unit has been misspelled as: "Celcius".

We advise implementers of an OCPP 1.6 JSON Central System to add the correct spelling (Celsius) to the enumeration. This way your implementation will support both the correct and the incorrect spellings. Somebody implementing a OCPP 1.6 JSON Charge Point, and using the object generated by the WSDL files to serialize JSON might send the correct spelling to a Central System.

We advise the implementers of OCPP 1.6 JSON Charge Points to correct the spelling mistake.



This issue will not be fixed in future schema files, Central Systems need to support both, and adding both in the Schema could be confusing for Charge Point implementers.

## 2.2. SendLocalList.json schema file incorrect

The OCPP 1.6 JSON schema file for SendLocalList is incorrect, as it is not a valid JSON Schema draft-04.

The part that defines the "idTagInfo" is incorrect: "expiryDate" and "parentIdTag" should have been part of the properties.

Incorrect

```
"idTagInfo": {
  "type": "object",
  "expiryDate": {
    "type": "string",
    "format": "date-time"
  },
  "parentIdTag": {
    "type": "string",
    "maxLength": 20
  },
  "properties": {
    "status": {
      "type": "string",
      "enum": [
        "Accepted",
        "Blocked",
        "Expired",
        "Invalid",
        "ConcurrentTx"
      ]
    }
  },
  "required": [
    "status"
  ]
}
```

Corrected

```

"idTagInfo": {
  "type": "object",
  "properties": {
    "expiryDate": {
      "type": "string",
      "format": "date-time"
    },
    "parentIdTag": {
      "type": "string",
      "maxLength": 20
    },
    "status": {
      "type": "string",
      "enum": [
        "Accepted",
        "Blocked",
        "Expired",
        "Invalid",
        "ConcurrentTx"
      ]
    }
  },
  "required": [
    "status"
  ]
}

```

A newer JSON Schema will be provided. When using JSON Schemas, this newer schema SHALL be used, the older version cannot be parsed by (most) JSON schema tools.

### 2.3. BootNotificationResponse.json schema file incorrect

The OCPP 1.6 JSON schema file for BootNotificationResponse contains an error. The field "interval" is defined as a 'number', but should have been: 'integer'

The WSDL files for OCPP-S is correct.

Incorrect

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "BootNotificationResponse",
  "type": "object",
  "properties": {
    "status": {
      "type": "string",
      "enum": [
        "Accepted",
        "Pending",
        "Rejected"
      ]
    },
    "currentTime": {
      "type": "string",
      "format": "date-time"
    },
    "interval": {
      "type": "number"
    }
  },
  "additionalProperties": false,
  "required": [
    "status",
    "currentTime",
    "interval"
  ]
}
```

Corrected

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "BootNotificationResponse",
  "type": "object",
  "properties": {
    "status": {
      "type": "string",
      "enum": [
        "Accepted",
        "Pending",
        "Rejected"
      ]
    },
    "currentTime": {
      "type": "string",
      "format": "date-time"
    },
    "interval": {
      "type": "integer"
    }
  },
  "additionalProperties": false,
  "required": [
    "status",
    "currentTime",
    "interval"
  ]
}

```

## 2.4. UpdateFirmware.json schema file incorrect

The OCPP 1.6 JSON schema file for UpdateFirmware contains 2 errors. The fields "retries" and "retryInterval" are defined as a 'number', but should have been: 'integer'

The WSDL files for OCPP-S is correct.

Incorrect

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "UpdateFirmwareRequest",
  "type": "object",
  "properties": {
    "location": {
      "type": "string",
      "format": "uri"
    },
    "retries": {
      "type": "number"
    },
    "retrieveDate": {
      "type": "string",
      "format": "date-time"
    },
    "retryInterval": {
      "type": "number"
    }
  },
  "additionalProperties": false,
  "required": [
    "location",
    "retrieveDate"
  ]
}
```

Corrected

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "UpdateFirmwareRequest",
  "type": "object",
  "properties": {
    "location": {
      "type": "string",
      "format": "uri"
    },
    "retries": {
      "type": "integer"
    },
    "retrieveDate": {
      "type": "string",
      "format": "date-time"
    },
    "retryInterval": {
      "type": "integer"
    }
  },
  "additionalProperties": false,
  "required": [
    "location",
    "retrieveDate"
  ]
}

```

## 2.5. StopTransaction.json "reason" incorrectly required

The OCPP 1.6 JSON schema message definitions set, as originally published, contain an error that renders it inconsistent with both the written OCPP 1.6 specification and the SOAP/XML Schema/WSDLs schema definitions. In the StopTransactionRequest message structure (StopTransaction.json), the "reason" element is incorrectly marked as being "required", whereas the human-readable protocol specification document and the SOAP WSDL service definition file both explicitly define it to be "optional", with an explicitly defined implied default value ("Local"), as stated in Sections 4.10 (p46) and 6.49 (p82) of the OCPP 1.6 Specification.

We advise implementers of an OCPP 1.6 JSON to remove the "reason" field from the list of required fields at the bottom of the file, or download the correct set of JSON schema from the OCA website.

To the best understanding of the TWG, the only case where any existing OCPP 1.6 implementation would be impacted by this correction is when a Central System implementation currently enforces the "required" status of "reason" in received JSON StopTransaction requests, and is controlling a network containing Charge Points that are using OCPP 1.6 JSON, but omitting the StopTransaction "reason", as is allowed by the specification.

## 3. Minor errata

Improvements to the descriptions on how the protocol (should) work.

### 3.1. JSON Schema files do allow for extra fields within inner objects.

It is not allowed to add extra fields/values to OCPP messages, this could cause interoperability issues in the field.

The WSDL files are correct, the original JSON Schema files allow extra fields on inner objects and extra values on enums, which was not intended.

Most of the JSON Schema files have been updated to fix this. The line: "additionalProperties": false has been added to the definition of all object and enum definitions.

### 3.2. No definition of: 'failure to process the message'

The OCPP 1.6 spec points to the OCPP-J spec for a definition on how a JSON implementation should respond to a request to indicate: 'failure to process message', but there was no definition of: 'failure to process the message' in the OCPP-J spec.

To following text should be added to: Page 9, par 3.2:

"For a definition on how a server SHALL report a 'failure to process the message', see: CallError"

To following text should be added to: Page 13, par 4.2.3:

"When a server needs to report a 'failure to process the message', the server SHALL use a Message Type: CallError (MessageTypeNumber = 4)."

### 3.3. Page 8, section: 3.2: Missing Charge Point should retry after having received HTTP 404.

When a Charge Point tries to connect the a Central System, the Central System, when it does not know the Charge Point might reject the connection by responding with HTTP 404 as defined in Section 3.2.

The Charge Point is then expected to retry the connection every X times, same as with the BootNotification, but this is not specified.

Addition text for the first bullet in Section 3.2

cols="2,10"]

Additional text
The Charge Point SHALL retry with an appropriate back-off mechanism to prevent overloading the Central System.

### 3.4. Page 10, section: 4.1.1: Synchronicity explanation can be improved

Not clear to all implementers that the Central System does not have to wait for a response from Charging Station 1 when sending something to Charging Station 2.

The following is to be added at the end of the first paragraph

Additional text	This does not mean that the Central System cannot send a message to another Charge Point, while waiting for a response of a first Charge Point, this rule is per OCPP-J connection.
-----------------	---

### 3.5. Page 10, par 4.1.3: 'Client-to-Server' and 'Server-to-Client' is confusing.

The table on page 9 contains a definition of the 3 different types of messages in the OCPP RPC framework. But the direction is confusing. With WebSockets, on the WebSocket level, the Central System is always the Server.

Old text	New text
Direction	Description
Client-to-Server	Request message
Server-to-Client (CALLRESULT)	Response message
Server-to-Client (CALLERROR)	Error response to a request message

### 3.6. Page 11, par 4.2: No reference to the JSON Schemas

The OCPP-J Spec has no reference to the JSON Schemas and schema version.

Additional reference in 1.6 references on page 3:

[JSON_SCHEMA]	<a href="http://json-schema.org/">http://json-schema.org/</a>
---------------	---

Additional text:

**4.2.4 Schemas** Together with this document, there are also JSON Schema files provided. These schemas can be used for validation of OCPP-J messages.

The OCPP-J 1.6 Schema files are [JSON\_SCHEMA] draft-04 schemas.

### 3.7. Page 14, section: 4.2.3: No example of CallError

There is no example of a CallError in the document, the following is an example.

```
[ 4,
  "162376037",
  "NotSupported",
  "SendLocalList.req not implemented",
  {}
]
```

### 3.8. Page 15, section: 5: Explanation how to respond to message before Charge Point is accepted.

There is no explanation for how a Central System should respond to any CALL message from the Charge Point (other than BootNotification.req) before being accepted by the Central System.

Additional section	<p>5.6 Charge Point CALL message before accepted</p> <p>When a Central System receives CALL messages (other then BootNotification) from a Charge Point before the Charge Point is accepted, the Central System is RECOMMENDED to respond with and RPC CallError: SecurityError.</p>
--------------------	---

### 3.9. Page 20, section: 7: How to handle half open connections not clear

It can sometimes happen that WebSockets connections are not opened or closed correctly. When this happens, a WebSocket ping should detect this and cause a WebSocket to be closed. By sending WebSockets pings at a regular interval, it should never happen that messages are dropped because of a half open connection.

Additional text	<p>Recommended to configure: WebSocketPingInterval smaller then: TransactionMessageAttempts * TransactionMessageRetryInterval.</p>
-----------------	--

### 3.10. Page 20, section: 7: Missing configurationKey: AuthorizationKey

On page 17, a ConfigKey is named, but it is not listed in the list on page 20:

New text:

KEY	VALUE
AuthorizationKey	<p>String</p> <p>WriteOnly</p> <p>Hexadecimal representation of the password that the Charging Station uses to authenticate itself if HTTP Basic authentication is used. Maximum string length is 40 characters.</p> <p>This configuration key is write-only, so that it cannot be accidentally stored in plaintext by the Central System when it reads out all configuration keys.</p>

## 4. Typos

Typos, fixes to incorrect links/reference, improve terms used etc. that have no impact on the description of the way the protocol works.

None known

## 5. Known issues that will not be fixed

### 5.1. Page 14, par 4.2.3: CallError: Typo in enum

Typo in enum: OccurenceConstraintViolation

Old text	OccurenceConstraintViolation
----------	------------------------------

New text	OccurrenceConstraintViolation
----------	-------------------------------

Do not fix, this is a message level change, might break implementations

Note: In the next version of OCPP we will add the correct spelling and make the incorrect (typo) value deprecated.

## 5.2. Page 14, par 4.2.3. CallError: incorrect name in enum: FormationViolation

Incorrect name in enum: FormationViolation

Old text	FormationViolation
New text	FormatViolation

Do not fix, this is a message level change, might break implementations

Note: In the next version of OCPP we will add the correct spelling and make the incorrect (typo) value deprecated.