

## eMabler Push API Messages

This document describes examples of eMabler Push API messages that can be received thru different means of transport. Transport layers to push these messages can be agreed with the customer. General ways have been to push messages into a customer hosted queue (Azure ServiceBus, RabbitMQ, AWS SNS etc.). Messages can be delivered also with REST by calling Customer hosted endpoint each time message arrives to eMabler. Also other custom solutions can be arranged.

### ChargerState message

This message is used to deliver the current status of a charging station and it's socket  
Status values are: Available, Error, Offline, Info, Charging, SuspendedCAR, SuspendedCHARGER,Preparing,Finishing,Booting, Unavailable

```
{
  "chargerId": "ChargerId1234",
  "socketId": 1,
  "timeStamp": "2022-03-07T07:35:25.523161Z",
  "status": "Available",
  "errorCode": "No Error",
  "message": "String with charger specific info"
}
```

### ChargerBoot message

This message is used to deliver more detailed info about charger when it boots. This is basically replacing the simple Booting State message

```
{
  "chargerId": "ChargerId1234",
  "timeStamp": "2022-03-18T14:25:40.23181Z",
  "serialNumber": "129019209",
  "chargerModel": "Model eMx",
  "manufacturer": "eMabler",
  "firmwareVersion": "1.12",
  "iccid": "328983298392",
  "imsi": "810038108401",
  "meterSerialNumber": "30190310",
  "meterType": "MeterType1"
}
```

### FullChargingTransaction message

This message is used to deliver information of a full charging transaction. This message is delivered after the charging has ended.

```
{
  "chargerId": "ChargerId1234",
  "socketId": 1,
  "timeStampStart": "2022-03-07T07:35:25.475521Z",
  "timeStampEnd": "2022-03-07T09:35:25.475575+02:00",
  "transactionId": 123445,
  "authorizedIdTag": "JFDKSJF",
  "meterReadStart": 0,
  "meterReadEnd": 12300,
  "consumptionWh": 12300
}
```

### ChargingTransaction message

This message is send when charging transaction starts and stops.  
Action can be: transaction\_start or transaction\_stop

```
{
  "chargerId": "ChargerId1234",
  "socketId": 1,
  "timeStamp": "2022-03-07T07:53:47.392618Z",
  "action": "transaction_start",
  "transactionId": 123445,
  "authorizedIdTag": "JFDKSJF",
  "meterRead": 0
}
```

### Measurements message

This message is send when charger is reporting meter values during the charging transaction.  
TypeofMeasurement, Phase and Unit are following OCPP1.6 model.

```

{
  "chargerId": "ChargerId1234",
  "socketId": 1,
  "timeStamp": "2022-03-07T07:35:25.535215Z",
  "measurements": [
    {
      "value": "10",
      "typeOfMeasurement": "Current.Import",
      "phase": "L1-N",
      "unit": "A"
    },
    {
      "value": "11",
      "typeOfMeasurement": "Current.Import",
      "phase": "L2-N",
      "unit": "A"
    },
    {
      "value": "10",
      "typeOfMeasurement": "Current.Import",
      "phase": "L3-N",
      "unit": "A"
    }
  ]
}

```

#### APIChargerChange message

If charger is created updated or deleted, thru our REST API call there can be a notification thru the push integration with this message. Action values can be: NEW\_CHARGER, DELETE\_CHARGER, CHARGER\_GROUP\_CHANGED, CHARGER\_UPDATED

```

{
  "chargerId": "ChargerId1234",
  "action": "NEW_CHARGER",
  "timeStamp": "2022-03-07T07:35:25.54818Z",
  "apiChargerChange": {
    "newChargerName": "",
    "newLocationInfo": null,
    "newSockets": null,
    "newGroupId": 123,
    "newEvseId": "",
    "newCapabilities": null
  }
}

```