PURPOSE

The Australian Energy Market Operator (AEMO) has prepared this Guide to VPP Demonstration APIs to provide information about VPP APIs. It provides participants with the API technical specifications to assist with the development of their own systems using VPP APIs.

PRIVACY AND LEGAL NOTICES

The material in this publication may be used in accordance with the privacy and legal notices on AEMO’s website.

TRADEMARK NOTICES

Copyright (c) 2002 JSON.org
Oracle and/or its affiliates: Copyright © 1995, 2018. All rights reserved. postmanlabs/postman is licensed under the Apache License 2.0 YAML: Copyright © 2001-2009 Oren Ben-Kiki, Clark Evans, Ingy döt Net

DISTRIBUTION

Available to the public.

DOCUMENT IDENTIFICATION

Application owner: IT Platforms
Document owner: Technology
Prepared by: Technology, Technical Writers
Last update: Tuesday, 30 July, 2019

DOCUMENTS MADE OBSOLETE

The release of this document changes only the version of Guide to AEMO’s e-Hub APIs.

FURTHER INFORMATION

For further information, please visit AEMO’s website www.aemo.com.au or contact:
AEMO Information and Support Hub Phone: 1300 AEMO 00 (1300 236 600) and follow the prompts.
Email: supporthub@aemo.com.au

FEEDBACK

Your feedback is important and helps us improve our services and products. To suggest improvements, please contact AEMO’s Information and Support Hub.

VERSION RELEASE HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Effective Date</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>31 July 2019</td>
<td>Initial draft.</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1. Purpose and scope
This document provides details of the API specification developed for VPP Demonstration project. The guide will help participants developing their own applications that will interface with AEMO system.

This is the initial draft API Specification document focused on VPP Enrolments. The final version of the API specification document will include VPP Operational Data APIs and FCAS Response APIs and will incorporate any enhancements to the Enrolment API specification.

1.2. Audience
The primary audience is the VPP participant’s technical staff responsible for building application using AEMO APIs for the VPP demonstration project.

The secondary audience is anyone who has an interest in understanding how AEMO’s APIs work.

1.3. How to use this guide
- This document is written in plain language for easy reading. Where there is a discrepancy between the National Electricity Rules (Rules) and information or a term in this document, the Rules take precedence.
- The references listed throughout this document are primary resources and take precedence over this document.
- Text in this format indicates a resource on AEMO’s website.
- Glossary terms have the meanings listed against them in the Glossary section.
- This guide assumes you have knowledge of the RESTful programming architecture.

1.4. What’s in this guide
AEMO has produced this document to provide participants with the guideline for data interface using the APIs developed by AEMO. This detail is to assist participants with design their own systems to use AEMO APIs. The guide explains the communication protocol and methods with examples.

1.5. Related documents

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Related documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Location</td>
</tr>
</tbody>
</table>

2. OVERVIEW AND INTERFACE SCOPE
Virtual Power Plant (VPP) is an emerging concept being trialled across Australia, largely driven by subsidy schemes incentivising the uptake of thousands of residential battery units.
AEMO has launched the VPP Demonstrations to test a new specification for distributed energy resources (DER) to deliver frequency control ancillary services (FCAS), increasing competition for FCAS and allowing VPPs to explore the commercial feasibility of stacking multiple value streams.

AEMO is also developing its systems to receive operational data from VPPs that will provide visibility of the distribution connected DER to AEMO. This will help AEMO learn how to integrate VPPs into the market at scale, which will then inform appropriate regulatory and operational changes.

The VPP Demonstrations’ functional scope includes building a system to support the VPP trials. The system will enable data transfer over internet via Open APIs between AEMO and VPP participants to facilitate and support the following functions:

- VPP Enrolment (i.e. enrolling in VPP demonstrations and ongoing data updates).
- Contingency FCAS assessment.
- Operational forecasting and on-boarding VPP operational data.

### Table 2  Functional scope and releases

<table>
<thead>
<tr>
<th>Functional areas</th>
<th>First release</th>
<th>Final release</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPP Enrolment</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FCAS Assessment</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Operational Forecasting</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

This document provides API specifications for the VPP Enrolment only. These Open APIs will only be accessible by internet. These Open APIs will not be accessible via MarketNet.

A new version of this specification will be published before the final release of the VPP Demonstrations APIs.

### 2.1. VPP Demonstration enrolment process

#### 2.1.1. VPP enrolment steps

### Figure 1  VPP enrolment process steps

### Table 3  Enrolment steps and interface type

<table>
<thead>
<tr>
<th>VPP enrolment step</th>
<th>Description</th>
<th>Method</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-enrolment</td>
<td>Interested participants to submit VPP Trial participation enrolment form.</td>
<td>Email/post</td>
<td>VPP ID and DUID Confirmed</td>
</tr>
<tr>
<td>VPP enrolment step</td>
<td>Description</td>
<td>Method</td>
<td>Outcome</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>During enrolment</td>
<td>A: VPP to submit data for the Load configuration assessment:</td>
<td>API</td>
<td>Determination of Load Configuration and Capacity to Deliver Contingency FCAS. Registration of VPP in trial</td>
</tr>
<tr>
<td></td>
<td>1. NMI and Device data (NMI &amp; Devices associated to DUID &amp; to VPP),</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Frequency Injection Test Data for distinct Battery types and site (i.e. DUID)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B: Additional data requested by AEMO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-enrolment</td>
<td>On-going data submission to update VPP fleet configuration (NMI and Device – data update)</td>
<td>API</td>
<td>Manage VPP’s fleet capacity and configuration</td>
</tr>
</tbody>
</table>

### 2.2. Conceptual architecture

#### Figure 2  Architecture context diagram

![Architecture context diagram](image)

**2.2.1. Interface message pattern**

The VPP Demonstration interface will follow both PUSH and PULL message patterns from the participants’ perspective. The diagram below describes the data exchange pattern for the VPP Demonstrations.
2.2.2. Interfaces

Table 4 Interface List

<table>
<thead>
<tr>
<th>Interface ID and name</th>
<th>Data</th>
<th>Message pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface 1-Submit NMI &amp; Device</td>
<td>NMI &amp; Device Enrolment Data</td>
<td>PUSH</td>
<td>Participant submits list of NMIs and associated device(s) with relevant attribute values for enrolment into VPP demonstration trial and load configuration assessment.</td>
</tr>
<tr>
<td>Interface 2- Retrieve NMI &amp; Device</td>
<td>NMI &amp; Device Enrolment Data</td>
<td>PULL</td>
<td>Participant retrieve (i.e. requests) the list of enrolled NMIs and associated devices for a specific DUID in an VPP.</td>
</tr>
<tr>
<td>Interface 3- Remove NMI &amp; Device</td>
<td>NMI Enrolment Data</td>
<td>PUSH</td>
<td>Participant submits a NMI or list of NMIs that needs to be disassociated (i.e. removed) from the DUID and the VPP. This will also disassociate all the devices attached to the NMI from the DUID in the VPP.</td>
</tr>
<tr>
<td>Interface 4- Remove Device</td>
<td>Device Enrolment Data</td>
<td>PUSH</td>
<td>Participant submits a device or list of devices that needs to be disassociated from a NMI. In addition to removing association to NMI, this will result in device(s) being removed from the DUID and VPP configuration as well.</td>
</tr>
<tr>
<td>Interface 5-Submit Frequency Injection data</td>
<td>Frequency Injection Test Data</td>
<td>PUSH</td>
<td>Participant submits Frequency Injection test data for the device and site (i.e. DUID) for Load Configuration Assessment for Contingency FCAS.</td>
</tr>
</tbody>
</table>

For all the interfaces, AEMO chose RESTful (REST) APIs because of its lightweight nature and ability to transmit data using HTTPS and JSON. The following section provides the detail protocol and API specification for VPP registrations.
3. **API STANDARD**

The VPP demonstration project uses AEMO e-Hub platform standards and leverages the current security framework. The AEMO e-Hub platform uses OpenAPI Specification (OAS).

For detail about AEMO’s e-HUB platform, architecture, and common standards, please refer to the AEMO e-Hub API guide (see Section 1.5 for link).

AEMO and the VPP Demonstrations project team are currently assessing the alignment with IEEE 2030.5. However, this API specification is developed based on current AEMO e-Hub specification.

3.1. **Design principles**

VPP Demonstrations API design uses the following principles:

- APIs are RESTful and use open standards.
- APIs are secure.
- APIs provide a good user experience for participants and developer.

3.2. **URL format**

VPP Demonstrations API URL design follows AEMO’s e-Hub APIs standards. The VPP Demonstrations APIs can only be accessed through internet. These APIs will not be accessible through MarketNet. VPP Demonstrations API endpoints will have the following format:

```plaintext
https://<host server>/<business function>/<API version>/<resource>?<Query string parameters>
```

Example URL for VPP Enrolment:

```plaintext
```

3.3. **HTTP request header**

<table>
<thead>
<tr>
<th>Header parameter</th>
<th>Description</th>
<th>Allowed value / example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>HTTP request format</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Accept</td>
<td>HTTP response format</td>
<td>Accept: application/json</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Content length of file. The value is populated when the request is sent.</td>
<td>Content-Length: nnn</td>
</tr>
<tr>
<td>X-initiatingParticipantID</td>
<td>The Participant ID</td>
<td>X-initiatingParticipantID:123456</td>
</tr>
<tr>
<td>X-market</td>
<td>The market type that the request applies.</td>
<td>NEMWholesale</td>
</tr>
<tr>
<td>Authorization</td>
<td>Specifies basic HTTP authentication containing the Base64[1] encoded username and password. The participant’s URM username and password are concatenated with a colon separator and then Base64 encoded.</td>
<td>Authorization: Basic YXNh2FzOmFzYXNh2Fz (for URM username “VPP01” and password “nsicu@$@#8asdsad”)</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>HTTP payload compression</td>
<td>gzip, deflate</td>
</tr>
</tbody>
</table>
3.4. HTTP methods

Table 6  Methods

<table>
<thead>
<tr>
<th>HTTP methods</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>For retrieval of information (single or collation)</td>
</tr>
<tr>
<td>POST</td>
<td>• To create a resource item</td>
</tr>
<tr>
<td></td>
<td>• To spawn an action</td>
</tr>
<tr>
<td></td>
<td>• Changes the state of the resource</td>
</tr>
<tr>
<td></td>
<td>• To disassociate NMI and/or Device from a DUID &amp; VPP configuration</td>
</tr>
</tbody>
</table>

3.5. HTTP Response

The HTTP Response will have a response code and description, with

- A successful request indicated by 200 OK.
- Other response codes for technical and Payload validation failures.
- Optional Payload.
**Figure 4  Example response**

HTTP/1.1, 200 OK  
Content-Type: application/json  
Content-Length: nnn

```json
{
  "transactionId": "string",
  "data": [
    {
      "vppId": "VSVAT13",
      "duid": "ASNCTEW1",
      "nmi": 9901002033,
      "DNSP": "ACTEWNGY",
      "NEMRegion": "ACT",
      "postalCode": 2601,
      "devices": [
        {
          "deviceSerialId": "TG9989893XX",
          "deviceModelVersion": "RESU10H",
          "deviceControlBoxModelVersion": "Reposit v2.1",
          "deviceManufacturer": "LG Chem",
          "deviceType": "Storage",
          "acEquipmentType": "Inverter",
          "inverterManufacturer": "Solar Edge",
          "inverterModelVersion": "SE5000",
          "inverterSerialNumber": "SE416735",
          "maxR6": 4,
          "maxR60": 9,
          "maxR5": 9,
          "maxL6": 9,
          "maxL60": 9,
          "maxL5": 9,
          "deviceSize": 12.6,
          "uom": "kW"
        }
      ],
      "createdDate": "2019-05-02 11:14:30:05",
      "createdBy": "string",
      "updatedDate": "2019-05-02 11:14:30:05",
      "updatedId": "string"
    }
  ]
}
```

**Table 7  HTTP response code**

<table>
<thead>
<tr>
<th>Response code</th>
<th>Description</th>
<th>Data condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
<td>Request Processed Successfully</td>
</tr>
<tr>
<td>400</td>
<td>The service cannot be found for the endpoint reference</td>
<td>Invalid API URL</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
<td>Malformed payload (JSON)</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>Invalid Credentials</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>Expired User Password</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>No BASIC Auth information in HTTP Request Header</td>
</tr>
</tbody>
</table>
### 3.6. Data Validation

The validations for the incoming API requests are categorised as:

1. Technical validations
   - Connectivity (for example, SSL authentication).
   - Throttling limits
2. Payload validations
   - Payload (for example, validation of the JSON payload)
   - HTTP request / response header parameters (for example, missing / invalid HTTP request / response header parameters)
   - Data Attribute Validation

### 3.7. Security

#### 3.7.1. SSL Certificate

All communications between the e-HUB and a participant’s gateway are carried out using HTTPS. HTTP is not supported on the e-HUB. mTLS/SSL encryption is managed using public/private key pairs, with a different key pair required to connect to each environment (pre-production/production). Each participant must create / obtain a private key and a Certificate Signing Request (CSR).

A private key and CSR is usually created at the same time, making a key pair. A CSR is usually generated on the server where the certificate will be installed and contains information that will be included in the certificate such as the organisation name, common name (CN), locality and country. It also contains the public key that will be included in the certificate.

#### 3.7.2. Authentication and Authorisation

When calling the VPP API(s), the Participants must authenticate their identity using Basic Authentication (by passing username & password). Participant Administrator (PA) will provide the username and password to the API development team. The HTTP Basic authentication header takes the following format:

```
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

```
3.8. **API Portal**

AEMO provides the API portal for detail documentation and access to API swagger files. The API portal can be accessed using the following URLs:

- **Pre-production:** https://preprod.apiportal.aemo.com.au
- **Production:** https://apiportal.aemo.com.au

Please refer to Guide-to-AEMOs-eHub-APIs (link Section-1.5) for details on the AEMO API Portal.

4. **VPP ENROLMENT APIS**

The VPP NMI and the devices need to be enrolled through APIs. The enrolment information needs to include Participant ID and NMI for each device. The following resources are implemented by AEMO for the VPP Demonstration APIs.

4.1. **API summary**

**Table 8**  
**VPP API common header attributes**

<table>
<thead>
<tr>
<th>API Name: VPP</th>
<th>Accept</th>
<th>Production Base URL (internet)</th>
<th>Pre-production Base URL (internet)</th>
<th>Content-Type</th>
<th>Content-Length</th>
<th>Accept_Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface ID</td>
<td>Transaction Description</td>
<td>Resource</td>
<td>Method</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Submit NMI &amp; Device data</td>
<td>submitNMIDevice</td>
<td>POST</td>
<td>Submit NMIs and attached devices under VPP ownership or control associated to the DUID, that are part for the VPP fleet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Retrieve NMI &amp; Device data</td>
<td>retrieveNMI Device</td>
<td>GET</td>
<td>Retrieve (to get a) list of valid NMIs and devices data associated to a specific DUID under a VPP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Remove NMIs from VPP fleet</td>
<td>removeNMI Device</td>
<td>POST</td>
<td>Submit the list of NMIs that needs to be disassociated from DUID and removed from the VPP fleet. Removing/disassociating NMI will also remove the devices attached to the NMI from the DUID and the VPP.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Remove device from VPP fleet</td>
<td>removeDevice</td>
<td>POST</td>
<td>Submit a list of devices that needs to be disassociated from a NMI. Thus removed from the DUID and the VPP fleet.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Submit Frequency Injection Test Data for Load Configuration</td>
<td>submitFreqInjTestData</td>
<td>POST</td>
<td>Submit the Frequency Injection Test Data for Contingency FCAS Load Configuration Assessment for the devices (distinct batteries in VPP) and site (i.e. DUID).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.2. API – submit NMI and Device Data

<table>
<thead>
<tr>
<th>Resource name</th>
<th>/submitNMIDevice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request parameter</td>
<td>None</td>
</tr>
<tr>
<td>Request payload</td>
<td>`{ data*: deviceSchema {   vppId*: string   duid*: string   nmis*: [{     nmi*: string     devices*: [device { deviceSerialId*: string deviceModelVersion*: string deviceControlBoxModelVersion*: string deviceManufacturer*: string deviceType*: string acEquipmentType*: string inverterManufacturer*: string inverterModelVersion*: string inverterSerialNumber*: string maxR6*: integer maxR60*: integer maxR5*: integer maxL6*: integer maxL60*: integer maxL5*: integer deviceSize*: integer uom*: string } ] } ] } }</td>
</tr>
</tbody>
</table>


Example

```json
{
  "data": {
    "vppId": "VSVAT13",
    "duid": "ASNACTEW1",
    "nmis": [
      {"nmi": 9901002033,
       "devices": [
         {"deviceSerialId": "TG9989893XX",
          "deviceModelVersion": "RESU10H",
          "deviceControlBoxModelVersion": "Reposit v2.1",
          "deviceManufacturer": "LG Chem",
          "deviceType": "Storage",
          "acEquipmentType": "Inverter",
          "inverterManufacturer": "Solar Edge",
          "inverterModelVersion": "SE5000",
          "inverterSerialNumber": "SE416735",
          "maxR6": 4,
          "maxR60": 9,
          "maxR5": 9,
          "maxL6": 9,
          "maxL60": 9,
          "maxL5": 9,
          "deviceSize": 12.6,
          "uom": "kW"
        }
      ]
    ]
  }
}
```

Response

200 - Submit Data for VPP Profile - Success
422 - Business validation failure
429 - This response is provided when the throttling limits are reached
500 - Internal Server Error

Example

```json
{"transactionID": "b85a35f8-f741-40ac-a701-a8cfebb25669",
 "data": {},
 "errors": [
  {
   "code": "429",
   "title": "Too Many Requests",
   "detail": "Number of inbound requests exceeded the throttling limits; try after sometime",
   "source": null
  }
 ]
}
```
## 4.3. API – retrieve NMI and device data

<table>
<thead>
<tr>
<th>Resource name</th>
<th>/retrieveNMIDevice</th>
</tr>
</thead>
</table>
| Query parameter   | "vppId": "string",
|                   | "duid": "string" |
| Request parameter | None               |
| Request payload   | None               |
| Response payload  | example            |

```json
{
  "transactionId": "string",
  "data": [
    {
      "vppId": "V5VAT13",
      "duid": "ASNACTEW1",
      "nmi": 9901002033,
      "dnsp": "ACTEWNGY",
      "nemRegion": "ACT",
      "postalCode": 2601,
      "devices": [
        {
          "deviceSerialId": "TG9989893XX",
          "deviceModelVersion": "RESU10H",
          "deviceControlBoxModelVersion": "Reposit v2.1",
          "deviceManufacturer": "LG Chem",
          "deviceType": "Storage",
          "acEquipmentType": "Inverter",
          "inverterManufacturer": "Solar Edge",
          "inverterModelVersion": "SE5000",
          "inverterSerialNumber": "SE416735",
          "maxR6": 4,
          "maxR60": 9,
          "maxR5": 9,
          "maxL6": 9,
          "maxL60": 9,
          "maxL5": 9,
          "deviceSize": 12.6,
          "uom": "kW"
        }
      ],
      "createdDate": "2019-05-02 11:14:30.05",
      "createdBy": "string",
      "updatedDate": "2019-05-02 11:14:30.05",
      "updatedId": "string"
    }
  ]
}
```

**Response**
- 200 - Success
- 429 - This response is provided when the throttling limits are reached
- 500 - Internal Server Error
### 4.4. API – remove NMI and device data

<table>
<thead>
<tr>
<th>Resource name</th>
<th>/removeNMI Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query parameter</td>
<td>None</td>
</tr>
<tr>
<td>Request parameter</td>
<td></td>
</tr>
</tbody>
</table>
```
{   data*: nmiSchema{
    nmis [
      {   nmi string
        }
      ]
    }
  }
``` |
| Request payload | ("data": {
  "nmis": [
    {
      "nmi": "9901002033"
    }
  ]
}) |
| Response payload example | 200 - Submit Data for VPP Profile - Success  
422 - Business validation failure  
429 - This response is provided when the throttling limits are reached  
500 - Internal Server Error |
| Response | ("transactionID": "b85a35f8-f741-40ac-a701-a8clebb25669",  
"data": {},  
"errors": [  
  {   "code": "429",  
      "title": "Too Many Requests",  
      "detail": "Number of inbound requests exceeded the throttling limits; try after sometime",  
      "source": null  
  }  
]  
) |
4.5. API – remove device data

<table>
<thead>
<tr>
<th>Resource name</th>
<th>/removeDevice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query parameter</td>
<td>None</td>
</tr>
<tr>
<td>Request parameter</td>
<td></td>
</tr>
</tbody>
</table>

```json
{
  data: [ deviceRemoveSchema {
    vppId*: string
    duid*: string
    nmi*: string
    devices: [ {
      deviceSerialId*: string
      deviceModelVersion*: string
      deviceControlBoxModelVersion*: string
      deviceManufacturer*: string
      deviceType*: string
      acEquipmentType*: string
      inverterManufacturer string
      inverterModelVersion string
      inverterSerialNumber string
    }
  }
}
```

Request payload

```json
{
  "data": [
    {
      "vppId": "VSVAT13",
      "duid": "ASNACTEW1",
      "nmi": 9901002033,
      "devices": [
        {
          "deviceSerialId": "TG9989893XX",
          "deviceModelVersion": "RESU10H",
          "deviceControlBoxModelVersion": "Reposit v2.1",
          "deviceManufacturer": "LG Chem",
          "deviceType": "Storage",
          "acEquipmentType": "Inverter"
        }
      ]
    }
  ]
}
```

Response payload example

200 - Submit Data for VPP Profile - Success
422 - Business validation failure
429 - This response is provided when the throttling limits are reached
500 - Internal Server Error

Response

```json
{"transactionID": "b85a35f8-f741-40ac-a701-a8cfebb25669",
 "data": {},
 "errors": [
  {
   "code": "429",
   "title": "Too Many Requests",
   "detail": "Number of inbound requests exceeded the throttling limits; try after sometime",
   "source": null
  }
 ]
}
```
### 4.6. API – submit frequency injection test data

<table>
<thead>
<tr>
<th>Resource name</th>
<th>/submitFreqInjTestData</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query parameter</td>
<td>None</td>
</tr>
<tr>
<td>Request parameter</td>
<td></td>
</tr>
</tbody>
</table>

```json
{
  "data": {
    "vppId": "VSVAT13",
    "duid": "ASNACTEW1",
    "fcasMarketType": "R6",
    "samplingRate": "20ms",
    "uom": "kW",
    "freqTestData": [{
      "samplingInterval": -4.6,
      "frequency": 50.00032169,
      "power": -12.24,
      "gridPower": 12.24,
      "measurementDatetime": "2019-05-02 11:14:30:05",
      "deviceName": "LG"
    }]
  }
}
```

<table>
<thead>
<tr>
<th>Request payload</th>
<th></th>
</tr>
</thead>
</table>

```json
{
  "data": {
    "vppId": "VSVAT13",
    "duid": "ASNACTEW1",
    "fcasMarketType": "R6",
    "samplingRate": "20ms",
    "uom": "kW",
    "freqTestData": [
      {
        "samplingInterval": -4.6,
        "frequency": 50.00032169,
        "power": -12.24,
        "gridPower": 12.24,
        "measurementDatetime": "2019-05-02 11:14:30:05",
        "deviceName": "LG"
      }
    ]
  }
}
```

<table>
<thead>
<tr>
<th>Response payload example</th>
<th></th>
</tr>
</thead>
</table>

**200 - Submit Data for VPP Profile - Success**

**422 - Business validation failure**

**429 - This response is provided when the throttling limits are reached**

**500 - Internal Server Error**

<table>
<thead>
<tr>
<th>Response</th>
<th></th>
</tr>
</thead>
</table>

```json
{"data": {"vppId": "VSVAT13", "duid": "ASNACTEW1", "fcasMarketType": "R6", "samplingRate": "20ms", "uom": "kW", "freqTestData": [{"samplingInterval": -4.6, "frequency": 50.00032169, "power": -12.24, "gridPower": 12.24, "measurementDatetime": "2019-05-02 11:14:30:05", "deviceName": "LG"}]}
```

```json
{"transactionID": "b85a35f8-f741-40ac-a701-a8cfebb25669", "data": {}}, "errors": [{"code": "429", "title": "Too Many Requests", "detail": "Number of inbound requests exceeded the throttling limits; try after sometime", "source": null}]}
```
## APPENDIX A. GLOSSARY

<table>
<thead>
<tr>
<th>Abbreviation/term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Gateway</td>
<td>The gateway on AEMO’s side providing participant communication options, accessible only over the internet. It uses resources and methods to push messages to Participants’ API Gateways.</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface; a set of clearly defined methods of communication between various software components.</td>
</tr>
<tr>
<td>API Web Portal</td>
<td>Where you can view available APIs, view and manage your API Keys, obtain OAS files.</td>
</tr>
<tr>
<td>CSR</td>
<td>Certificate Signing Request is a block of encoded text given to a Certificate Authority when applying for an SSL Certificate. It also contains the Public Key to include in the certificate. Usually, a Private Key is created at the same time, making a Key Pair.</td>
</tr>
<tr>
<td>e-Hub</td>
<td>Consists of the API Web Portal and the API Gateway for both electricity and gas.</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>Key Pair</td>
<td>SSL uses a technique called public-key cryptography, based on the concept of a Key Pair. The Key Pair consists of encrypted Public and Private Key data. It is only possible to decrypt the Public Key with the corresponding Private Key.</td>
</tr>
<tr>
<td>MarketNet</td>
<td>AEMO’s private data network connection. VPP APIs are not available via MarketNet.</td>
</tr>
<tr>
<td>OAS</td>
<td>OpenAPI Specification</td>
</tr>
<tr>
<td>Participant API Gateway</td>
<td>The interface implemented by participants where AEMO’s API pushes messages.</td>
</tr>
<tr>
<td>Participant ID</td>
<td>Registered participant identifier</td>
</tr>
<tr>
<td>Payload</td>
<td>The data sent by a POST request. The Payload sections sits after the header.</td>
</tr>
<tr>
<td>PID</td>
<td>Participant ID</td>
</tr>
<tr>
<td>Private Key</td>
<td>The secret Private Key is a text file used initially to generate a Certificate Signing Request (CSR), and later to secure and verify connections.</td>
</tr>
<tr>
<td>Public Key</td>
<td>The Public Key is included as part of your SSL certificate, and works together with your Private Key to make sure your data is encrypted, verified, and not tampered with. Anyone with access to the Public Key (i.e. the certificate) can verify the digital signature is authentic without having to know the secret Private Key.</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer, cryptographic protocol providing API communication security.</td>
</tr>
<tr>
<td>Swagger file</td>
<td>The OpenAPI Specification (OAS) definition of the API.</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security, cryptographic protocol providing API communication security.</td>
</tr>
<tr>
<td>VPP</td>
<td>Virtual Power Plant</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>