Project EDGE
Partner Introduction

Disclaimer: This document has been provided to AEMO and partner organisations and is current as at the date of publication. Further updates and amendments to the document may take place to align with most current solution design and development specifications.

Version: Initial Draft

Important notice

NO RELIANCE OR WARRANTY

This document does not constitute legal or business advice, and should not be relied on as a substitute for obtaining detailed advice about the National Gas or Electricity Law, the Rules or any other applicable laws, procedures or policies. While AEMO has made every effort to ensure the quality of the information in this Guide, neither AEMO, nor any of its employees, agents and consultants make any representation or warranty as to the accuracy, reliability, completeness, currency or suitability for particular purposes of that information.

LIMITATION OF LIABILITY

To the maximum extent permitted by law, AEMO and its advisers, consultants and other contributors to this Guide (or their respective associated companies, businesses, partners, directors, officers or employees) are not liable (whether by reason of negligence or otherwise) for any errors, omissions, defects or misrepresentations in this document, or for any loss or damage suffered by persons who use or rely on the information in it.

TRADEMARK NOTICES

Microsoft, Windows and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the US and other countries.

© 2015 Google Inc, used with permission. Google and the Google logo are registered trademarks of Google Inc.

DISTRIBUTION

Available to the public.

DOCUMENT IDENTIFICATION

Prepared by: Energy Web

Last update: Friday, 17 December 2021 12:21 PM

DOCUMENTS MADE OBSOLETE

No documents are made obsolete by publication of this document.

FEEDBACK

Your feedback is important and helps us improve our services and products. To suggest improvements, please contact AEMO's Support Hub. To contact AEMO's Support Hub use Contact Us on AEMO's website or Phone: 1300 AEMO 00 (1300 236 600) and follow the prompts.

Contents

VERSION RELEASE HISTORY

Version	Effective date	Summary of changes
Initial Draft		

1. Purpose and Audience

This document describes the process for integrating with Energy Web's technology solution to facilitate participation in the Project EDGE DER Marketplace operation and to deliver Wholesale Services (to AEMO) and Local Services (to DNSPs, Distribution Network Service Providers).

Energy Web Foundation, on behalf of the Australian Energy Market Operator (AEMO), provides this information which is addressed to key business technical and IT staff in participating organisations.

2. Energy Web

Energy Web Foundation (EW) is a nonprofit organisation whose mission is to accelerate a low-carbon, customer-centric electricity system by unleashing the potential of open-source, decentralised technologies. In 2019 EW launched the Energy Web Chain, the world's first open-source, enterprise blockchain platform tailored to the energy sector. EW's technology roadmap has since grown into the Energy Web Decentralized Operating System (EW-DOS), a full stack that includes front-end applications and a variety of software development toolkits.

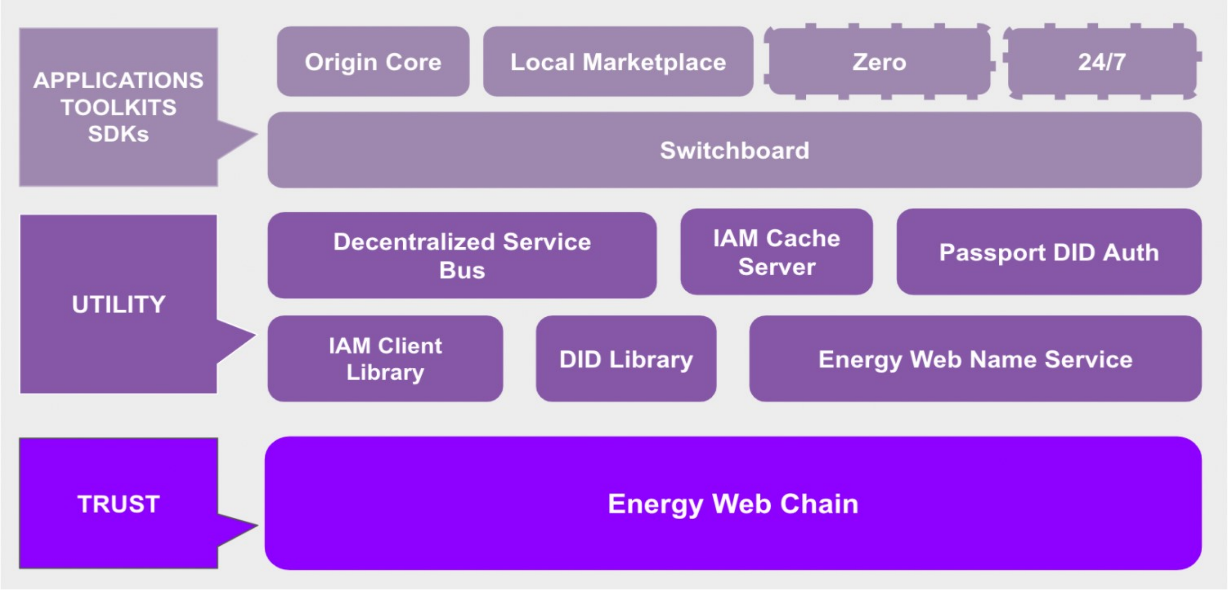
For more about EW, visit here.

3. EW Technology Overview

The Energy Web Decentralised Operating System (EW-DOS) is comprised of three layers:

1. the **Trust** layer,
2. the **Utility** layer, and
3. the **Toolkit** layer.

Figure 1 Energy Web's Decentralised Operating System (EW-DOS)



For more information about EW-DOS, please visit here.

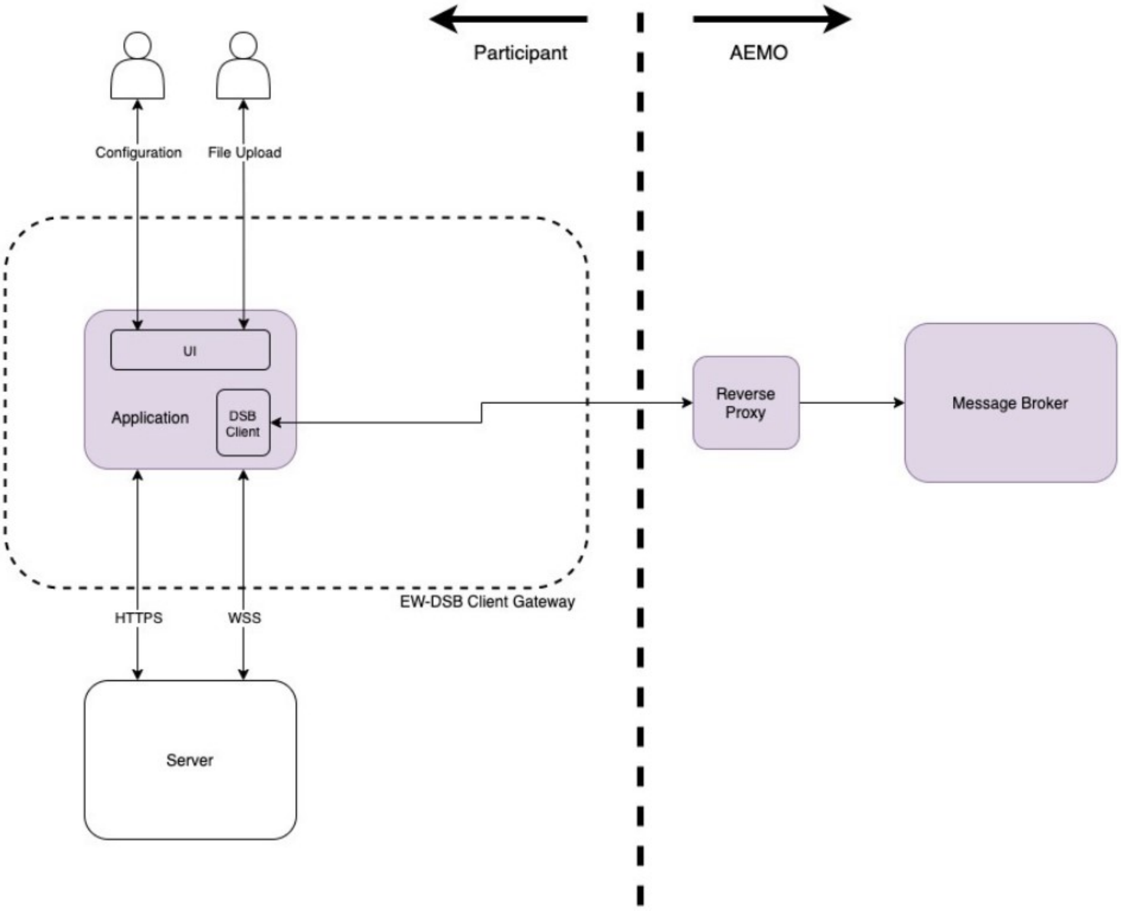
4. EW-DSB Gateway Container

4.1 EW-DSB installation and configuration

This section steps through how to setup the interface gateway container and configure it to access the Project EDGE applications. There are three key steps

1. Install and run the Gateway container
2. Generate a Decentralised Identifier (DID) and request enrolment approval
3. Interact with RESTful and/or web-socket interfaces to send and receive messages to/from the Project EDGE applications.

Figure 2 Participant integration with AEMO via EW-DSB Client Gate



4.2 Container Installation

The following instructions show how a Client Gateway container can be set up to participate in the EDGE project. In this case, we will run the container directly by pulling it from the EWF Azure Container Registry.

The following command will allow you to pull the latest version of the gateway container:

```
docker pull aemocontainerregistry.azurecr.io/dsb/client-gateway:latest
```

Alternatively, an exact image tag can be installed, for example:

```
v0.4.0
```

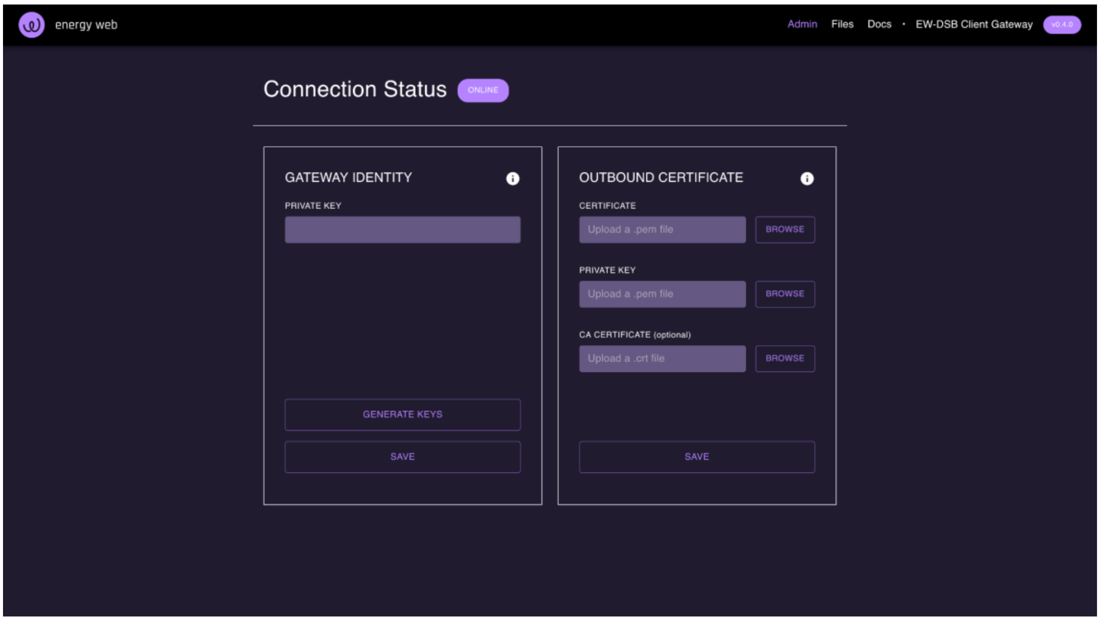
An overview of current stable releases (matching image tags) can be found at DSB Client Gateway Releases.

4.3 Running/Configuring the Container

Once the desired image has been pulled, you can use `docker run` to run the container. For the latest instructions on how to configure the container (including environment variables), see the DSB Client Gateway Setup Guide. The DSB Client Gateway Setup guide contains further detail that enhances and extends the general guidance provided below.

Once setup and running, navigating to http://localhost:3000/ should display the Gateway's Configuration UI. Note that if authentication is enabled you will be asked by your browser to enter the credentials provided in the environment variables.

Figure 3 Gateway Configuration UI

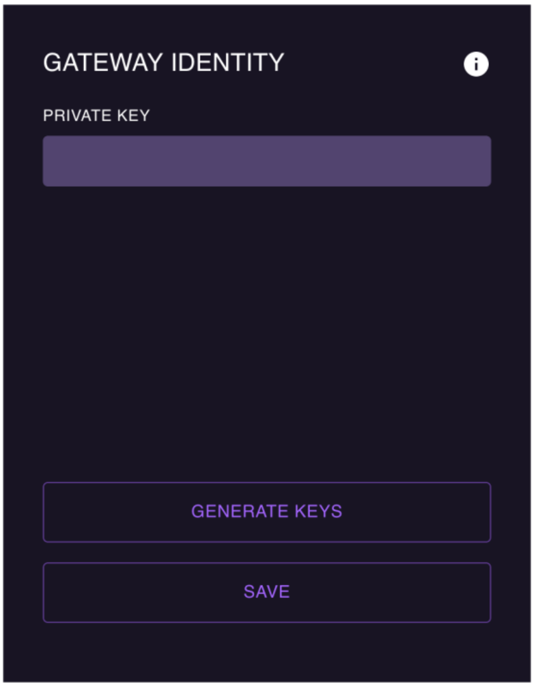


4.4 Enrolling

Enrolment of your gateway is necessary to participate in the EDGE project. It is possible to do this through the DSB Client Gateway UI. To achieve this, a private key is needed.

To generate and save a new private key, click the "generate keys" button as seen on the gateway identity component.

Figure 4 Gateway UI



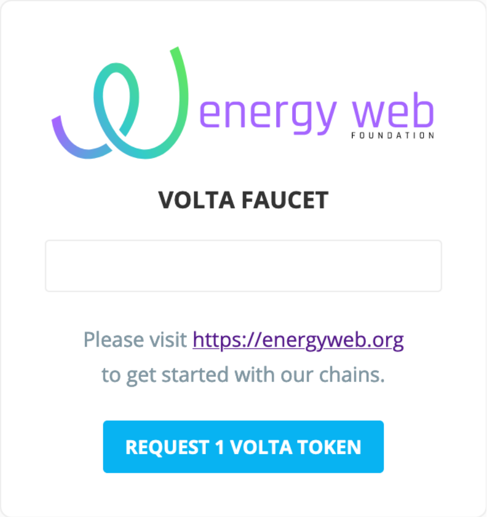
GATEWAY IDENTITY

PRIVATE KEY

You should see a confirmation message which tells you that the private key has been saved. You will need to visit EWF Volta testnet faucet to freely request a token on Volta (the test network of the Energy Web Chain) this facilitates verification and enrolment message transactions to provision role based access to the Project EDGE applications.

You may still generate or save a different key, though note that only one key can be used at a time with a client gateway.

Figure 5 Volta Faucet

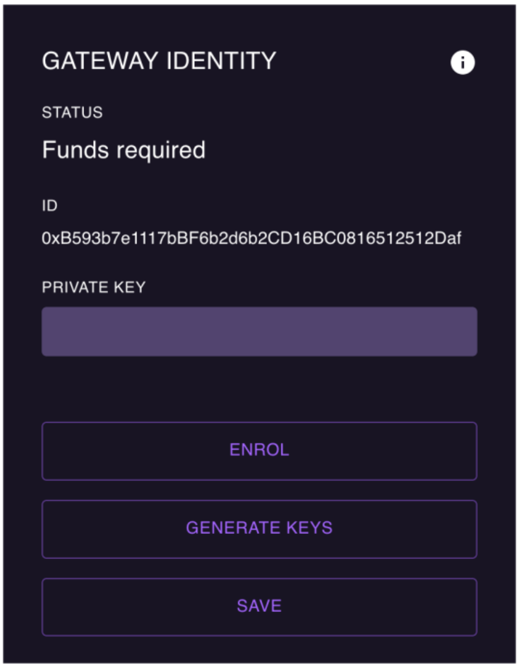


VOLTA FAUCET

Please visit https://energyweb.org to get started with our chains.

The “ID” in the image below is the corresponding address of the private key. It is important to note the difference between the address and key itself. This can be copied and pasted into the above input to request the Volta token.

Figure 6 Gateway UI with Address (ID)



GATEWAY IDENTITY

STATUS

Funds required

ID

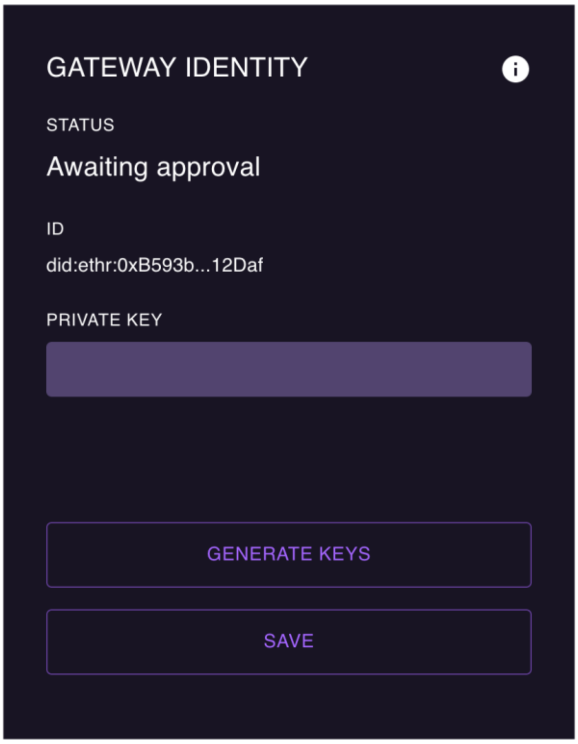
0xB593b7e1117bBF6b2d6b2CD16BC0816512512Daf

PRIVATE KEY

Now that a private key has been stored by the gateway with a Volta token associated, it can be used to enrol on to the Energy Web Decentralized Service Bus. This can be done by clicking the “enrol” button. It will take a short while (the browser tab should not be closed during this time).

Once done, your identity component should look similar to the image below.

Figure 7 Gateway Identity


GATEWAY IDENTITY

STATUS

Awaiting approval

ID

did:ethr:0xB593b...12Daf

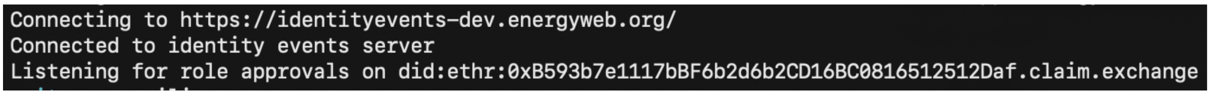
PRIVATE KEY

The address that was previously shown has now been replaced with a Decentralized Identifier (DID) (which contains the address of the private key after "did:ether:").

The status has changed to "awaiting approval", telling us that the enrolment request must first be approved by AEMO in Energy Web Switchboard.

If you look at the container logs, you will see that a listener has been started in order to receive notification of an approved enrolment request.

Figure 8 Container log


```
Connecting to https://identityevents-dev.energyweb.org/
Connected to identity events server
Listening for role approvals on did:ethr:0xB593b7e1117bBF6b2d6b2CD16BC0816512512Daf.claim.exchange
```

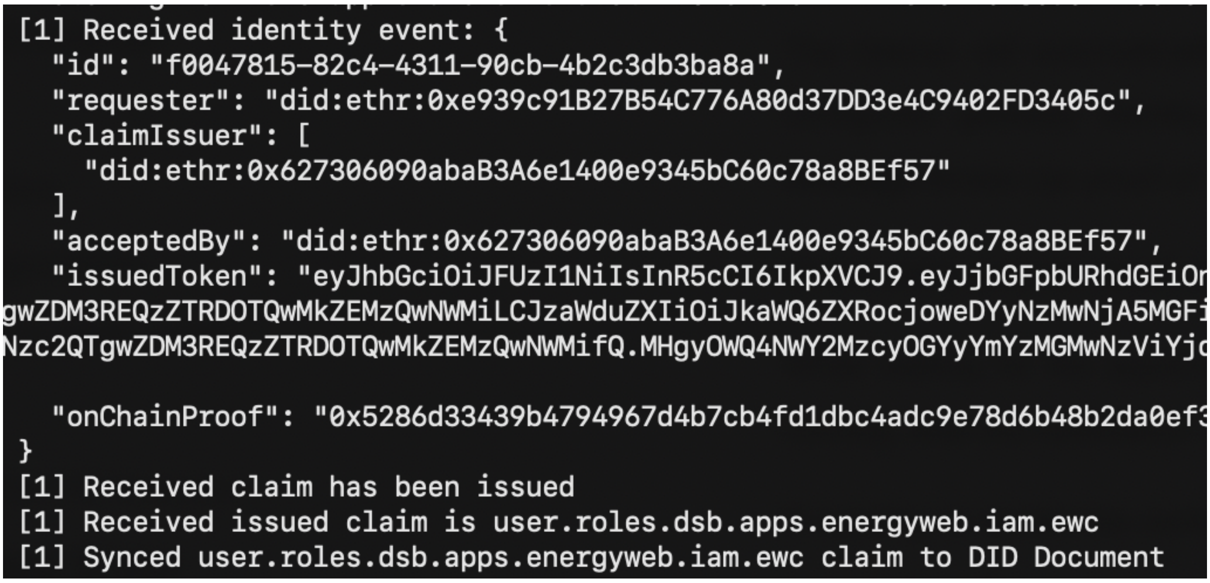
The listener will automatically sync the approved enrolment claims to the DID Document of the configured gateway identity. This allows the gateway to authenticate itself on the DSB Message Broker (as proof of the approved role will be resolved from the document). More information about the role of the DID and DID document can be found at:

https://energy-web-foundation.gitbook.io/energy-web/foundational-concepts/self-sovereign-identity#did-documents

While waiting for the approval, it is important to keep the listener (and therefore the container) running, else the notification will currently be missed.

The container log will also show approval notifications and tells you that the claim has been successfully synced to the DID document.

Figure 9	Container logs with approvals



```
[1] Received identity event: {
  "id": "f0047815-82c4-4311-90cb-4b2c3db3ba8a",
  "requester": "did:ethr:0xe939c91B27B54C776A80d37DD3e4C9402FD3405c",
  "claimIssuer": [
    "did:ethr:0x627306090abaB3A6e1400e9345bC60c78a8BEf57"
  ],
  "acceptedBy": "did:ethr:0x627306090abaB3A6e1400e9345bC60c78a8BEf57",
  "issuedToken": "eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJjbGFpbURhdGEiO
gwZDM3REQzZTRDOTQwMkZEMzQwNWMiLCJzaWduZXIiOiJkaWQ6ZXRocjoweDYyNzMwNjA5MGFi
Nzc2QTgwZDM3REQzZTRDOTQwMkZEMzQwNWMifQ.MHgyOWQ4NWY2MzcyOGYyYmYzMGMwNzViYj

  "onChainProof": "0x5286d33439b4794967d4b7cb4fd1dbc4adc9e78d6b48b2da0ef
}
[1] Received claim has been issued
[1] Received issued claim is user.roles.dsb.apps.energyweb.iam.ewc
[1] Synced user.roles.dsb.apps.energyweb.iam.ewc claim to DID Document
```

Now that your user role has been approved and synced, you are able to send and retrieve messages on the DSB.

4.5 Using the Documentation

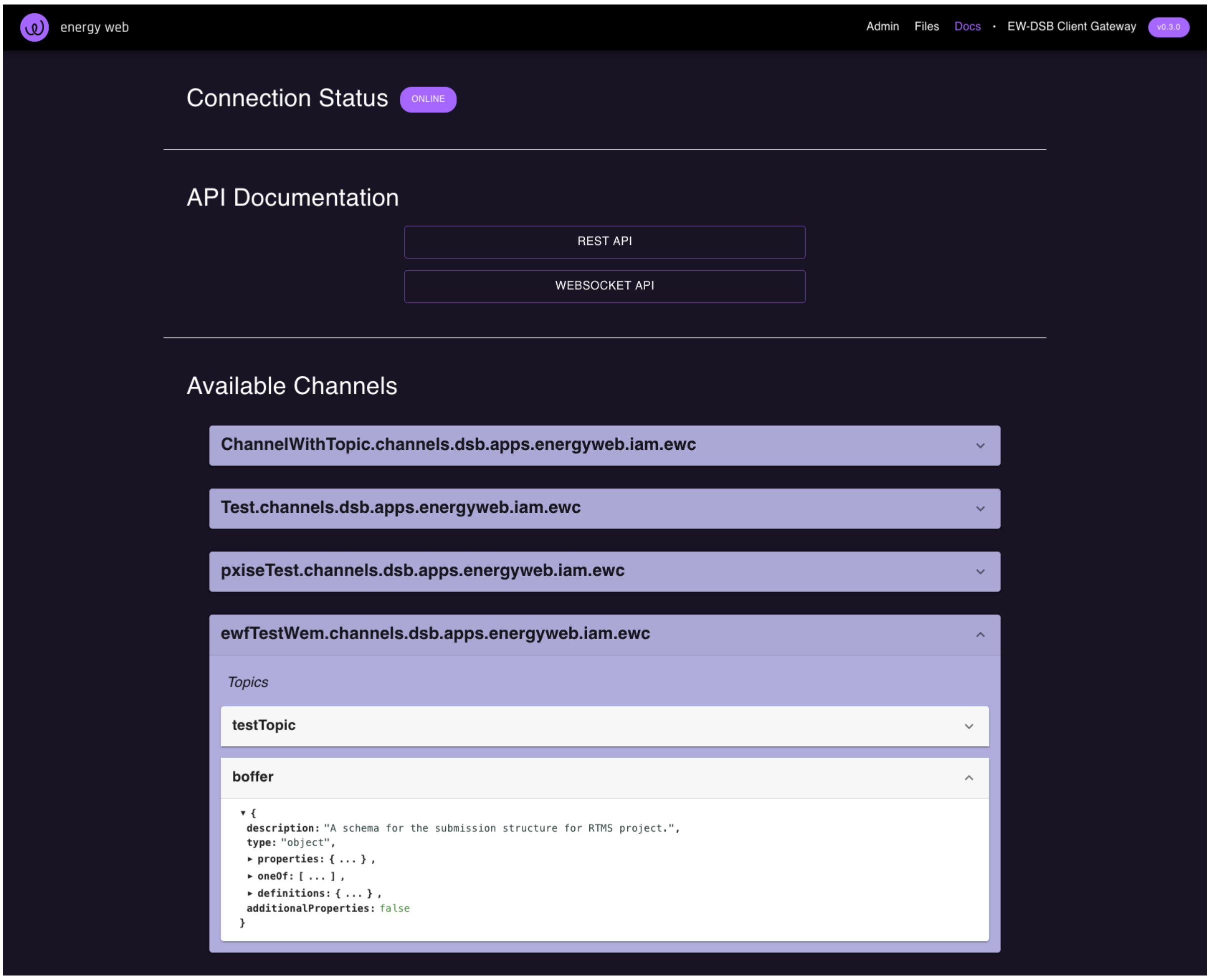
DSB works based on two fundamental building blocks: channels and messages. Channels are structures that allow publishers and subscribers to exchange messages with at-least-once delivery semantics. DSB uses built-in persistence for every channel.

Visit http://localhost:3000/docs to find the documentation for the DSB Client Gateway API. You can also click on the “Docs” navigation bar link on the gateway homepage.

4.6 Channels

In order to send or receive messages on the DSB, you must be assigned to a channel by AEMO. You can find out which channels you have publish/subscribe rights to on the documentation page.

Figure 10 Documentation page showing Channels



Connection Status ONLINE

API Documentation

Available Channels

ChannelWithTopic.channels.dsb.apps.energyweb.iam.ewc

Test.channels.dsb.apps.energyweb.iam.ewc

pxiseTest.channels.dsb.apps.energyweb.iam.ewc

ewfTestWem.channels.dsb.apps.energyweb.iam.ewc

Topics

testTopic

boffer

```
▾ {
  description: "A schema for the submission structure for RTMS project.",
  type: "object",
  ▸ properties: { ... } ,
  ▸ oneOf: [ ... ] ,
  ▸ definitions: { ... } ,
  additionalProperties: false
}
```

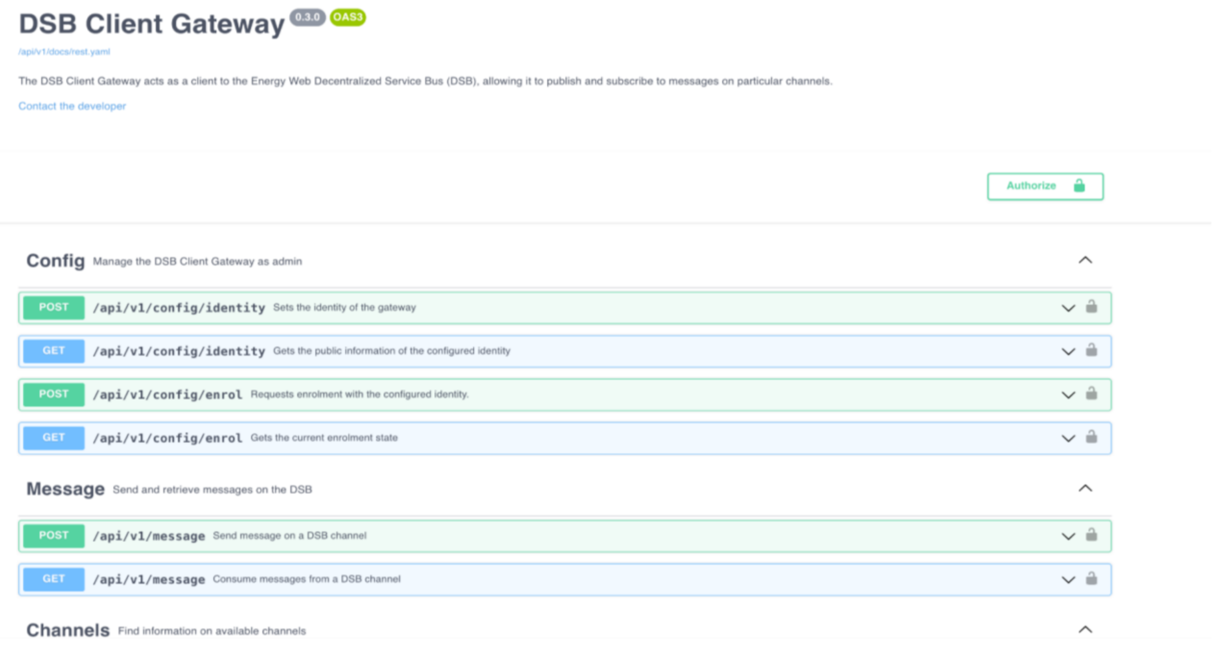
Once you have been added to a channel, it will show up on this page, where you can view the JSON schema of message topics within the channel. The channels, topics and schema properties can be expanded by clicking on them.

<u>Open API (REST) / Async API (WebSockets)</u>

Links on the main documentation page provide access to the REST and WebSocket API documentation.

The RESTful API docs are available at http://localhost:3000/docs/rest:

Figure 11 Client Gateway API



DSB Client Gateway 0.3.0 OAS3

/api/v1/docs/rest.yaml

The DSB Client Gateway acts as a client to the Energy Web Decentralized Service Bus (DSB), allowing it to publish and subscribe to messages on particular channels.

Contact the developer

Config Manage the DSB Client Gateway as admin

- POST `/api/v1/config/identity` Sets the identity of the gateway
- GET `/api/v1/config/identity` Gets the public information of the configured identity
- POST `/api/v1/config/enrol` Requests enrolment with the configured identity.
- GET `/api/v1/config/enrol` Gets the current enrolment state

Message Send and retrieve messages on the DSB

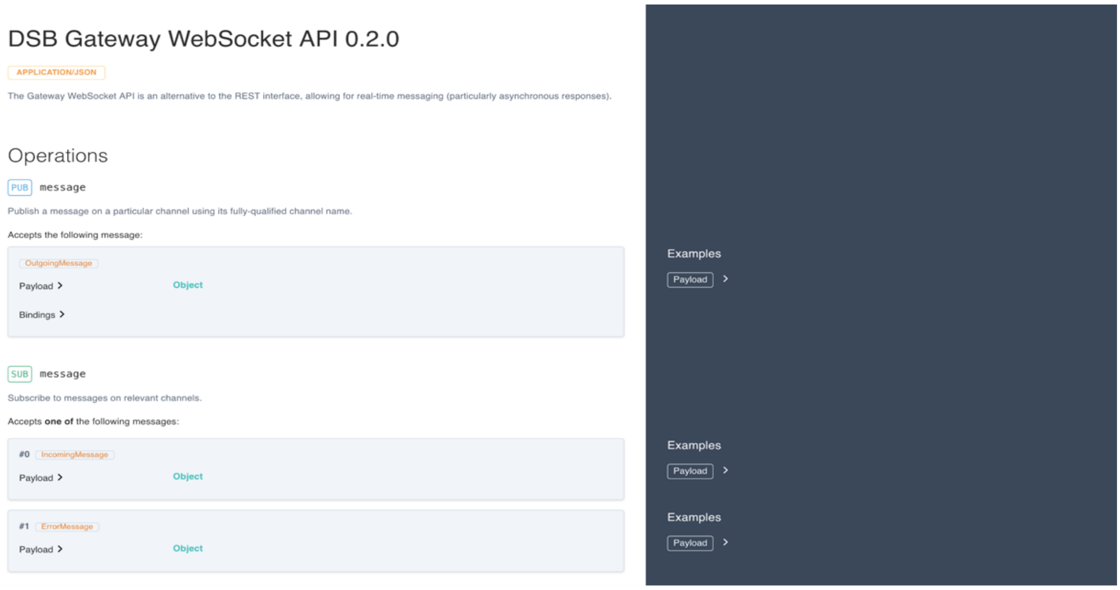
- POST `/api/v1/message` Send message on a DSB channel
- GET `/api/v1/message` Consume messages from a DSB channel

Channels Find information on available channels

Once you have been assigned channels, you can use the Message API to begin testing your implementation. More information on this is provided in the following section.

Messages sent/received over WebSockets are described at http://localhost:3000/docs/ws.

Figure 12 Gateway Websocket API



DSB Gateway WebSocket API 0.2.0

APPLICATION/JSON

The Gateway WebSocket API is an alternative to the REST interface, allowing for real-time messaging (particularly asynchronous responses).

Operations

PUB `message`

Publish a message on a particular channel using its fully-qualified channel name.

Accepts the following message:

OutgoingMessage

Payload › Object

Bindings ›

SUB `message`

Subscribe to messages on relevant channels.

Accepts **one of** the following messages:

#0 IncomingMessage

Payload › Object

#1 ErrorMessage

Payload › Object

The WebSocket API allows you to subscribe to messages in order to receive them in real-time and avoid polling the RESTful Message API.

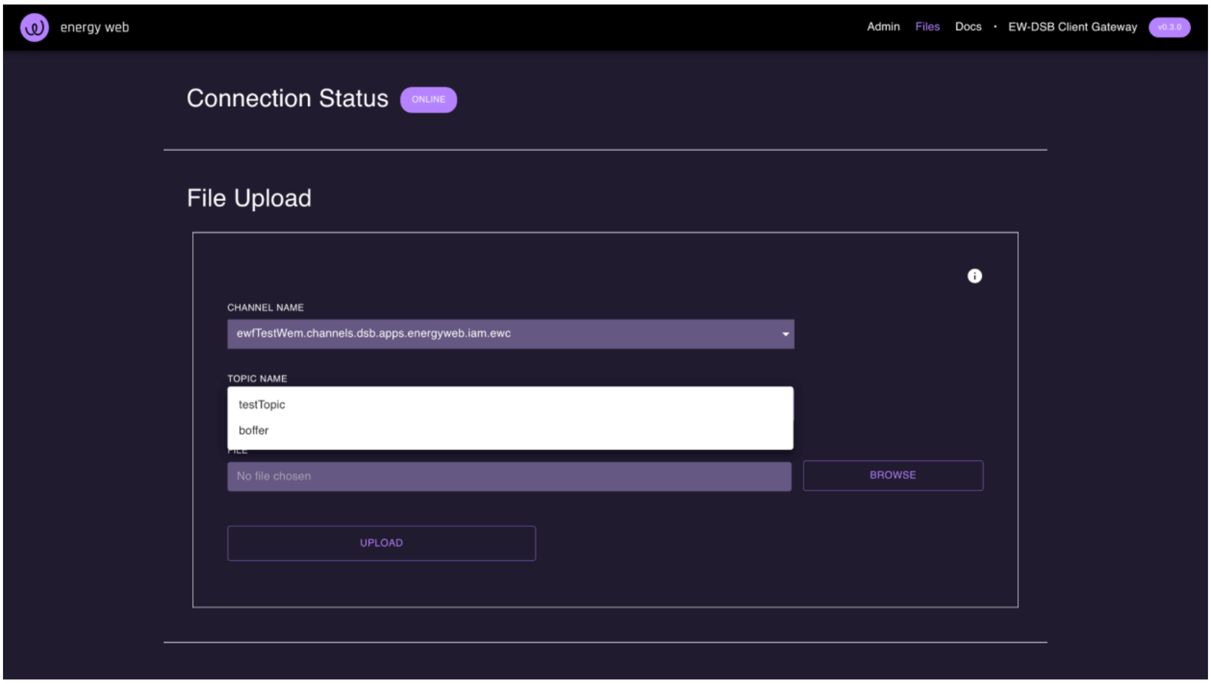
The WebSocket API displayed pertains to the gateway’s server implementation. It is also possible to run the gateway as a WebSocket client (and therefore your service will have a server implementation).

In such a case you will need to subscribe to messages of type "OutgoingMessage" and publish messages of type "IncomingMessage".

4.7 File Upload

As an alternative to API (direct) interaction, the Gateway Client UI allows a file to be uploaded to interact with the services provided (eg. Boffer, Operating Envelope, etc). The channel name can be selected from a dropdown menu containing the list of available channels. Once selected, the topic name can be selected from a dropdown menu containing the list of available topics within that channel.

Figure 13 File Upload Page



Connection Status ONLINE

File Upload

CHANNEL NAME
ewfTestWem.channels.dsb.apps.energyweb.iam.ewc

TOPIC NAME
testTopic
boffer

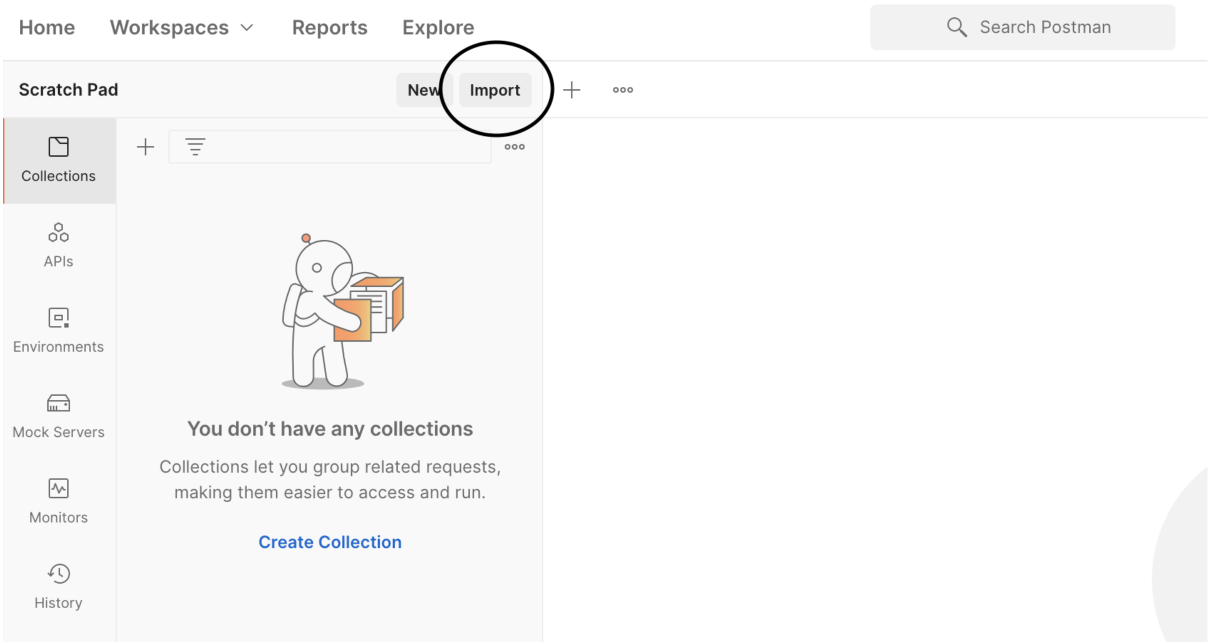
FILE
No file chosen

4.8 Sending and Receiving Messages

Most transactions are initiated with a POST to a service (asynchronous posts are also possible).

Offer submission through the API can be made through a client (e.g. Postman). As a starting point for integration, you can import the gateway's OpenAPI specification into Postman:

Figure 14 Importing the OpenAPI specification



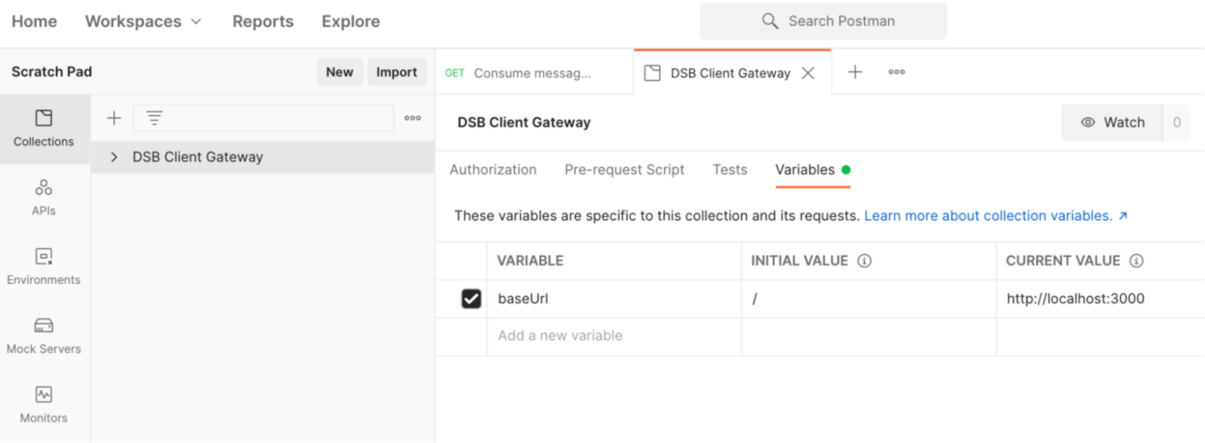
You don't have any collections

Collections let you group related requests, making them easier to access and run.

Once you click "import", you will be prompted to drag-and-drop or upload a file. The file can be obtained from the gateway API (http://localhost:3000/api/v1/docs/rest.yaml) or found on GitHub: https://github.com/energywebfoundation/dsb-client-gateway/blob/master/public/rest.yaml

Once imported, you should see the "DSB Client Gateway" collection. You will need to click on it and navigate to the "Variables" tab, where you can set the "baseUrl" to "http://localhost:3000" or equivalent.

Figure 15 Configuring the baseUrl

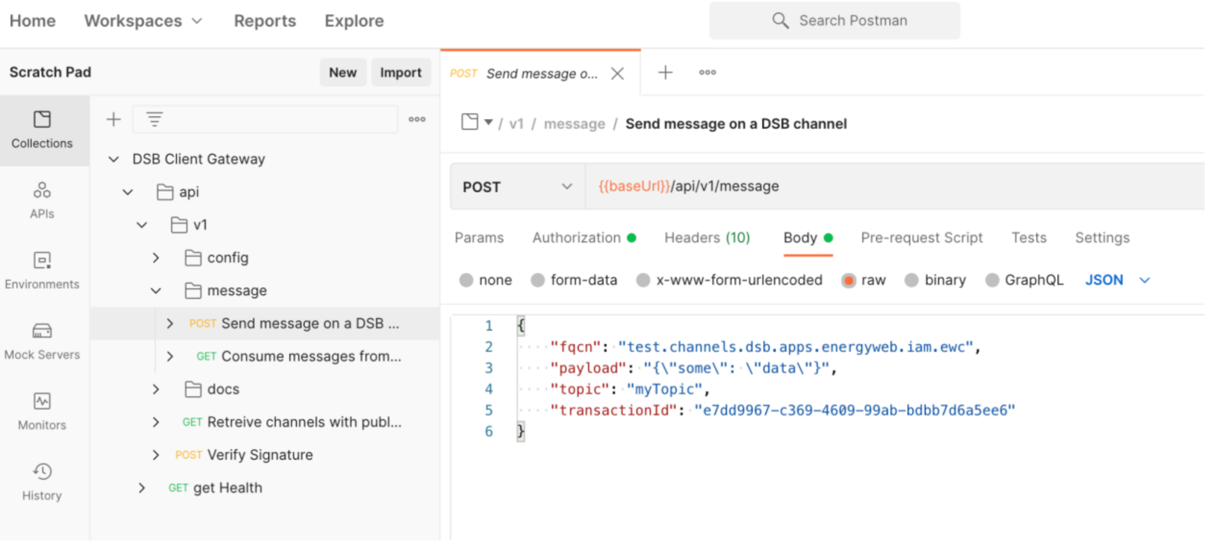


These variables are specific to this collection and its requests. Learn more about collection variables. ↗

	VARIABLE	INITIAL VALUE	CURRENT VALUE
☑	baseUrl	/	http://localhost:3000
	Add a new variable		

With that now set, you can navigate through the path to /api/v1/message where there are two request types (POST for sending a message, GET for retrieving messages).

Figure 16 Sending a message



/ v1 / message / Send message on a DSB channel

POST {{baseUrl}}/api/v1/message

Body: ● raw JSON

```
{
····"fqcn": "test.channels.dsb.apps.energyweb.iam.ewc",
····"payload": "{\"some\": \"data\"}",
····"topic": "myTopic",
····"transactionId": "e7dd9967-c369-4609-99ab-bdbb7d6a5ee6"
}
```

You will need to fill in the fqcn and topic for the relevant message type (see the channels documentation highlighted by Figure 10). A transaction ID can optionally be included for identification and idempotency.

A request such as a boffer, or operating envelope, that is sent to AEMO will have an initial response that confirms the message is available for processing on the transport layer. If a business acknowledgement (i.e. from the recipient) is required you would need to either GET the acknowledgement on a specific channel/topic (e.g. by polling), or use web-sockets for real-time acknowledgement (with the transactionId helping to identify the asynchronous response).

A1. Further Resources

Further material on the Energy Web solution and architecture can be found at

A1.1 General

https://energy-web-foundation.gitbook.io/energy-web/

A1.2 Self-Sovereign Identify

https://energy-web-foundation.gitbook.io/energy-web/foundational-concepts/self-sovereign-identity

A1.3 EW-DSB

https://energy-web-foundation.gitbook.io/energy-web/technology/the-stack/utility-layer-1#decentralized-service-bus-dsb

A1.4 Decentralised Identifiers (DIDs)

https://energy-web-foundation.gitbook.io/energy-web/foundational-concepts/self-sovereign-identity#decentralized-identifiers-dids