

Contents

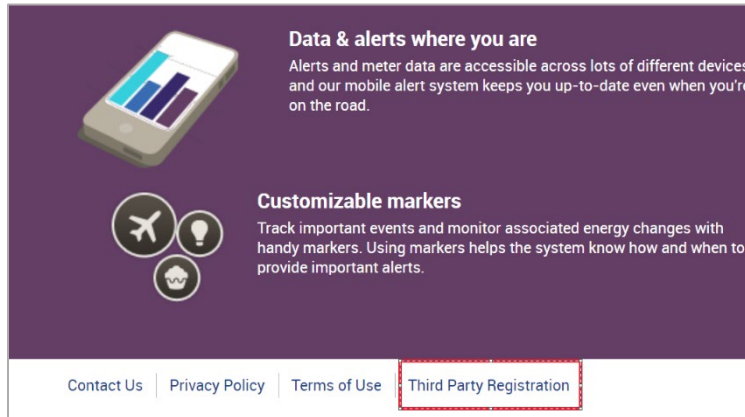
Section 1: Third-Party Registration	3
Scenario 1: Third-Party perspective	3
Scenario 2: HECO Admin Perspective	4
Section 2: Customer Authorization (OAuth)	6
1. Retail customer is redirected from 3P website to MyMeter portal.	6
a. Example authorize call:	6
b. Authorize call parameters:.....	6
o Function blocks supported by MyMeter*.....	6
2. Retail customer reaches the MyMeter Landing page and enters their CDC username & password:	7
3. Call is made to CDC's accounts.login REST API to authenticate CDC user.	8
4. Retail customer is presented with an authorization confirmation screen.	8
5. Retail customer is redirected to the third-party application.....	8
a. If they selected No:	8
b. If they selected Yes:	9
6. Third party converts authorization code to Access Token.	9
7. AI will authenticate the third party and their access token request.	10
8. If the third party passes this verification, AI will respond with the following information:.....	10
Section 3: Data Exchange	12
1. The third-party requests data from with their customer-specific access token	12
2. AI's resource server validates the customer-specific access token with the authorization server.	13
3. AI's authorization server makes a call to CDC's API to verify that said customer is still active.....	13
4. CDC responds with customer's status (active/inactive)	13
5. AI's authorization server responds to the resource server that the customer-specific access token is/isn't valid	13
6. AI sends the requested data to the third party.	13
Section 4: Complete List of Endpoints and Their Uses	14
- Types of authorizations:.....	14
a. Registration_access_token	14
b. Client_access_token	14
c. Access_token	15

2.	Authorization Endpoint:.....	15
3.	Token Endpoint:.....	15
4.	Bulk Request URI:.....	15
5.	Resource Endpoint:.....	16
a.	GET Application Information.....	16
b.	GET Authorization.....	16
c.	GET Authorizations	16
d.	GET Usage Points	16
e.	GET Usage Point.....	16
f.	GET Electric Power Quality Summary	16
g.	GET Meter Readings.....	16
h.	GET Meter Reading	17
i.	GET Usage Summaries	17
j.	GET Usage Summary	17
k.	GET Local Time Parameters	17
l.	GET Local Time Parameter	17
m.	GET All Meter Readings.....	17
n.	GET Interval Block	18
o.	GET Reading Type	18
p.	GET Service Status.....	18

Section 1: Third-Party Registration

Scenario 1: Third-Party perspective

1. Navigate to the HECO MyMeter landing page (for dev, <https://demo.mymeter.co/hecolab/>)
2. Locate the Third-Party Registration link in the footer of the page:



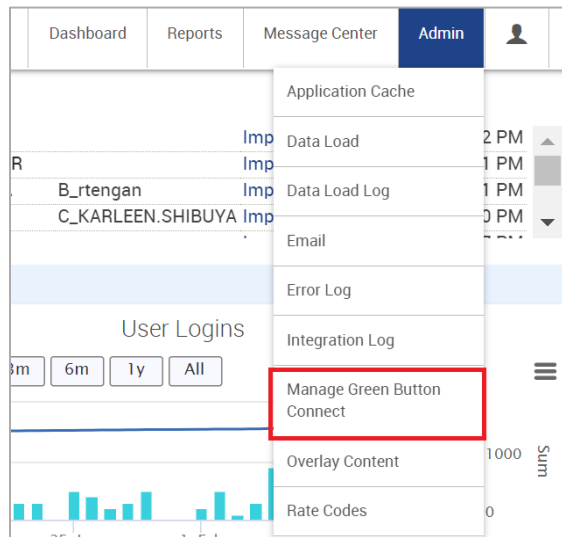
3. Fill out the Green Button Connect registration form based on your third-party information:

The image shows a web form titled 'Green Button Connect Registration'. The form is organized into two columns of input fields. The left column includes: 'Type of User' (a dropdown menu with 'Standard User' selected), 'Client Name', 'Company Address', 'Agent Name', 'Redirect URI', 'Token Endpoint Authentication Method', 'Scope', 'Grant Types', 'Response Types', 'Third-Party Notify URI', 'Software ID', and 'Policy URI'. The right column includes: 'Software Version', 'Third-Party Name', 'Contact' (with the example email 'support@example.com, retail@example.com'), 'Third-Party Application Description', 'Third-Party Application Status' (a dropdown menu with 'Development' selected), 'Third-Party Application Type' (a dropdown menu with 'Web' selected), 'Third-Party Application Use' (a dropdown menu with 'Energy Management' selected), 'Third-Party Phone' (with a placeholder 'X-XXX-XXX-XXXX'), 'Third-Party User Portal Screen URI', 'Logo URI', and 'Client URI'. At the bottom right of the form, there is a checkbox labeled 'I agree to Hawaiian Electric's Privacy Policy and Terms of Use' and a blue 'Submit' button.

4. Review & agree to Hawaiian Electric’s privacy policy and terms of use. Submit registration form.
5. Receive Registration Token and Application Information from HECO admin after you are approved.
6. You’re now a registered third party with HECO. You can use your Registration Token to retrieve a Client Access Token and begin the Customer Authorization and Data Exchange processes.

Scenario 2: HECO Admin Perspective

1. Query being built by HECO will trigger a notification when a third-party registers with the portal.
2. HECO Admin logs into the portal as an admin and accesses the Manage Green Button Connect page from the Admin dropdown:



3. Third parties who have submitted the registration form will appear on this page. The columns included are:
 - a. Third Party name
 - b. Active radio button
 - c. Date Registered On
 - d. Date Expires On
 - e. Edit
 - f. Generate Metadata
 - g. Delete

Third Party	Active	Date Registered On	Date Expires On	Edit	Generate Metadata	Delete
test_client	<input type="checkbox"/>	2/10/2021 8:23:20 AM	02/25/2022	Edit	Generate Metadata	Delete

4. The admin should select the “Edit” button to review the information that the third party has submitted with their registration form. They can use this time to vet the third party and change any information needed. Once they have internally approved the third party, the admin should click “Save” to return to the Manage Green Button Connect page.
5. The Date Expires On field will automatically be set to one year after the third party submitted their registration form. This date can be changed at any point by a HECO admin.
 - a. If the Date Expires On field has passed, the third party’s registration access token will be inactive, and they will not be able to retrieve any customer data.
 - b. By extending this expiration date through the UI, HECO admins can extend the expiration date of the same registration token for the third party. This allows the third party to access data using the same access token without re-registering after a year.
6. After the third party has been vetted, the HECO admin must select the “Active” radio button to finish the registration process. If this button isn’t checked, the third party will not be enabled, and won’t be able to retrieve data for any customers.
7. The admin should then click the “Generate Metadata” button to generate the third party’s Registration Token along with their additional metadata, which will be displayed in an overlay:

Third Party Metadata

Client Name:	Test Client
Status:	False
Authorization Endpoint:	https://localhost/lowertown/OAuthServer/Authorize
Token Endpoint:	https://localhost/lowertown/OAuthServer/Token
Bulk Request URI:	https://services.mymeter.co/221/GBC/espi/1_1/Resource/Batch/Bulk/
Resource Endpoint:	https://services.mymeter.co/221
Client Id:	test_client
Client Secret:	9TLUcEEw
Client Id Issued At:	2/10/2021 8:23:20 AM
Client Secret Expires At:	2/25/2022 12:00:00 AM
Registration Client URI:	
Registration Access Token:	7w8bKi2RdYPZAXn8zeCC8KA1zITTHzoDVYo2zW+ZeYl/K+3xCa99ZpAG3iiPVMhEj5cgphB+0dAD/eYrBH/eFNXxpclu1NZpPoVjdV9TWdfZaF5VnCiPBqL2TyJ8SYG8Lib+Flwl8vhDKDvBsADC061eKvKdIeYTYzYugmpdl58uhF+rUoOH1QOWynQEvk22tEYEBW4pmRnaBXk4ZD+vs9fjD0Jqg5Ja9heouermeEiAsyfY4dAcNBhYopIBSeX8qT+lulRYmWxnRQoHLw1d5k/lmoxxbL+st51RW2FdfzXfo5os+JGEd3Jk7qdcDjJX3To5HSwUh6/DMLSVbbgpehCSkqZ02fEhR1dPbzeUMtFa8B6t2U1YR1PJ/6ZKhCcqFH/CxkeroWb72TrmKdP2Kc0849gv1n/ubpW3x5bN0lbbIRB+ASE5nYt8OG9Kv7lwuqYnfyhlykVeNvZJbdqz7ZDOWNNEWAPA+gM2h9DkQfwZNgeHbeqrG+rhpF5dJl7ATn1OxMHJrtPAK3VPOnjw==
AuthorizationServerURI:	

- a. This data needs to be securely sent to the third party by a HECO admin. This information will be accessible from the UI at any time by clicking the “Generate Metadata” button. The Client Secret Expires At field will be updated if the Date Expires On field on the Manage Green Button Connect page is updated.
 - b. The third party should save this information and use accordingly during the Customer Authorization and Data Exchange processes. See [Section 4: Complete List of Endpoints and Their Uses](#) for detailed information about this data and when it should be used.
8. If HECO would like to remove a third party from this admin page, they can click the “Delete” button. This will remove the third party from the UI but will not remove their Client record from the database.

Section 2: Customer Authorization (OAuth)

1. Retail customer is redirected from 3P website to MyMeter portal.
 - Using a GET request to the /authorize endpoint using an HTTP Content-Type value of "application/x-www-form-urlencoded" format.

a. Example authorize call:

`https://demo.mymeter.co/hecolab/OAuthServer/Authorize?scope=FB%3D1_3;HistoryLength=34128000;IntervalDuration=900&client_id=gbc&response_type=code&redirect_uri=https%3A%2F%2Fexample.com%2Fsomewhere`

b. Authorize call parameters:


- **response type**
 - Value must be set to "code."
- **client_id**
 - Client name that was entered by the third party upon registration.
- **Redirect_uri**
 - Where the retail customer will be redirected after authorization.
 - type="xs:anyURI" maxOccurs="unbounded", e.g., "https://server.example.com/ThirdParty/espi/1_1/OAuthCallBack"
- **Scope:** see [this link](#) for more info:
 - The value of the scope parameter is expressed as a list of space-delimited, case-sensitive strings. If the value contains multiple space-delimited strings, their order does not matter, and each string adds an additional access range to the requested scope.
 - Define which types of data they are requesting and for what period
 - Can use however many of these parameters needed:
 - **FunctionBlock:** format: list
 - [required] the list of data and functionality being requested in the authorization, separated by underscores. **Note that the function block for each desired endpoint must be included in this scope parameter.**
 - **HistoryLength:** format: integer
 - How far back you want historical data (in seconds). If you don't want historical data, set this to zero.
 - e.g., 2 years would be: HistoryLength=63072000
 - **PreferredAuthEndDate:** format: integer
 - A UNIX timestamp (in seconds) for when you want to automatically revoke the authorization. if you don't want to set an automatic end date, set this to zero.
 - **Additional Scope:** format: list
 - A list of additional universal and utility-specific scope options separated with underscores. See [this link](#) for list of possible values.
 - Function blocks supported by MyMeter*:
 - 1 – Common – common services
 - 3 – Connect My Data; specific endpoints:

- ApplicationInformation
- Authorization
- Batch
- ServiceStatus
- 32 – Resource-Level REST; specific endpoints:
 - Batch
 - ElectricPowerQualitySummary
 - ElectricPowerUsageSummary
 - IntervalBlock
 - LocalTimeParameters
 - MeterReading
 - ReadingType
 - UsagePoint
- 33 – Management REST services; specific endpoints:
 - ApplicationInformation
 - Batch
 - ElectricPowerQualitySummary
 - ElectricPowerUsageSummary
 - IntervalBlock
 - LocalTimeParameters
 - MeterReading
 - ReadingType
 - UsagePoint
- 35 – REST for bulk transfer; specific endpoints:
 - Batch
- 41 – Manage ApplicationInformation resource; specific endpoints:
 - ApplicationInformation
- 44 – Manage Authorization resource; specific endpoints:
 - Authorization
- 99 – Upload a bulk transfer file; specific endpoints:
 - Batch

* Please see [this link](#) for full calls to each endpoint.

- state
 - Optional opaque value to maintain state between request and callback.
 - String of undefined length.

2. Retail customer reaches the MyMeter Landing page and enters their CDC username & password:

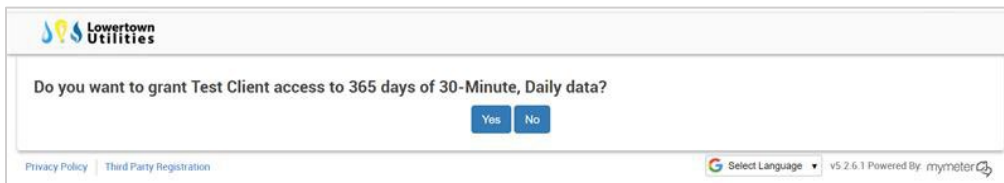


Hawaiian
Electric

Existing Customer? [Log on from our Online Customer Service Center >](#)

Remember Me [Forgot your Password?](#)

3. Call is made to CDC's [accounts.login REST API](#) to authenticate CDC user.
 - For testing, it is common to use a user key and secret key pair in place of the user's CDC username and password.
 - Credentials for test:
 - User id: gridmod1
 - Password: TheHeco12345
 - NOTE: This validation will currently only work with residential/UCES customers who have logged into the portal via SSO before.
4. [Retail customer is presented with an authorization confirmation screen.](#)
9. This includes information about the scope they are approving. The example below shows what this screen would look like for a customer approving access to Test Client for 365 days of 30 minute and daily data.
10. This data being approved is based off the parameters sent in the initial GET request from the third party:
 - Client_id = Test Client
 - HistoryLength = 365 days
 - IntervalDuration = Daily



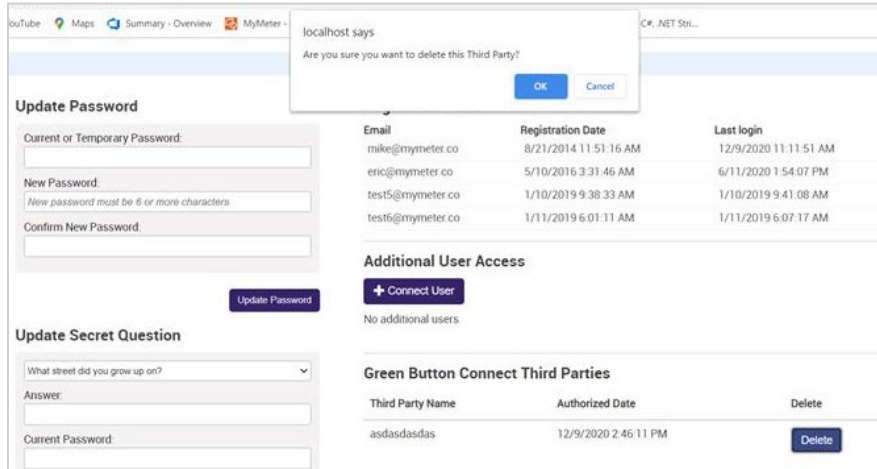
11. When testing, it should be verified that the information being displayed in the UI matches what was sent in the initial GET request from the third party.
 - Client Name will correspond with the client_id in the GET request.
 - HistoryLength corresponds to the number of seconds of historical data the customer is granting access to – e.g., 365 days would be HistoryLength = 31536000
 - IntervalDuration corresponds to the interval of data being authorized in seconds – e.g., 30-minute data would be IntervalDuration = 1800, daily data would be IntervalDuration = 86400.
5. [Retail customer is redirected to the third-party application.](#)
 - a. If they selected No:
 - a. we will redirect third-party application to the redirect_uri without an authorization code, and an error so the third party knows their authorization request was denied.
 - b. Example:

```
https://example.com/somewhere?error=access_denied&state=d71cebc8047a41a5b42661d9b2bf2f25
```


- c. Other possible error responses can be found here:
<https://tools.ietf.org/html/rfc6749#section-4.1.2.1>.
- b. If they selected Yes:
- d. an authorization code will be sent in the response to the redirect_uri:
- e. Example:

```
https://example.com/somewhere?code=f38647d645d54be28ae3efaab9565a47&state=d71cebc8047a41a5b42661d9b2bf2f25
```

- i.
 - i. [This link](#) provides more information about each item being passed in this step.
 - ii. **Code** - the authorization code issued by the authorization server (must be short-lived. Recommended lifetime of 10 minutes. Can only be used once. Bound to client id and redirect URI). Code = 1*VSCHAR
 - iii. **state** (required if state was present in request)
- f. **This is the end of the authorization process for the retail customer.**
 - i. After being redirected to the third-party application, they don't have to do anything else, unless they would like to remove the authorization or submit a new authorization.
 - ii. If the retail customer would like to remove this authorization from the third party, they will need to log into the portal and navigate to the User Profile page. This is where they will see a list of third parties they have authorized, including the third party's name, the authorization date, and a Delete button:



- iii. If the user clicks Delete, they will be prompted with a message asking them to confirm the request. Once they agree, the third party will be removed from this list, and all authorizations will be removed. This means that if the third party were to try to request data using their previously authorized access token, they would be denied.

6. Third party converts authorization code to Access Token.

- 12. Once the third party has received their authorization code, they can convert this code to an Access Token with a GET request to the AI server's /token endpoint using an HTTP Content-Type value of "application/x-www-form-urlencoded" format.
- 13. The values that are passed in this request include:

- a. **Grant_type:** value must be set to “authorization code.”
 - b. **Code:** authorization code received from AI’s server in previous step. Must be UTF-8 encoded.
 - c. **redirect uri:** The client’s redirection endpoint previously established with the authorization server during the client registration process or when making the authorization request. It MUST be identical to the redirect_uri value provided in the authorization server’s /authorize endpoint request.
 - d. **Scope:** this scope parameter must match the scope parameter included in the authorization request. See Section 2b (Authorize Call Parameters) above for specifics of the Scope parameter.
14. The Third Party’s client_id is encoded using the “application/x-www-form-urlencoded” encoding algorithm described in [Appendix B of RFC 6749](#), and the encoded value is used as the username: the client_secret is encoded using the same algorithm and used as the password
15. Example request:

```
curl \
-u "$CLIENT_ID:$CLIENT_SECRET" \
-d "grant_type=authorization_code" \
-d "code=f38647d645d54be28ae3efaab9565a47" \
-d "redirect_uri=https%3A%2F%2Fexample.com%2Fsomewhere" \
"https://utilityapi.com/DataCustodian/demo/oauth/token"
```

7. AI will authenticate the third party and their access token request.
- o This verification includes:
 - Ensuring the authorization code is issued to the authorized third party.
 - Ensuring the authorization code is valid.
 - Ensuring that the redirect_uri parameter is present and identical to the redirect_uri parameter used in the /authorize endpoint request.
 - Verifying that the 3P’s client_id and client_secret are from a registered third party.

8. If the third party passes this verification, AI will respond with the following information:

- **access token:** The access token issued by the authorization server
 - o access-token=1*VSCHAR
- **token type (bearer):** token-type=type-name/URI-reference
 - o The access token type provides the client with the information required to successfully utilize the access token to make a protected resource request. The only token value supported by the “NAESB REQ.21 ESPI ver. 3.3” standard is “bearer” [RFC 6750]. Value is case insensitive.
- **expires_in** (recommended. Tells the lifetime in seconds of the access token): expires-in=1*DIGIT
- **refresh token** (optional, recommended. Can be used to obtain new access tokens using the same authorization grant – the refresh token is typically long-lived): refresh-token=1*VSCHAR
- **Scope:** The scope of the access token

The third party has now completed the authorization process. They can use their Access Token for the lifetime indicated by the expires_in parameter. If the Access Token has expired and the third party has a refresh token, they may request a new Access Token using this Refresh Token.

16. Example response:

```
{
```

```
"token_type": "Bearer",
"access_token": "64999645a0b5449b871ad0333df6cb114415c9a5522d41118a0f7939d
d3f0208",
"refresh_token": "76fd81fd6a3b42a592eed9ff9b8d5cfda9d53324919643f4a45fe34d
664442f9",
"expires_in": 3600,
"scope": "FB=1_3_4_5_8_13_14_18_19_34_35_39_46;IntervalDuration=900_3600;B
lockDuration=daily;HistoryLength=34128000;SubscriptionFrequency=daily;AccountC
ollection=2",
"resourceURI": "https://utilityapi.com/DataCustodian/espi/1_1/resource/Sub
scription/1111",
"authorizationURI": "https://utilityapi.com/DataCustodian/espi/1_1/resourc
e/Authorization/1111",
}
```

Section 3: Data Exchange

1. The third-party requests data from with their customer-specific access token

Requests are from the Resource Server. Exchanges are made using the REST API interface. Included in this request is:

- **Resource Path:** path to resource server
- **Resource ID** (types of data), e.g.:
 - UsagePoint – the individual point of measured usage (e.g. a meter).
 - ReadingType – the characteristics associated with a set of meter readings.
 - IntervalBlock – a set of interval reading values for a specific time period
 - MeterReading – set of values obtained from the meter
 - ElectricPowerQualitySummary – a summary of the electric power quality for a specific time period.
 - LocalTimeParameters – information about the timezone for the usage point.
 - Authorization – information about a specific customer authorization to share data with a third party.
 - ApplicationInformation – information about third party registration status and base urls for the Green Button API.
- **Resource server** – AI is the resource server
- **Authorization** – this is the customer-specific access token (bearer) obtained in the customer authorization process.

The format of the request looks like:

HTTP GET Request

```
GET {ResourcePath}<resource>/{ResourceID*} HTTP/1.1
Host: {ResourceServer}
Content-Type: application/atom+xml
Authorization: Bearer {AccessToken}
```

Example*:

```
GET /DataCustodian/espi/1_1/resource/UsagePoint/1 HTTP/1.1
Accept-Encoding: gzip,deflate
Authorization: Bearer 2a85f4bd-30db-4b7d-8f41-b046b0566cb3
Content-Type: Application/atom+xml
Host: openspvm:8443
```

*** NOTE: it is recommended to start with the GET Service Status call (see section 4.p.) to ensure you can connect. Current status of 1 indicates success.**

Possible query parameters include:

- published-max, published-min
- updated-max, updated-min
- max-results
- start-index
- depth

2. AI's resource server validates the customer-specific access token with the authorization server.
3. AI's authorization server makes a call to CDC's API to verify that said customer is still active.
 - CDC documentation for this API call:
<https://developers.gigya.com/display/GD/accounts.getAccountInfo+REST>.
4. CDC responds with customer's status (active/inactive)
5. AI's authorization server responds to the resource server that the customer-specific access token is/isn't valid
17. Server confirms that the request is coming from a valid & registered third-party.
6. AI sends the requested data to the third party.
18. This data should correspond to the defined scope authorized by the retail customer.

Section 4: Complete List of Endpoints and Their Uses

The following includes information about each endpoint that third parties can access after registration. The metadata that is sent to them will include the specific endpoints to hit. This is the information that the third party will receive from HECO admins after registration:



If an authorization is specified in the request, include the authorization as an Authorization: Bearer <token_here> header in the API request.

Example:

```
curl \  
-H "Authorization: Bearer a2e7fd6a0c2f474c9a63ec18321c9989f805868237b14d1e9029948a2d7  
97121" \  
https://services.mymeter.co/310/GBC/esp/1_1/Resource/ServiceStatus
```

- Types of authorizations:

a. Registration_access_token

- This is the token that is sent from the HECO admin to the third party. This is used to access the ApplicationInformation endpoint.

b. Client_access_token

- This token is obtained by making a Token request to the Token Endpoint. This is used to get information about authorizations and bulk data.

c. Access_token

- This token is obtained by exchanging an authorization code for an access token after customer authorization. This is used to obtain usage data.

2. Authorization Endpoint:

19. Endpoint that the third-party sends the retail customer to so that they can complete the [Customer Authorization \(OAuth\)](#) process.
20. Example:

```
https://demo.mymeter.co/hecolab/OAuthServer/Authorize?response_type=code&client_id=12345&redirect_uri=https%3A%2F%2Fexample.com%2Fsomewhere&scope=FB%3D4_46&state=d71cebc8047a41a5b42661d9b2bf2f25
```

3. Token Endpoint:

21. Endpoint that the third-party will hit to exchange their authorization code for a data access token.
22. Example:

```
# Request
curl \
  -u "$CLIENT_ID:$CLIENT_SECRET" \
  -d "grant_type=authorization_code" \
  -d "code=f38647d645d54be28ae3efaab9565a47" \
  -d "redirect_uri=https%3A%2F%2Fexample.com%2Fsomewhere" \
  "https://utilityapi.com/DataCustodian/demo/oauth/token"
# Response
{
  "token_type": "Bearer",
  "access_token": "64999645a0b5449b871ad0333df6cb114415c9a5522d41118a0f7939d
d3f0208",
  "refresh_token": "76fd81fd6a3b42a592eed9ff9b8d5cfda9d53324919643f4a45fe34d
664442f9",
  "expires_in": 3600,
  "scope": "FB=1_3_4_5_8_13_14_18_19_34_35_39_46;IntervalDuration=900_3600;B
lockDuration=daily;HistoryLength=34128000;SubscriptionFrequency=daily;AccountC
ollection=2",
  "resourceURI": "https://utilityapi.com/DataCustodian/espi/1_1/resource/Sub
scription/1111",
  "authorizationURI": "https://utilityapi.com/DataCustodian/espi/1_1/resourc
e/Authorization/1111",
}
```

4. Bulk Request URI:

23. The endpoint that third parties will hit if they want to retrieve bulk data that has been authorized by multiple retail customers.
24. Example: https://services.mymeter.co/310/GBC/espi/1_1/Resource/Batch/Bulk/bulkid
25. Authentication: client_access_token

5. Resource Endpoint:

26. The endpoint the third party should hit if they want to retrieve information about their authorization(s) or pull data for individual authorizations.

a. GET Application Information

- Used by third party to retrieve their Application Information.
- Authentication: registration_access_token
- Response: ApplicationInformation object
- Example: **[https://services.mymeter.co/310
GBC/espi/1_1/Resource/ApplicationInformation](https://services.mymeter.co/310/GBC/espi/1_1/Resource/ApplicationInformation)**

b. GET Authorization

- Get information about a specific customer's authorization.
- Authentication: client_access_token
- Response: Authorization object
- Example: **[https://services.mymeter.co/310 /GBC/espi/1_1/Resource/Authorization](https://services.mymeter.co/310/GBC/espi/1_1/Resource/Authorization)**

c. GET Authorizations

- Get list of authorizations of all retail customers.
- Authentication: client_access_token
- Response: List of authorization objects
- Example: **[https://services.mymeter.co/310
/GBC/espi/1_1/Resource/Authorization/:authorizationId](https://services.mymeter.co/310/GBC/espi/1_1/Resource/Authorization/:authorizationId)**

d. GET Usage Points

- Get a list of usage points (meters) for an authorization.
- Authentication: access_token
- Response: List of Usage Point entries
- Example: **[https://services.mymeter.co/310 /GBC/espi/1_1/Resource/UsagePoint](https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint)**

e. GET Usage Point

- Get the related URLs for a specific usage point (meter)
- Authentication: access_token
- Response: UsagePoint object
- Example: **[https://services.mymeter.co/310
/GBC/espi/1_1/Resource/UsagePoint/UsagePointId](https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint/UsagePointId)**

f. GET Electric Power Quality Summary

- Get the list of electric power quality summaries for a specific meter (usagePointId).
- Authentication: access_token
- Response: List containing all electric power quality summary entries.
- Example: **[https://services.mymeter.co/310
/GBC/espi/1_1/Resource/UsagePoint/:usagePointId/ElectricPowerQualitySummary](https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint/:usagePointId/ElectricPowerQualitySummary)**

g. GET Meter Readings

- Get the list of meter readings for a specific meter (usagePointId)
- Authentication: access_token

- Response: list of meter reading entries
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint/:usagePointId/MeterReading**
- h. GET Meter Reading
- Get a specific meter reading (meterReadingId) from a specific meter (usagePointId)
 - Authentication: access_token
 - Response: meter reading object
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint/:usagePointId/MeterReading/:meterReadingId**
- i. GET Usage Summaries
- Get the list of usage summaries (e.g. bills) for a meter.
 - Authentication: access_token
 - Response: list of usage summaries for meter.
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint/:usagePointId/UsageSummary**
- j. GET Usage Summary
- Get a specific usage summary (e.g. bill) for a meter.
 - Authentication: access_token
 - Response: usage summary for meter.
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/UsagePoint/:usagePointId/UsageSummary/:usageSummaryId**
- k. GET Local Time Parameters
- Get information about all local time parameters.
 - Authorization: access_token
 - Response: list of all local time parameters
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/LocalTimeParameters**
- l. GET Local Time Parameter
- Get information about a specific local time parameter.
 - Authorization: access_token
 - Response: local time parameter object.
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/LocalTimeParameters/:localTimeParameterId**
- m. GET All Meter Readings
- Get meter readings for all authorizations
 - Authorization: access_token
 - Response: list of meter readings for all authorizations
 - Example: Get information about all local time parameters.
 - Authorization: access_token

- Response: list of all local time parameters
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/MeterReading**
- n. GET Interval Block
- Get information about a particular interval block
 - Authorization: access_token
 - Response: Interval Block object
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/IntervalBlock/:intervalBlockId**
- o. GET Reading Type
- Get information about a specific reading type (characteristics associated with a set of meter readings)
 - Authentication: access_token
 - Response: reading type object
 - Example: **https://services.mymeter.co/310/GBC/espi/1_1/Resource/ReadingType/:readingTypeId**
- p. GET Service Status
- Check the status of the Green Button API.
 - Authentication: client_access_token
 - Response: service status object.
 - Example: **https://services.mymeter.co:447/resourceapi/310/GBC/espi/1_1/Resource/ServiceStatus**
-
- GET SS (service status):
 - o Client access token
 - o https://services.mymeter.co:447/resourceapi/310/GBC/espi/1_1/Resource/ServiceStatus
 - GET AI (application information):
 - o Currently – Client access token
 - o Should be – Registration access token
 - o https://services.mymeter.co:447/resourceapi/310/GBC/espi/1_1/Resource/ApplicationInformation
 - GET A (authorizations):
 - o Client access token
 - o https://services.mymeter.co:447/resourceapi/310/GBC/espi/1_1/Resource/Authorization?published-max=2021-12-31
 - o Returns information about each authorization associated with a third party
 - GET A 11 (authorization):
 - o Client access token
 - o Auth id (from access token response; currently **uuid** associated with specific authorization)
 - o https://services.mymeter.co:447/resourceapi/310/GBC/espi/1_1/Resource/Authorization/000003600000000000000000ac196543