Authorization and Bulk Transfer in Green Button Connect My Data

7/27/2015 9:19:22 PM

Table of Contents

[Table of Contents ii](#_Toc425795292)

[Table of Figures iii](#_Toc425795293)

[Table of Tables 4](#_Toc425795294)

[1 Overview 5](#_Toc425795295)

[1.1 Energy Services Provider Interface 7](#_Toc425795296)

[1.2 OAuth 2.0 Authorization flow from RFC 6749 and RFC 6750 8](#_Toc425795297)

[1.3 Enhancements to the OAuth and ESPI Standards 11](#_Toc425795298)

[2 Energy Services Provider Interface Use Cases under OAuth 2.0 12](#_Toc425795299)

[2.1 Web URLs and Other Data Needed for the GBCMD Use Cases 12](#_Toc425795300)

[2.2 Use Case #1: Third Party Establishes Relationship with Data Custodian 21](#_Toc425795301)

[2.3 Use Case #2: ESPI Implementation of Authorization via OAuth 2.0 26](#_Toc425795302)

[2.4 Use Cases #3-12: Exchange of Resources 43](#_Toc425795303)

[2.5 Use Case #13: Bulk Transfer of Multiple Authorized Resources 55](#_Toc425795304)

[2.6 Encoding of OAuth 2.0 “scope” 56](#_Toc425795305)

[2.7 Security and Privacy for Green Button Connect My Data 61](#_Toc425795306)

[2.8 Behavior of GET UsageSummary with query parameters 64](#_Toc425795307)

[3 Appendix A: References 65](#_Toc425795308)

[4 Appendix B: APIs 66](#_Toc425795309)

[4.1 API Type Behaviors 66](#_Toc425795310)

[4.2 Access Tokens that Regulate Access to Green Button Connect My Data APIs 69](#_Toc425795311)

[4.3 APIs 70](#_Toc425795312)

[5 Appendix C: Function Blocks for Green Button Connect My Data 76](#_Toc425795313)

[6 Appendix D: ApplicationInformation and Authorization Resource Details 78](#_Toc425795314)

[6.1 ApplicationInformation 78](#_Toc425795315)

[6.2 Authorization 82](#_Toc425795316)

Table of Figures

[Figure 1: Green Button Connect My Data Context 5](#_Toc425795317)

[Figure 2: Data Custodian Point of View 28](#_Toc425795318)

[Figure 3: Data Custodian Point of View UML Model 29](#_Toc425795319)

[Figure 4: Third Party Point of View 32](#_Toc425795320)

[Figure 5: Retail Customer Starts at Third Party 33](#_Toc425795321)

[Figure 6: Authorization Storyboard 37](#_Toc425795322)

[Figure 7: Authorization Flow 38](#_Toc425795323)

[Figure 8: GET Exchanges with the DataCustodian 46](#_Toc425795324)

[Figure 9: PUT or POST Exchanges with the DataCustodian 49](#_Toc425795325)

[Figure 10: Notification 52](#_Toc425795326)

Table of Tables

[Table 1: HTTP Servers 12](#_Toc425795327)

[Table 2: URL Endpoints 13](#_Toc425795328)

[Table 3: Exchange Parameters 15](#_Toc425795329)

[Table 4: ESPI Use Case Parameters for Resource Exchange 19](#_Toc425795330)

[Table 5: CRUD Services for ESPI 44](#_Toc425795331)

[Table 6: HTTP Error Response 46](#_Toc425795332)

[Table 7: Access Tokens used in Green Button Connect My Data APIs 69](#_Toc425795333)

[Table 8: Green Button Connect My Data APIs 71](#_Toc425795334)

[Table 9: Function Blocks for Green Button Download My Data and Green Button Connect My Data 76](#_Toc425795335)

# Overview

This document presents the culmination of months of collaboration among industry experts to develop an interoperable and extensible authorization capability for Green Button Connect My Data (CMD). The architecture described herein integrates the NAESB REQ.21 Energy Services Provider Interface (ESPI) standard [1] and IETF OAuth 2.0 as specified in RFC 6749 [6] and RFC 6750 [7]. This solution responds to customer requirements by leveraging the strengths of each of these standards to achieve a robust, secure, and interoperable authorization scheme for Green Button Connect My Data.

The diagram below depicts the logical components of both Green Button Download My Data (GBDMD) and Green Button Connect My Data (GBCMD). The names and roles of the actors (the titles at the top of the boxes) are described in the next section.

Green Button Download My Data represents the access of Energy Usage Information (EUI) by a Retail Customer from a Data Custodian (typically a utility) via a web browser through the Data Custodian's consumer web portal. GBDMD results in a file containing the Retail Customer's Energy Usage Information being downloaded to the Retail Customer’s computer for further use.

The remainder of this document will focus on Green Button Connect My Data.

Green Button Connect My Data provides the Retail Customer with the ability to “authorize” a service provider (the “Third Party”) to access his or her EUI directly from the Data Custodian without further interaction with the Retail Customer.

This architecture presents a consistent mechanism for authorized exchange of energy usage information. The entity and information exchange requirements are defined in the ESPI standard. The Red arrows show the GBDMD flow. Blue arrows show actions requiring authorization mechanisms for GBCMD.

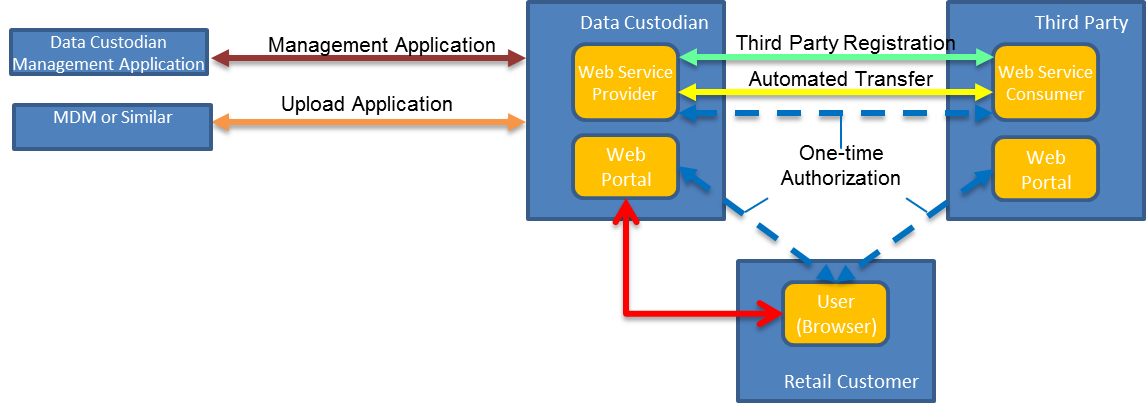


Figure : Green Button Connect My Data Context

To summarize, the figure introduces the following terms used in the balance of the document:

|  |  |
| --- | --- |
| **Data Custodian** | Entity holding Retail Customer Energy Usage Information. |
| **Third Party** | Third Party that is authorized to acquire Energy Usage Information from the Data Custodian. |
| **Retail Customer** | Customer who's Energy Usage Information' is being exchanged. |
| **Web Portal** | A web site hosted by a Data Custodian or Third Party and accessed by the Retail Customer to authorize or otherwise manage services related to exchange of EUI data. |
| **Web Service Consumer** | The client side of the web service sending requests to the provider. |
| **Web Service Provider** | An automated web service at the site hosted by the Data Custodian or Third Party. Typically RESTful web services in the ESPI context. |
| **User (Browser)** | The web browser or other web application through which the Retail Customer interacts with the Web Portal. For example in a hand-held device such as a cell phone, a mini browser might be opened for this purpose. On a desktop the default web browser would be utilized. |
| **Third Party Registration** | A process by which a Third Party establishes a relationship with the Data Custodian enabling them to perform authorization and data exchange. |
| **Data Custodian Management** | A trusted management application that communicates with the Data Custodian using RESTful web services. |
| **Upload Application** | A remote application with limited rights to upload meter data to the Data Custodian via RESTful web services. |

As shown in the figure above for GBCMD, a one-time authorization occurs via the web portals at the Data Custodian and Third Party. The Retail Customer uses his browser (User Agent) to navigate a series of web pages to determine the parameters of the authorization. We term this “One-time Authorization” to emphasize the fact that the authorization obtained may be long lived (e.g. many months) with many data exchanges (the “Automated Transfer”) between the Third Party and the Data Custodian in between.

Authorization in OAuth 2.0 is accomplished via a web browser, through web browser redirection, and by specific exchanges of data in the background.

ESPI standardizes the information that must flow between the Data Custodian and the Third Party. However the means of authenticating the Retail Customer, as well as other interactions between the Retail Customer and either Data Custodian or Third Party web portals are left up to each provider to determine (for the most part).

However, at critical points in the authorization scenario, “redirections” are required in order to transfer specific parameters “behind the scenes” between the Data Custodian and Third Party. “Behind the scenes” here means information is passed in the redirections that are not typically presented to the Retail Customer although the transfers occur via their web browser. These redirections are standardized.

This separation of standard versus non-standard parts of the scenarios allows for the maximum flexibility for service providers while preserving a guaranteed interoperable standard means of achieving authorization. The principle goal is for any Third Party to obtain Authorization from any Data Custodian in an interoperable fashion and without customization of their software.

Throughout this document the topic of authorizing information exchange is discussed. In that, the word “authorization” and several similar words refer to several distinct concepts. The concepts should be discernible from the context in which they are used but are summarized here for clarity:

Third Party This party is defined above

Authorization A process by which this three-party agreement is established

Authorization A data structure defined in ESPI and representing the three-party agreement between Retail Customer, Data Custodian, and Third Party

## Energy Services Provider Interface

ESPI defines the entities depicted in the diagram above[[1]](#footnote-2). Each of these entities is involved in orchestrated exchanges described through use cases in the ESPSI standard. These exchanges enable a Retail Customer to grant access to their Energy Usage Information (EUI) to an Third Party. This enables the Third Party to provide products and services to the customer based on the customer’s EUI. The Data Custodian holds the Retail Customer’s Energy Usage Information and shares that information with an Third Party energy service provider designated by the Retail Customer as an entity authorized to receive that EUI.

Data Custodians may restrict which Third Party service providers can gain access through the ESPI interfaces. Such constraints and arrangements are outside the scope of this design. However, this design describes how such arrangements can be automated.

There are some dozen Use Cases in the ESPI standard that describe how to construct, modify, revoke, terminate, and otherwise utilize authorizations for the exchange of EUI.

Section 2 describes the implementation of authorization related exchanges in the context of the ESPI use cases.

In ESPI, the exchange of information is based on the REST model of web services [8]. This model considers the exchange of information containing objects called “resources”. Resources can be exchanged through HTTP protocol services. In ESPI, there are three principle types of resources:

EUI One or more components of energy usage information including UsagePoint, MeterReading, IntervalBlock, ReadingType, LocalTimeParameters, UsageSummary, and ElectricPowerQualitySummary

ApplicationInformation A single data structure that contains definitions of all information shared between a Data Custodian and Third Party that govern their communications relationship.

Authorization A single data structure that represents the three-party agreement between Retail Customer, Data Custodian, and Third Party which covers the life of the authorization for EUI exchange.

## OAuth 2.0 Authorization flow from RFC 6749 and RFC 6750

It is the goal of the OAuth protocol to allow a three-party authorization to be established between the above parties without any private information about the *Resource Owner* being exchanged between the *Client* and the *Resource Server*. The standard adds an additional actor that allows for the authorization function to be supported independently of the data exchange by an Authorization Server. In many cases, the Resource Server and the Authorization Server are one and the same.

Once an authorization is established, a credential is established by which the Third Party can present the credential along with its request to exchange data which is the subject of the Authorization.

### Actors in the Authorization Sequence

OAuth and ESPI were developed by different independent standards bodies which use somewhat differing terminology for the actors. The table below relates each of the actors/parties from the two standards.

The basic flow of the Authorization establishment found in the standard(s) consists of a binding of the following actors and the information to be exchanged:

resource owner this is the ESPI Retail Customer.

client An application making protected resource requests on behalf of the resource owner and with its authorization. The term “client” does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices). **This is the ESPI Third Party.**

resource server The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens. **This is the primary role of the ESPI Data Custodian**.

authorization server The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization. **This is the authorization management role of the ESPI Data Custodian**.

user-agent typically a web browser or other software under the control of the resource owner.

It is the goal of the protocol to allow the three-party Authorization to be established without any private information about the Resource Owner being exchanged between the Client and the Resource Server. The standard adds an additional actor that allows for the “Authorization” function to be supported independently of the data exchange by an Authorization Server.

### Access Token Credentials

OAuth 2.0 defines the concept of access tokens. Access Tokens are credentials used to access protected resources. Green Button identifies the following access-tokens based on the OAuth RFC 6749 [6]. RFC 6750 [7] describes how these credentials are used in information exchange messaging:

#### Access Tokens for RetailCustomer Authorized Exchanges

access\_token An authorization-specific credential used to enable access to the resources shared between Retail Customer, Data Custodian, and Third Party. This is persisted in the Authorization resource structure. When accompanying an HTTP message request, evidences the sender’s authorization to make the exchange.

Obtained: During retail customer authorization process.

resourceUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Subscription/{subscriptionId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Batch/Subscription/%7bSubscriptionID%7d)

authorizationUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1//resource/Authorization/{authorizationId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1//resource/Authorization/%7bauthorizationId%7d)

refresh\_token A credential used to retrieve a new access\_token when the previous access\_token expires from the Data Custodian. This is persisted in the Authorization resource structure. Note that refresh\_token is only created for a RetailCustomer authorized relationship.

Obtained: During retail customer authorization process

#### Access Tokens for Management Entity Exchanges

data\_custodian\_access\_token A credential used by a trusted back-end application of the DataCustodian to populate and otherwise maintain the repository of the DataCustodian. This token enables access to any API.

Obtained: The access token is either “Configured or obtained using the OAuth client\_credentials based flow”.

resourceUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/\*](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/*)

authorizationUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization/{authorizationId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Authorization/%7bauthorizationId%7d)

client\_access\_token A credential (a specific access token) used by a Third Party to access collections of individually authorized resources via “Bulk Transfer”. This is persisted in an Authorization resource structure. Also authorizes access to all Authorization resources held by the ThirdParty with the corresponding DataCustodian.

Obtained: access token is obtained using the oauth client\_credentials based flow after the Third Party has completed registration.

resourceUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Bulk/{bulkId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Bulk/%7bbulkId%7d)

authorizationUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization/{authorizationId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Authorization/%7bauthorizationId%7d)

upload\_access\_token A credential used by an AMI management application to PUSH data to a DataCustodian through a backend interface. This token can only be used for resource.

Obtained: Using a Client Credentials flow similar to client\_access\_token above.

resourceUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource /Batch/RetailCustomer/{retailCustomerId}/UsagePoint](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource%20/Batch/RetailCustomer/%7bRetailCustomerID%7d/UsagePoint%20)

authorizationUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization/{authorizationId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Authorization/%7bauthorizationId%7d)

registration\_access\_token A credential (a specific access token) obtained during Third Party registration with the Data Custodian to enable access to the ApplicationInformation resource. This is persisted in the ApplicationInformation resource structure.

Obtained: During ThirdParty registration (either dynamic or manually) Always results in an ApplicationInformation and Authorization data structure.

resourceUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation/{applicationInformationId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/ApplicationInformation/%7bapplicationInformationId%7d)

authorizationUri: [https://services.greenbuttondata.org/DataCustodian/espi/1\_1//resource/Authorization/{authorizationId}](https://services.greenbuttondata.org/DataCustodian/espi/1_1//resource/Authorization/%7bauthorizationId%7d)

Note that access tokens are entirely under the control of the Data Custodian which can determine the life cycle of the access tokens and refresh tokens. The Third Party is required to react to the changes in the access token and/or refresh token as they occur.

## Enhancements to the OAuth and ESPI Standards

In addition to the application of OAuth 2.0 to the current revision of the ESPI standard, two important and expansion details are identified:

Scope In OAuth, the “Scope” is what is being authorized for exchange. Its meaning is usually implicit. In the ESPI implementation of “Scope” there is a detailed specification describing the content of information exchange being negotiated. This definition is also used by the Authorized Retail Customer, Third Party, and Data Custodian to negotiate mutually acceptable exchange parameters. See section 0

Encoding of OAuth 2.0 “scope” for details.

Validity Period The duration over which this authorization is in force unless otherwise modified. OAuth is typically used for short duration authorizations – often minutes and for a one time exchange of information. ESPI, on the other hand, uses a “one-time authorization” to enable repeated exchange of EUI between the Data Custodian and the Third Party over the course of many months. While the terms of the “authorization” may be modified, they are still expected to be longstanding. Note that there is a short time duration parameter in OAuth, expires\_in, which allows the Data Custodian to control the validity period of the credential specifically for security and privacy purposes. This concept is distinct from the ESPI Validity Period.

# Energy Services Provider Interface Use Cases under OAuth 2.0

## Web URLs and Other Data Needed for the GBCMD Use Cases

For the authorization process and data exchange, the Use Cases in this section make reference to URLs and other parameters used by web browsers and web services. In the text, such parameters and macros appear as strings in UpperCamelCase surrounded by braces “{}”. Here is a summary of these and their purpose. In the table which follows, find the endpoints listed and their description/purpose.

Note that to assist implementers, an on-line API is exposed at <http://energyos.github.io/OpenESPI-GreenButton-API-Documentation/API/>. This live API is designed to conform to the contents of this text document with an easy to use interface.

### URLS

Table : HTTP Servers

| URL MACRO | Description |
| --- | --- |
| {AuthorizationServer} | The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.  https://hostname:port# |
| {ResourceServer} | The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens. This is the ESPI Data Custodian.  https://hostname:port# |
| {ThirdParty} | An application making protected resource requests on behalf of the resource owner and with its authorization. The term “client” does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).  https://hostname:port# |

Table : URL Endpoints

| URL MACRO | Description |
| --- | --- |
| {AuthorizationPath} | /oauth/authorize |
| {AuthorizationServerAuthorizationEndpoint} | An OAuth 2.0 URL used by the client to obtain authorization from the resource owner via user-agent redirection.  {AuthorizationServer}{AuthorizationPath} |
| {AuthorizationServerRegistrationEndpoint} | A URL used by the client to register a Third Party with a Data Custodian via its {AuthorizationServer}  {AuthorizationServer}{RegistrationPath} |
| {AuthorizationServerTokenEndpoint} | An OAuth 2.0 URL used by the client to exchange an authorization grant for an access token, typically with client authentication.  {AuthorizationServer}­­{TokenPath} |
| {AuthorizationURL} | URL of this authorization  {AuthorizationServer}{AuthorizationPath}/{authorizationId} |
| {ClientConfigurationURL} | A URL used by a registered client to manage registration information. This URL is returned by the AuthorizationServer in the “registration\_client\_uri” field of the client information response.  {AuthorizationServerRegistrationEndpoint}/ApplicationInformation/{applicationInformationId} |
| {DataCustodianBatchRequestURL} | URL of DataCustodian’s Batch Request endpoint. The value is provided by the Data Custodian and cannot be modified by the ThirdParty. The field MUST be a valid URL. |
| {DataCustodianBulkRequestURL} | URL of DataCustodian’s Bulk Request endpoint. The value is provided by the Data Custodian and cannot be modified by the ThirdParty. The field MUST be a valid URL. |
| {DataCustodianSubscriptionRequestURL} | URL of Data Custodian’s Subscription Request endpoint. The value is provided by the Data Custodian and cannot be modified by the ThirdParty. The field MUST be a valid URL. |
| {DataCustodianThirdPartySelectionScreenURL} | URL of Data Custodian’s Third Party Selection web page. The field MUST be a valid URL. |
| {DataCustodianResourceEndpoint} | {ResourceServer}{ResourcePath} |
| {RegistrationPath} | /espi/1\_1/register |
| {ResourcePath} | /espi/1\_1/resource |
| {ResourceURL} | URL of accessible Resource  {DataCustodianResourceEndpoint}/<resource>/{resourceId}  where <resource> is an ESPI exposed resource (see section 2.4.2) |
| {SelectedThirdParty} | Third Party selected by Data Custodian’s Retail Customer. |
| {SelectedThirdPartyURL} | URL of Third Party selected by Data Custodian’s Retail Customer. |
| {ThirdPartyNotifyURL} | URL that points to a Third Party’s endpoint used by the Data Custodian’s {ResourceServer} to notify the Third Party of Batch data availability. The Third Party SHOULD then request the data from the Data Custodian’s {ResourceServer} using the URIs in the BatchList in the notification. |
| {ThirdPartyDataCustodianSelectionScreenURL} | URL of Third Party’s supported Data Custodian Selection web page. The field MUST be a valid URL. |
| {ThirdPartyDataUsagePolicyURL} | URL that points to a human-readable Policy document for the ThirdParty. The Data Custodian’s {AuthorizationServer} SHOULD display this URL to the end user. The policy usually describes how an end user’s data will be used by the ThirdParty. The value of the field MUST point to a valid web page. |
| {ThirdPartyHomepageURL} | URL of the homepage of the ThirdParty. The Data Custodian’s {ResourceServer} SHOULD display this URL to the end user in a clickable fashion. The value of this field MUST point to a valid web page. |
| {ThirdPartyLoginScreenURL} | URL of the Login Screen used by the Third Party to authenticate the identity of the Retail Customer. The field MUST be a valid URL. |
| {ThirdPartyLogoURL} | URL that references a logo for the Third Party. If present, the Data Custodian’s {ResourceServer} SHOULD display this image to the end user during approval. The value of the field MUST point to a valid image file. |
| {ThirdPartyDataCustodianSelectionScreenURI} | Optional URI that points to the Data Custodian selection screen of the Third Party service provider. The value of the field MUST point to a valid web page. |
| {TokenPath} | /oauth/token |
| {ThirdPartyName} | Human-readable name of the Third Party to be presented to the end user. |
| {ThirdPartyOAuthCodeCallbackURL} | An OAuth 2.0 client URL to which a resource owner’s user-agent is redirected by the {AuthorizationServer} after completing its interaction with the resource owner during an authorization grant request. The redirection URL MUST be an absolute URL as defined by [RFC3986].  {ThirdPartyServer}/espi/1\_1/OAuthCallBack |
| {ThirdPartyScopeSelectionScreenURL} | URL of the Scope Selection Screen used by the Retail Customer to select the characteristics of the Green Button data to be shared with the ThirdParty. The field MUST be a valid URL. |
| {ThirdPartyTermsOfServiceURL} | URL that points to a human-readable Terms of Service document for the ThirdParty. The DataCustodian {AuthorizationServer} SHOULD display this URL to the end user. The Terms of Service usually describe a contractual relationship between the end user and the Third Party that the end user accepts when authorizing the ThirdParty. The value of this field MUST point to a valid web page. |
| {ThirdPartyUserPortalScreenURL} | URL of a Third Party’s web page the Retail Customer user-agent is returned to after authorizing the Third Party access to the Retail Customer’s energy usage information (EUI). The value of this field MUST be a valid web page. |

### Parameters

In the message flows that follow, there are parameters in the message sequences that are summarized in the following table.

Table : Exchange Parameters

| Parameter Value MACRO | | Description / Purpose | |
| --- | --- | --- | --- |
| {AccessToken} | The access token issued by the DataCustodian {AuthorizationServer}. |
| {AccessTokenExpirationTime} | The lifetime in seconds of the access token. For example, the value 3600 denotes the access token will expire in one hour from the time the response was generated. |
| {AuthorizationCode} | The authorization code generated by the {AuthorizationServer}. The authorization code MUST expire shortly after it is issued to mitigate the risk of leaks. A maximum authorization lifetime of 5 minutes is REQUIRED. The Third Party MUST NOT use the authorization code more than once. If an authorization code is used more than once, the {AuthorizationServer} MUST deny the request and SHOULD revoke (when possible) all tokens previously issued based on that authorization code. The authorization code is bound to the {ThirdParty} identifier and {ThirdPartyOAuthCodeCallbackURL}. |
| {ClientIDIssuedTimestamp} | Time at which the {ThirdPartyClientID} was issued. The time is represented as the number of seconds from 1970-01-01T00:00:00Z as measured in UTC until the date/time. |
| {ClientSecretExpirationTimestamp} | Time at which the {ThirdPartyClientSecret} will expire (0 if it will not expire). The time is represented as the number of seconds from 1970-01-01T00:00:00Z as measured in UTC until the date/time. |
| {DataCustodianID} | Data Custodian Identification |
| {DataCustodianUserID} | User ID used by the DataCustodian to authentication the Retail Customer. |
| {DataCustodianUserPassword} | Password used by the DataCustodian to authenticate the Retail Customer. |
| {ESPIXMLResourceRepresentation} | The Atom+xml representation of a specified ESPI resource. |
| {RefreshToken} | The refresh token issued by the Data Custodian {AuthorizationServer}. The refresh token is used to obtain new access tokens. |
| {ResourcePostData} | The Atom/XML representation of an ESPI resource in the POST DATA portion of an HTML message. |
| {ScopeString} | A space separated list of scope values (as described in section 2.7 Encoding of OAuth 2.0 “scope”) the Third Party can use when requesting access tokens. |
| {SoftwareID} | An identifier for the software that comprises the Third Party Application. The software\_id is asserted by the Third Party software and is intended to be shared between all copies of the Third Party software. The value of this field MAY be a UUID [RFC4122]. |
| {SoftwareVersion} | A version identifier for the software that comprises a Third Party Application. The value of this field is a string that is intended to be compared using string equality matching. The value of the string software\_version SHOULD change on any update to the Third Party software. |
| {State} | An opaque value used by the Third Party to maintain state between the request and callback. The Data Custodian’s {AuthorizationServer} includes this value when redirecting the user-agent back to the Third Party. The state field is used to prevent cross-site request forgery as described in “The OAuth 2.0 Authorization Framework” [RFC6749] Section 10.12. |
| {ThirdPartyApplicationDescription} | A description of the Third Party’s application. |
| {ThirdPartyAuthorizationCode} | A set of client credentials used by the ThirdParty to authenticate with the Data Custodian’s {AuthorizationServer} during the OAuth 2.0 token request. The client credentials are supplied by the Third Party in the HTTP header’s Authorization field as a BASIC parameter and are composed of the {ThirdPartyClientID} and {ThirdPartyClientSecret} assigned by the Data Custodian’s {AuthorizationServer} during the Third Party’s Application registration process. The format of the ThirdPartyAuthorizationCode is {ThirdPartyClientID}:{ThirdPartyClientSecret}. |
| {ThirdPartyClientID} | A unique client identifier string issued by the Data Custodian’s {AuthorizationServer} representing the registration information provided by the ThirdParty. The client identifier is not a secret; it is exposed to the resource owner and MUST NOT be used alone for client authentication. The client identifier is unique to the Data Custodian’s {AuthorizationServer} and MUST NOT be currently valid for any other registered Third Party. |
| {ThirdPartyClientSecret} | The client secret MUST be unique for each {ThirdPartyClientID}. This value is used by confidential Third Parties to authenticate to the DataCustodian’s {AuthorizationServer} token endpoint as described in “The OAuth 2.0 Authorization Framework [RFC6749]” Section 2.3.1. |
| {ThirdPartyEmailContact1} | An element of an array of email addresses for people responsible for the Third Party. The Data Custodian’s {AuthorizationServer} MAY make these addresses available to end users for support requests for the ThirdParty. A Data Custodian’s {AuthorizationServer} MAY use these email addresses as identifiers for an administrative page for the Third Party. |
| {ThirdPartyEmailContact2} | An element of an array of email addresses for people responsible for the Third Party. The Data Custodian’s {AuthorizationServer} MAY make these addresses available to end users for support requests for the ThirdParty. A Data Custodian’s {AuthorizationServer} MAY use these email addresses as identifiers for an administrative page for the Third Party. |
| {ThirdPartyPhone} | The phone number of the Third Party organization to which access will be granted. (For debugging – not to be shared with customers). |
| {ThirdPartyUserID} | User ID used by the {ThirdParty} to authenticate the Retail Customer. |
| {ThirdPartyUserPassword} | Password used by the {ThirdParty} to authenticate the Retail Customer. |
| {ThirdPartyRegistrationAccessToken} | Access token used by the {ThirdParty} at the {ClientConfigurationURL} to perform subsequent changes to the {ThirdParty}’s registration. |

### Web Screens

| Web Screen Name MACRO | Description | Example |
| --- | --- | --- |
| {DataCustodianAuthorizationScreen} | A Data Custodian’s web page that allows the Retail Customer to approve the authorization of a Third Party to access their energy usage information (EUI) | /oauth/authorize |
| {DataCustodianHomeScreen} | URL of the homepage of the Data Custodian. | /home |
| {DataCustodianLoginScreen} | The Login Screen used by the Data Custodian’s {ResourceServer} to authenticate the identity of the Retail Customer. | /login |
| {DataCustodianThirdPartySelectionScreen} | A Data Custodian’s web page that allows the Retail Customer to select an Third Party. | /RetailCustomer/{retailCustomerId}/ThirdPartyList |
| {ThirdPartyDataCustodianSelectionScreen} | A Third Party’s web page that allows the Retail Customer to select a supported Data Custodian from who they wish to authorize the {ThirdParty} access to the Retail Customer’s energy usage information (EUI). | /RetailCustomer/{retailCustomerId}/DataCustodianList |
| {thirdPartyUserPortalScreenURI} | A Third Party’s web page that is the default landing web page for the {ThirdParty} | /home |
| {ThirdPartyLoginScreen} | The Login Screen used by the {ThirdParty} to authenticate the identity of the Retail Customer. | /login |
| {ThirdPartyScopeSelectionScreen} | A Third Party’s web page that allows the Retail Customer to select the characteristics of the Green Button data to be shared with the {ThirdParty}. | /RetailCustomer/{retailCustomerId}/ScopeSelection |
| {ThirdPartyUserPortalScreen} | A Third Party’s web page the Retail Customer user-agent is returned to after authorizing the {ThirdParty} access to the Retail Customer’s energy usage information (EUI). | /RetailCustomer/{retailCustomerId}/home |

### Exchange Syntax Parameters

The following table summarizes the specific parameters needed for various Green Button Connect My Data information exchanges.

Table : ESPI Use Case Parameters for Resource Exchange

| ESPI Use Case | Access Token To Use | HTTP Service | Endpoint[[2]](#footnote-3) | Resource | Positive Response Data |
| --- | --- | --- | --- | --- | --- |
| 1a | None[[3]](#footnote-4) | POST | {AuthorizationServerRegistrationEndpoint} | None | See section 2.1.4 |
| 1b | None[[4]](#footnote-5) | POST | {AuthorizationServerTokenEndpoint} | None | See section 2.1.4 |
| 2a | None[[5]](#footnote-6) | … | {AuthorizationServerAuthorizationEndpoint} | None | See section 2.3.2.1 |
| 2b | None[[6]](#footnote-7) | … | {AuthorizationServerTokenEndpoint} | None | See section 2.3.2.2 |
| 3 | access\_token | PUT | {DataCustodianResourceEndpoint} | /<resource>/ ⮩{resourceId} | None |
| 4 | access\_token | DELETE | {DataCustodianResourceEndpoint} | /<resource>/ ⮩{resourceId} | None |
| 5 | None | POST | {ThirdParty} | {ThirdPartyNotifyUri} | None |
| 6 | registration\_ ⮩access\_token | DELETE | {DataCustodianResourceEndpoint} | /ApplicationInformation/ ⮩{applicationInformationId} | None |
| ~~7[[7]](#footnote-8)~~ | ~~access\_token~~ | ~~POST~~ | ~~{DataCustodianResourceEndpoint}/~~ | ~~/subscription/ ⮩{SubscriptionD}~~ | ~~None~~ |
| 8 | access\_token | GET | {DataCustodianResourceEndpoint} | /Subscription/ ⮩{subscriptionId} | <feed>[[8]](#footnote-9) |
| ~~9[[9]](#footnote-10)~~ | ~~None~~ | ~~PUT~~ | ~~{ThirdParty}~~ | ~~/Subscription/ ⮩{subscriptionId}~~ | ~~None~~ |
| 10 | None | POST | {ThirdParty} | {ThirdPartyNotifyUri} | None |
| 11 | access\_token | GET | {DataCustodianResourceEndpoint} | /Subscription/ ⮩{subscriptionID} | <feed> |
| ApplicationInformation -- Update | registration\_ ⮩access\_token | PUT | {DataCustodianResourceEndpoint} | /ApplicationInformation/ ⮩{applicationInformationId} | None |
| Bulk | client\_ ⮩access\_token | GET | {DataCustodianResourceEndpoint} | /Batch/Bulk/ ⮩{bulkId} | <feed> |

## Use Case #1: Third Party Establishes Relationship with Data Custodian

Use Case #1 is used to establish a relationship between ThirdParty applications and a DataCustodian. During this process the ThirdParty provides the DataCustodian with information describing the type of application the ThirdParty will provide (i.e. smart phone, web server, etc.), ThirdParty webpage URLs (i.e. Terms of Service, Application Logo, ThirdParty Homepage, etc.), OAuth 2.0 information (i.e. token endpoint authorization method, grant types supported, scope elements supported, etc.). The DataCustodian provides the ThirdParty with a unique ThirdParty identification, ThirdParty Secret, and an Application Information Access Token. Although the registration process may be completed in an offline mode, Use Case #1 provides a method that can be used by DataCustodian implementations that support online registration of ThirdParty applications.

### Obtain OAuth Credentials

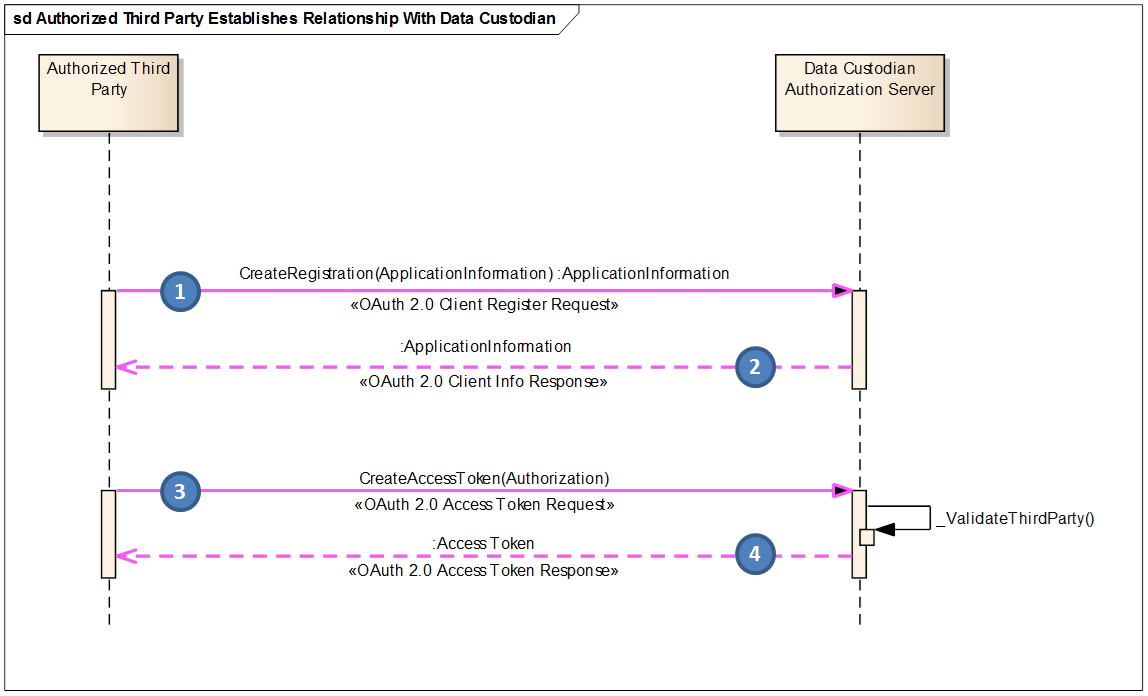


Figure 1: Third Party Establishes Relationship with Data Custodian

1. HTTP POST Request

POST {RegistrationPath} HTTP/1.1

Host: {AuthorizationServer}

Content-Type: application/json

Connection: keep-alive

Content-Length: nnnn

{

“logo\_uri”:“{ThirdPartyLogoURL}” ,

“client\_name”:”{ThirdPartyName}” ,

“client\_uri”:“{ThirdPartyHomepageURL}” ,

“redirect\_uri”:”{ThirdPartyOAuthCodeCallbackURL}”,

“tos\_uri”:“{ThirdPartyTermsOfServiceURL}” ,

“policy\_uri”:“{ThirdPartyDataUsagePolicyURL}” ,

“software\_id”:”{SoftwareID}”,

“software\_version”:”{SoftwareVersion}”,

“contacts”:[“{ThirdPartyEmailContact1}”, “{ThirdPartyEmailContact2}”, …] ,

“token\_endpoint\_auth\_method”:“client\_secret\_basic”,

“scope”:[“{ScopeString}”] ,

“grant\_types”:[”authorization\_code”, ”client\_credentials”, ”refresh\_token”] ,

“response\_types”:”code”,

“third\_party\_application\_description”:”{ThirdPartyApplicationDescription}”,

“third\_party\_application\_status”:[”1” | “2” | “3” | “4” | “5” | “6”], \*

“third\_party\_application\_type”:[”1” | “2” | “3”],\*

“third\_party\_application\_use”:[”1”| “2”| “3” | “4” | “5 ], \*

“third\_party\_phone”:”{ThirdPartyPhone}”,

“third\_party\_notify\_uri”:”{ThirdPartyNotifyURL}”,

“third\_party\_data\_custodian\_selection\_screen\_uri”:

“ThirdPartyDataCustodianSelectionScreenURI}”, (optional)

“third\_party\_login\_screen\_uri”:”{ThirdPartyLoginScreenURI}”, (optional)

“third\_party\_scope\_selection\_screen\_uri”:”{ThirdPartyScopeSelectionScreenURI}”,

“third\_party\_user\_portal\_screen\_uri”:”{ThirdPartyUserPortalScreenURI}”

}

Notes:

\* See *REQ.21 Energy Services Provider Interface* for meaning of listed values

1. HTTP POST Response

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Cache-Control: no-store

Pragma: no-store

Content-Type: application/json

{

“client\_secret”:”{ThirdPartyClientSecret}”,

“logo\_uri”:“{ThirdPartyLogoURL}”,

“client\_name”:”{ThirdPartyName}” ,

“client\_uri”:“{ThirdPartyHomepageURL}”,

“redirect\_uri”:”{ThirdPartyOAuthCodeCallbackURL}”,

“client\_id”:”{ThirdPartyClientID}”,

“tos\_uri”:“{ThirdPartyTermsOfServiceURL}” ,

“policy\_uri”:“{ThirdPartyDataUsagePolicyURL}” ,

“software\_id”:”{SoftwareID}”,

“software\_version”:”{SoftwareVersion}”,

“client\_id\_issued\_at”:“{ClientIDIssuedTimestamp}”,

“client\_secret\_expires\_at”:”{ClientSecretExpirationTimestamp}” | “0”,

“contacts”:[“{ThirdPartyEmailContact1}”, “{ThirdPartyEmailContact2}”, …],

“token\_endpoint\_auth\_method”:“client\_secret\_basic”,

“scope”:[“{ScopeString}”],

“grant\_types”:[”authorization\_code”, ”client\_credentials”, ”refresh\_token”] ,

“response\_types”:”code”,

“registration\_client\_uri”:“{ClientConfigurationURL}”, “registration\_access\_token”:”{ThirdPartyRegistrationAccessToken}”,

“data\_custodian\_id”:{dataCustodianId},

“data\_custodian\_application\_status”:[”1” | “2” | “3” | “4” | “5” | “6”],\*

“data\_custodian\_default\_batch\_resource”:“{ DataCustodianBatchRequestURL}”,

“data\_custodian\_default\_subscription\_resource”:

“{DataCustodianSubscriptionRequestURL}”,

“third\_party\_application\_description”:”{ThirdPartyApplicationDescription}”,

“third\_party\_application\_status”:[”1” | “2” | “3” | “4” | “5” | “6”],\*

“third\_party\_application\_type”:[”1” | “2” | “3”],\*

“third\_party\_application\_use”:[”1”| “2”| “3” | “4” | “5 ],\*

“third\_party\_phone”:”{ThirdPartyPhone}”,

“authorization\_server\_uri”:”{AuthorizationServer}”,

“third\_party\_notify\_uri”:”{ThirdPartyNotifyURL}”,

“authorization\_server\_authorization\_endpoint”:

”{AuthorizationServerAuthorizationEndpoint}”,

“authorization\_server\_token\_endpoint”:

“{authorizationServerTokenEndpoint”,

“authorization\_server\_registration\_endpoint”:

“{AuthorizationServerRegistrationEndpoint}”,

“data\_custodian\_bulk\_\_request\_uri”:“{DataCustodianBulkRequestURL}”,

“data\_custodian\_third\_party\_selection\_screen\_uri”:

“{DataCustodianThirdPartySelectionScreenURI}”,

“data\_custodian\_resource\_endpoint”:”{DataCustodianResourceEndpoint}”,

“third\_party\_data\_custodian\_selection\_screen\_uri”:

“ThirdPartyDataCustodianSelectionScreenURI}”,

“third\_party\_login\_screen\_uri”:”{ThirdPartyLoginScreenURI}”,

“third\_party\_scope\_selection\_screen\_uri”:”{ThirdPartyScopeSelectionScreenURI}”,

“third\_party\_user\_portal\_screen\_uri”:”{ThirdPartyUserPortalScreenURI}”,

“data\_custodian\_scope\_selection\_uri”:{dataCustodianScopeSelectionURI}”

}

Notes:

All data supplied by client in the client\_register request as well as any data fields provisioned or changed by the authorization server are included in the response.

\* See *REQ.21 Energy Services Provider Interface* for meaning of listed values

1. HTTP POST Request

POST {TokenPath} HTTP/1.1

Host: {AuthorizationServer}

Authorization: Basic {ThirdPartyAuthorizationCode}\*

Content-Type: application/x-www-form-urlencoded

grant\_type = client\_credentials

1. HTTP POST Response

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Cache-Control: no-store

Pragma: no-store

Content-Type: application/json; charset=UTF-8

{

“access\_token”:“{AccessToken}”

“token\_type”:“bearer”

“expires\_in”:{AccessTokenExpirationTime}

“scope”:“{ScopeString}”

“resourceURI”:“{ResourceURI}”

“authorizationURI”:“{AuthorizationURI}”

}

### Confirm Service Status

Use Case #1 also requires the ThirdParty to confirm their ability to access the Data Custodian {ResourceServer} by requesting the Data Custodian’s current Service Status. This requires the {ThirdParty} to utilize the DataCustodian’s **ReadServiceStatus** interface. The **ReadServiceStatus** interface is secured and requires the use of an OAuth 2.0 access token.

The ThirdParty uses the “registration\_access\_token” obtained during the registration process to be able to access the DataCustodian’s **ReadServiceStatus** interface as shown in Figure 2: Third Party request Data Custodian’s Service Status

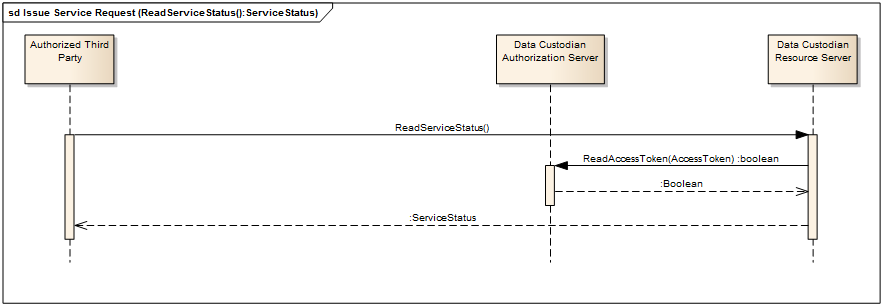


Figure 2: Third Party request Data Custodian’s Service Status

1. HTTP GET Request

GET {ResourcePath}/ReadServiceStatus HTTP/1.1

Host: {ResourceServer}

Authorization: Bearer {AccessToken}

1. HTTP GET Response

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: application/atom+xml

<ServiceStatus xsi:schemaLocation="http://naesb.org/espi espiDerived.xsd" xmlns="http://naesb.org/espi" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<currentStatus>1</currentStatus>

</ServiceStatus>

## Use Case #2: ESPI Implementation of Authorization via OAuth 2.0

### Overview

This section describes the implementation of the Authorization of exchange of Green Button data via ESPI standard protocols. Sample web screens are depicted notionally and color coded to indicate which web site displays the web screen – Data Custodian (orange background), Third Party (green background).

The transitions from screen to screen are shown in a story board format. The contents of the screens and the screen transitions are examples of potential implementations and are not part of the ESPI implementation standard. However, transitions between the web sites are standardized.

Following the screen flow diagram, an UML sequence diagram depicts the screen to screen story board scenario. While the screen to screen story board focuses on the Retail Customer’s web browser experience, the UML sequence diagrams focuses on the message flow which is most critical to the standardization of the OAuth 2.0 Authorization scenario. Only the redirections between the Data Custodian and Third Party web sites are shown. Internal redirections at either web site is considered a private matter.

After the UML diagram is a section that describes the message sequences in terms of HTTP headers and message content. These are the standardized and required exchanges. Although only required headers are shown, other headers are optional at the discretion of the implementer.

Oauth 2.0 works primarily through HTTP redirection of the Retail Customer’s browser. During these redirections, metadata about the OAuth 2.0 process is contained in the redirection messages as query parameters.

Once redirected to either the Data Custodian or the Third Party it is up to the redirected web site to progress to the next OAuth 2.0 step. The active web site may present the Retail Customer a series of proprietary screens and dialogs as appropriate to the needs of the host and the particular OAuth 2.0 step being performed. When ready to progress to the next OAuth 2.0 step, a redirect to the other host occurs.

As shown in Figure 1, the Retail Customer, through a web browser, navigates to the web portal of either the Data Custodian or Third Party to establish authorization for the Third Party to access their Energy Usage Information.

Authorization occurs in two distinct phases. The first phase is called the “*Scope Negotiation Phase*”. This phase has *no* OAuth 2.0 required message exchanges but is a Green Button ESPI implementation standard to establish and exchange a set of valid OAuth 2.0 scopes the Retail Customer may then select from during the “*OAuth 2.0 Authorization Phase*”. Once the “*Scope Negotiation Phase*” is completed, the “*OAuth 2.0 Authorization Phase*” occurs. The “*OAuth 2.0 Authorization Phase*” MUST begin at the Third Party.

The storyboard and UML diagram scenario depicted in this section, expect the Retail Customer to authenticate themselves with the Data Custodian and Third Party. Both the OAuth 2.0 and ESPI specifications have no requirements about how Retail Customer authentication is performed. This allows for maximum flexibility for Data Custodian and Third Party implementations while preserving a guaranteed interoperable standard for achieving authorization.

A Data Custodian may elect to implement independent Authorization and Resource servers. The interactions between the Authorization (which handles OAuth 2.0 Authorization and Retail Customer Authentication) and Resource servers is not standardized, nor depicted in the screen to screen storyboard or UML diagram scenario.

### Scope Negotiation Phase

#### Retail Customer Starts at Data Custodian

##### Story Board

This scenario begins when the Retail Customer starts at the web portal of the Data Custodian.

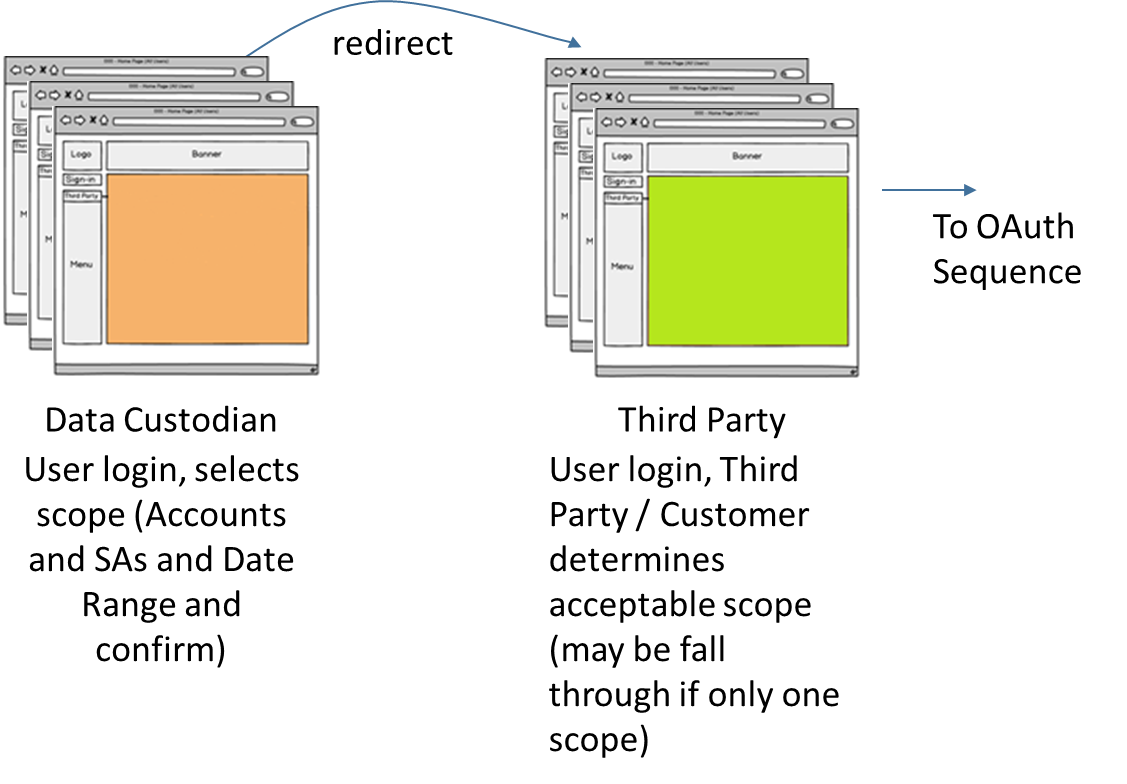


Figure : Data Custodian Point of View

The following steps represent “transitions” between the Data Custodian and Third Party application initiated by the Data Custodian’s Retail Customer. The steps below include all actions taken by the Retail Customer, starting with their initial browser selection and transitions to the start of the “*OAuth 2.0 Authorization Phase*” sequence:

1. Retail Customer selects the Data Custodian’s web site.
2. Retail Customer completes login and completes all Data Custodian required documents – the Data Custodian may have the Retail Customer navigate through any desired number of web screens to determine the availability of customer data the Third Party may access. These interactions are not a requirement of the Green Button ESPI implementation. At an appropriate step (determined by the Data Custodian), the customer is redirected to the selected Third Party “ScopeSelectionScreenURI” – {thirdPartyScopeSelectionScreenURI}. Contained within the HTTP redirection message are query parameters that identify the Data Custodian (DataCustodianID={dataCustodianId}) and the resources the Third Party may be granted access to by the Retail Customer (Scope=={ScopeString}&[scope={ScopeString}&]) during the “*Oauth 2.0 Authorization Phase*”.
3. Retail Customer completes login. Based on the metadata contained within the Data Custodian’s Third party “ScopeSelectionScreenURI” – {thirdPartyScopeSelectionScreenURI} -- HTTP redirection message the Third Party provides the Retail Customer with a web screen allowing them to select what Data Custodian resources the Retail Customer wants to share. If there is only one acceptable choice, the Third Party may proceed to the next step without displaying a web screen of resources to the Retail Customer. When the Retail Customer completes selecting the resources the Third Party is allowed to access the Third Party proceeds to the next step, which is a HTTP redirection message to the Data Custodian’s OAuth 2.0 authorization endpoint -- *{authorizationServerAuthorizationEndpoint}* -- as diagrammed in section [2.3.3 OAuth 2.0 Phase](#_OAuth_2.0_Phase)

##### UML Sequence Diagram

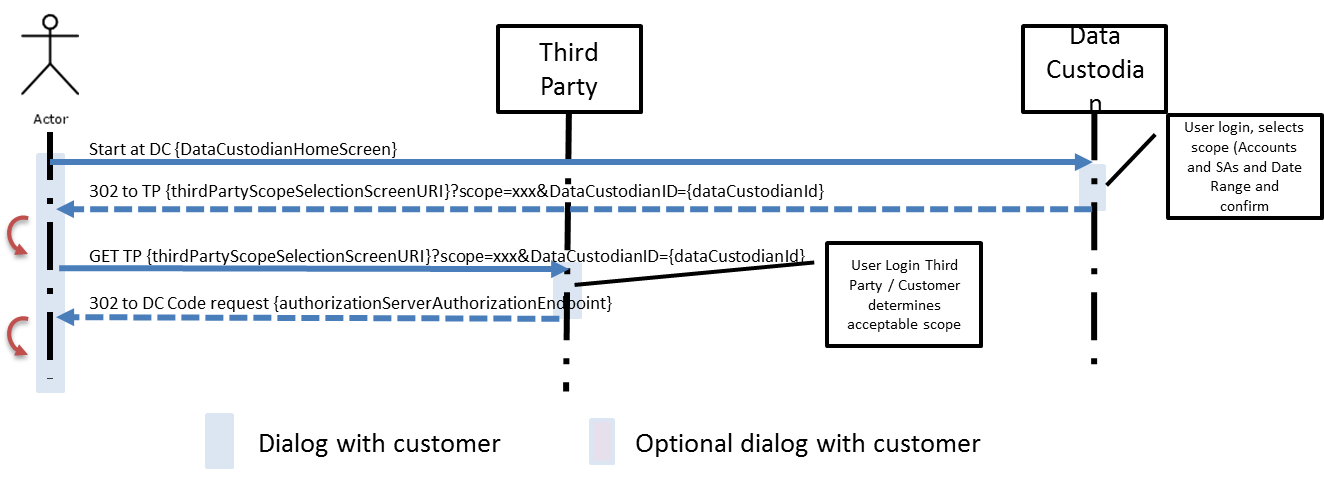


Figure : Data Custodian Point of View UML Model

##### Message Exchanges

1. **(Green Button *ESPI Implementation Standard*)** Transition: To Data Custodian web site

**Caused by:** Retail Customer enters Data Custodian web site address into browser

GET {DataCustodianHomeScreen} HTTP/1.1

Host: {ResourceServer}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

**Return:** 200 OK

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Set-Cookie: JSESSIONID=5D863CB2DE8CE163AB80CF5FF7478776; Secure; HttpOnly

Content-Type: text/html;charset=ISO-8859-1

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

Line-based text data: text/html {DataCustodianHomeScreen}

1. Intermediate messaging

The Data Custodian may perform any number of interactions with the Retail Customer as required to authenticate the Retail Customer, determine their account information and establish one or more valid OAuth 2.0 scopes which are then provided to a Third Party via a HTTP redirection message to the Third Party’s “*ScopeSelectionScreenURI*” as query elements. These interactions are not part of the Green Button ESPI implementation standard.

1. **(Green Button *ESPI Implementation Standard*)** Transition: To Third Party Selection Screen

**Caused by:** Retail customer completing Data Custodian web site processing

POST {Site implementation dependent} HTTP/1.1

Host: {ResourceServer}

**Return:** 302 Found (Redirect)

**Note**: The redirection includes Data Custodian ID – DataCustodianID={dataCustodianId} -- and Scope -- scope={ScopeString}&[scope={ScopeString}&] -- query parameters. The Scope element is a list of scope values supported by the Data Custodian for this Retail Customer.

HTTP/1.1 302 Found

Server: Apache-Coyote/1.1

Location: {ThirdParty}{ThirdPartyScopeSelectionScreen}?DataCustodianID={dataCustodianId]&scope={ScopeString}&[scope={ScopeString}&]\*

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

GET {ThirdPartyScopeSelectionScreenURL}  
?DataCustodianID={dataCustodianId]&scope={ScopeString}&[scope={ScopeString}&] HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: [Site implementation dependent]

Cookie: JSESSIONID=8C0590EFBA4003C2C720D6B8300123D0; Secure

Connection: keep-alive

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: text/html;charset=ISO-8859-1

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

Line-based text data: text/html {ThirdPartyScopeSelectionScreen}

See section [2.3.3 OAuth 2.0 Phase](#_OAuth_2.0_Phase) to continue to the “*OAuth 2.0 Authorization Phase*”.

#### Retail Customer Starts at Third Party

##### Story Board

This scenario begins when the Retail Customer starts at the Third Party web site.

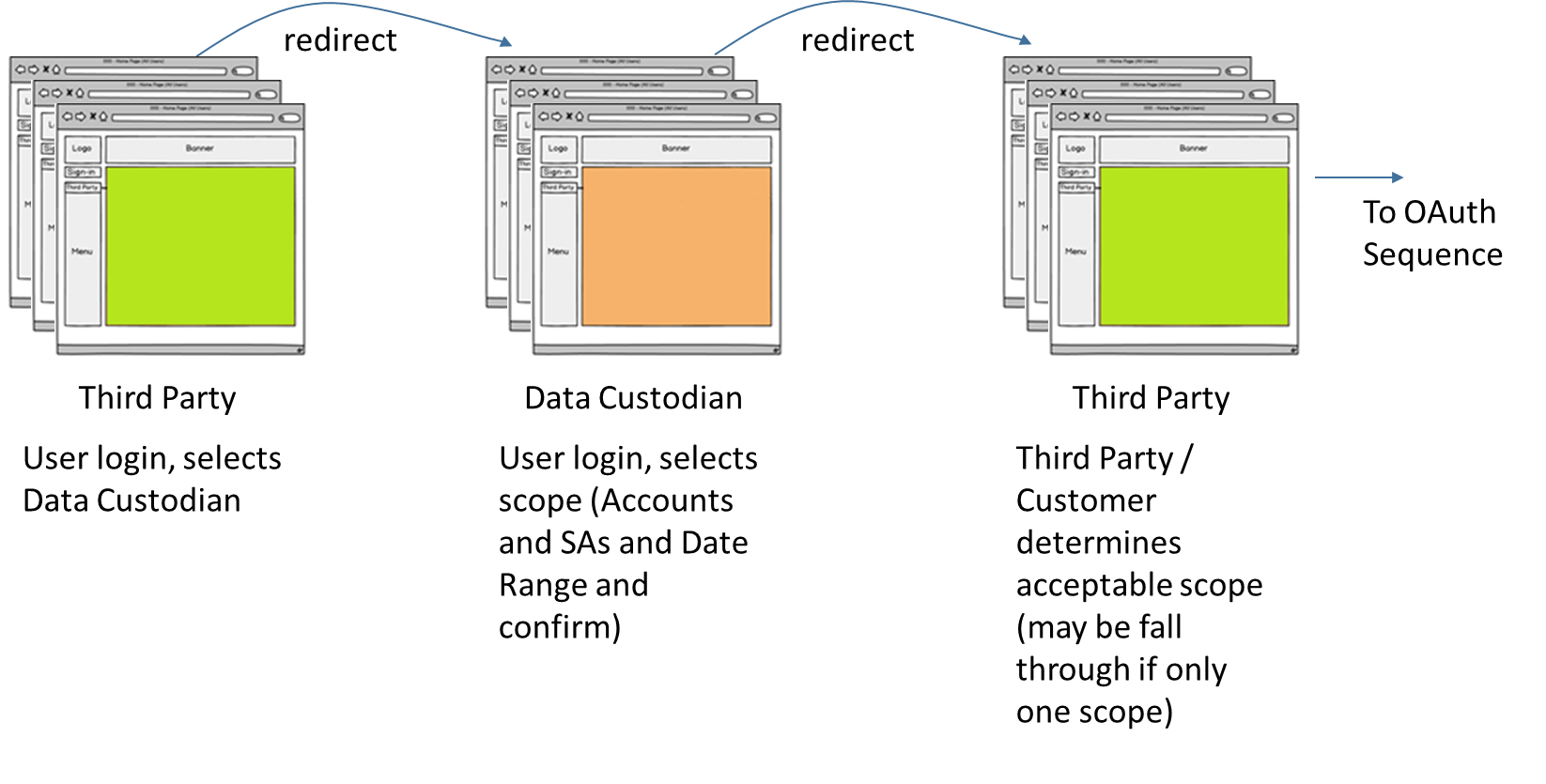


Figure : Third Party Point of View

The following steps represent “transitions” between the Data Custodian and Third Party application initiated by the Third Party’s Retail Customer. The steps below include all actions taken by the Retail Customer, starting with their initial browser selection and transitions to the start of the “*OAuth 2.0 Authorization Phase*” sequence:

1. Retail Customer selects the Third Party’s web site.
2. Retail Customer completes login and completes all Third Party required documents – the Third Party may have the Retail Customer navigate through any desired number of web screens. These interactions are not a requirement of the Green Button ESPI implementation. Once the Retail Customer has completed navigation of all desired web screens, the Third Party presents the Retail Customer with a web screen and allows them to select the Data Custodian they want to authorize the Third Party to access. When the Retail Customer has selected their Data Custodian, the Retail Customer’s browser is redirected to the selected Data Custodian’s “ScopeSelectionScreenURI” -- {dataCustodianScopeSelectionScreenUri}. Contained within the HTTP redirection message is a query parameter that identifies the Third Party to the Data Custodian (ThirdPartyID={thirdPartyId})
3. Retail Customer completes login and completes all Data Custodian required documents – the Data Custodian may have the Retail Customer navigate through any desired number of web screens to determine the availability of customer data the Third Party may access. These interactions are not a requirement of the Green Button ESPI implementation. At an appropriate step (determined by the Data Custodian), the customer is redirected to the selected Third Party “ScopeSelectionScreenURI” – {thirdPartyScopeSelectionScreenURI}. Contained within the HTTP redirection message are query parameters that identify the Data Custodian (DataCustodianID={dataCustodianId}) and the resources the Third Party may be granted access to by the Retail Customer (Scope=={ScopeString}&[scope={ScopeString}&]) during the “*Oauth 2.0 Authorization Phase*”.
4. Based on the metadata contained within the Data Custodian’s Third Party “ScopeSelectionScreenURI” – {thirdPartyScopeSelectionScreenURI} -- HTTP redirection message the Third Party provides the Retail Customer with a web screen allowing them to select what Data Custodian resources the Retail Customer wants to share. If there is only one acceptable choice, the Third Party may proceed to the next step without displaying a web screen of resources to the Retail Customer. When the Retail Customer completes selecting the resources the Third Party is allowed to access the Third Party proceeds to the next step, which is a HTTP redirection message to the Data Custodian’s OAuth 2.0 authorization endpoint -- *{authorizationServerAuthorizationEndpoint}* -- as diagrammed in section [2.3.3 OAuth 2.0 Phase](#_OAuth_2.0_Phase)

##### UML Sequence Diagram

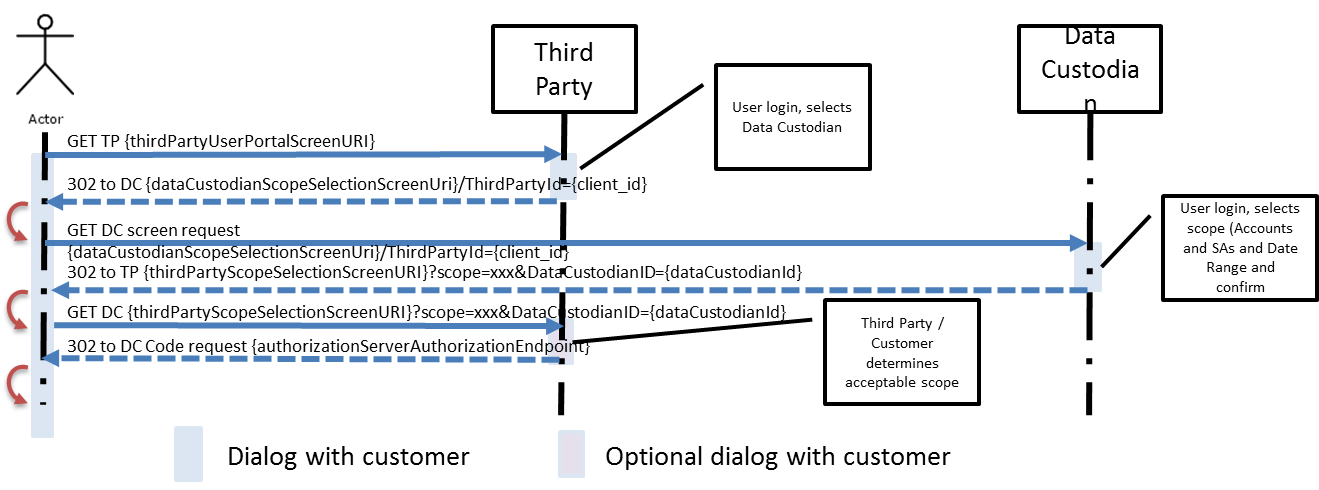


Figure : Retail Customer Starts at Third Party

##### Message Exchanges

1. **(Green Button *ESPI Implementation Standard*)** Transition: To Third Party web site

**Caused by:** Retail Customer enters Third Party web site address into browser

GET {ThirdPartyHomeScreen} HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

**Return:** 200 OK

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Set-Cookie: JSESSIONID=F9128C4A30686004CDAFED270C9A667F;

Path={ThirdPartyHomePageURL}; Secure; HttpOnly

Content-Type: text/html;charset=ISO-8859-1

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

Line-based text data: text/html {ThirdPartyHomeScreen}

1. Intermediate messaging

The Third Party may do any number of interactions with the Retail Customer as required to authenticate the Retail Customer, determine their account information and identify the Data Custodian that will be the source of data. These interactions are not part of the Green Button ESPI implementation standard.

1. **(Green Button *ESPI Implementation Standard*)** Transition: To Data Custodian Selection Screen

GET {ThirdPartyDataCustodianSelectionScreen} HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: [Site Implementation Dependent]

Cookie: JSESSIONID= F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: text/html;charset=ISO-8859-1

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

Line-based text data: text/html {ThirdPartyDataCustodianSelectionScreen}

1. **(Green Button *ESPI Implementation Standard*)** Transition: To Data Custodian web site

**Caused by:** Retail customer clicks a selection on the Data Custodian selection web screen

POST {ThirdPartyDataCustodianSelectionScreen} HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: {ThirdParty}{ThirdPartyDataCustodianSelectionScreen}

Cookie: JSESSIONID= F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: nnnn

Data\_custodian={DataCustodianID}&Data\_custodian\_URL={DataCustodianHomeScreen}

**Return:** 302 Found (Redirect)

**Note**: The redirection includes Third Party ID – ThirdPartyID={thirdPartyClientId} -- and Scope -- scope={ScopeString}&[scope={ScopeString}&] -- query parameters. The Scope element is a list of scope values supported by the Third Party for this Retail Customer.

HTTP/1.1 302 Found

Server: Apache-Coyote/1.1

Location: {ResourceServer}{DataCustodianScopeSelectionScreen}  
?scope={ScopeString}&[scope={ScopeString}&]ThirdPartyClientID={ThirdPartyClientID}

GET {DataCustodianScopeSelectionScreenURL}  
?scope={ScopeString}&[scope={ScopeString}&]ThirdPartyID={ThirdPartyClientID} HTTP/1.1

Host: {ResourceServer}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: {ThirdParty}{DataCustodianSelectionScreen}

Cookie: JESSIONID= F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

1. **(Green Button *ESPI Implementation Standard*)**  Transition: to Third Party Scope Selection Screen

GET {DataCustodianScopeSelectionScreen}  
?scope={ScopeString}&[scope={ScopeString}&]ThirdPartyID={ThirdPartyClientID} HTTP/1.1

Host: {ResourceServer}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: {ResourceServer}{DataCustodianLoginScreen}

Cookie: JSESSIONID= F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

HTTP/1.1 302 Found

Server: Apache-Coyote/1.1

Location: {ThirdParty}{ThirdPartyScopeSelectionScreen}  
?scope={ScopeString}&[scope={ScopeString}&]DataCustodianID={DataCustodianID}

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

GET {ThirdPartyScopeSelectionScreen}? scope={ScopeString}&[scope={ScopeString}&]\*DataCustodianID={DataCustodianID} HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: {ResourceServer}{DataCustodianScopeSelectionScreen}/  
?scope={ScopeString}&[scope={ScopeString}&]ThirdPartyID={ThirdPartyClientID}

Cookie: JSESSIONID= F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: text/html;charset=ISO-8859-1

Content-Length: nnnn

Date: Wed, 06 Mar 2013 19:22:08 GMT

Line-based text data: text/html {ThirdPartyScopeSelectionScreen}

See section [2.3.3 OAuth 2.0 Phase](#_OAuth_2.0_Phase) to continue to the “*OAuth 2.0 Authorization Phase*”.

### OAuth 2.0 Phase

#### Story Board

Authorization always begins at the Third Party.

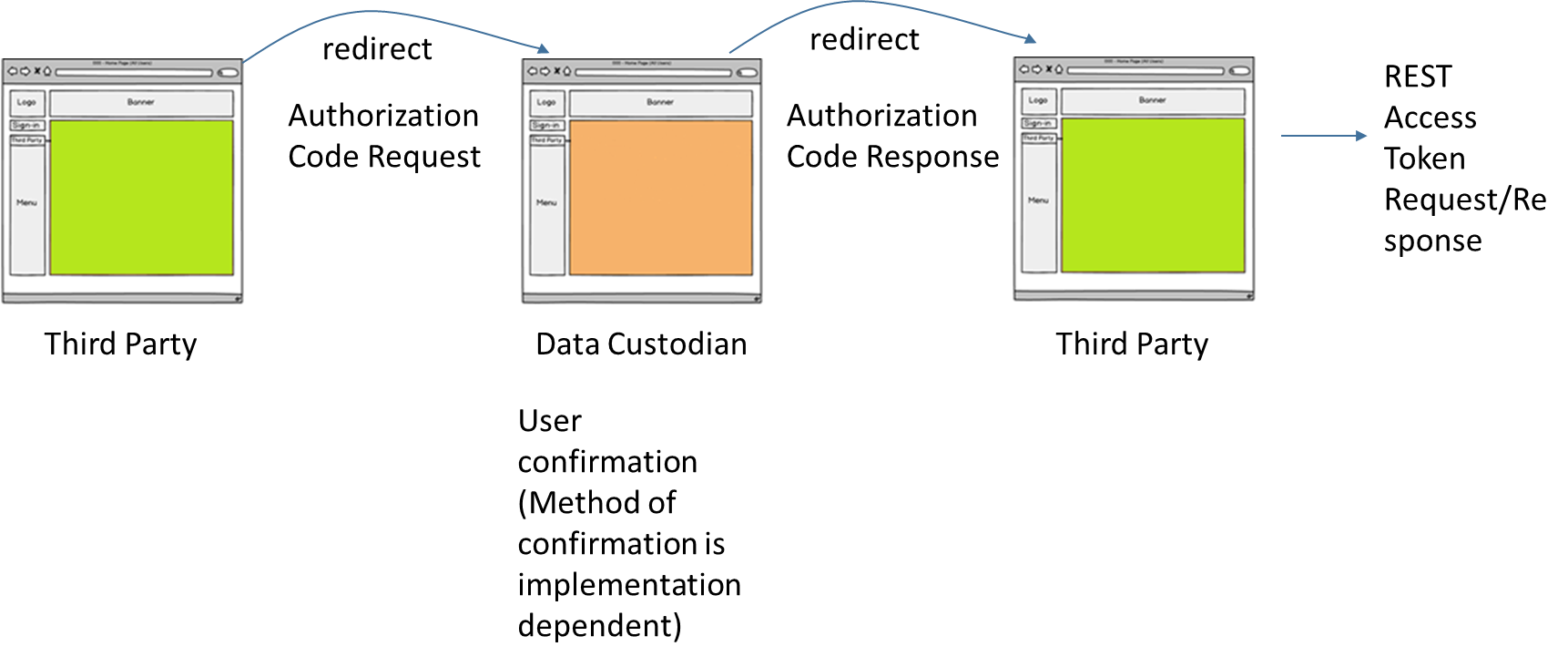


Figure : Authorization Storyboard

At this point, all required parameters needed to complete the Authorization process have been established – the Retail Customer, Data Custodian, Third Party and a Scope have been selected that is acceptable for all three. Now the authorization process can be completed.

1. The Third Party redirects the Retail Customer’s browser to the Data Custodian’s authorization endpoint -- {authorizationServerAuthorizationEndpoint}.
2. The Data Custodian performs the required Retail Customer OAuth 2.0 “authorization code” flow authentication and authorization process. The result of this process is a HTTP redirection message to a URL contained within the {redirect\_uri} query parameter of the HTTP redirection message. If the Retail Customer has authorized the Third Party access to the Data Custodian, the HTTP redirection message contains an OAuth 2.0 “authorization code” as a query parameter, otherwise the redirection contains an OAuth 2.0 Authorization request error response. The contents of the HTTP redirection message should NOT be displayed by the Retail Customer’s browser for security reasons.
3. Upon processing a HTTP redirection message containing an “authorization code” parameter the Third Party sends a request to the Data Custodian’s token endpoint -- {authorizationServerTokenEndpoint}. The Third Party then receives an OAuth 2.0 access token JSON response that includes access\_token, token\_type, expires\_in, refresh\_token, scope, authorizationUri, and resourceUri.

#### UML Sequence Diagram

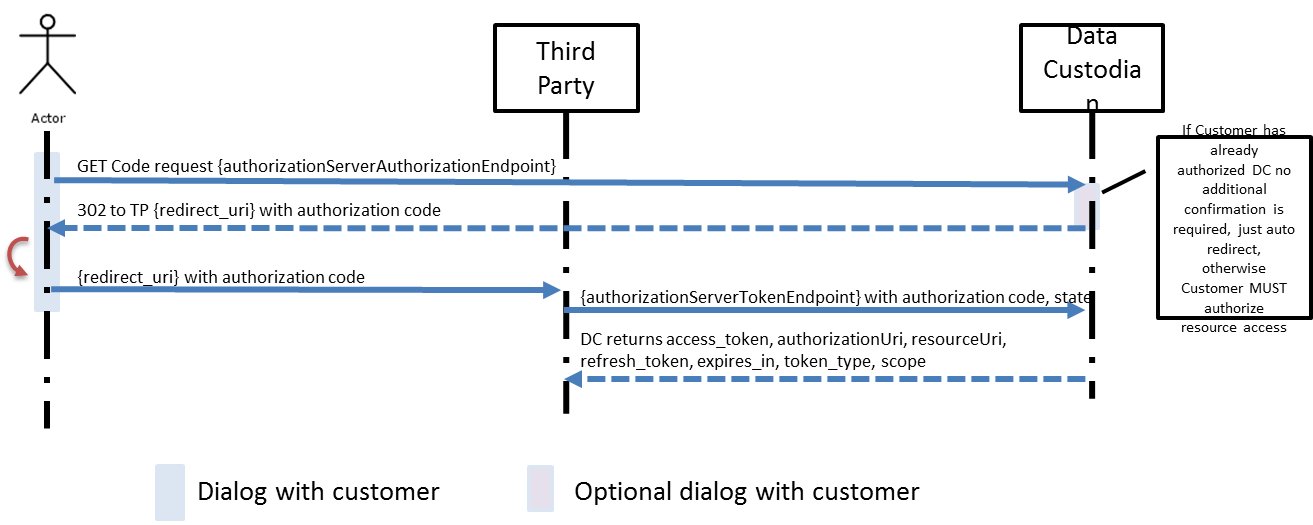


Figure : Authorization Flow

#### Message Exchanges

1. **(Green Button *ESPI Implementation Standard*)** Transition: To Customer Authorization

**Caused by: Retail Customer completes Third Party Scope Selection Screen**

POST {ThirdPartyScopeSelectionScreen} HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

DNT: 1

Referer: {ThirdParty}{ThirdPartyScopeSelectionScreen}

Cookie: JSESSIONID= F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: nnnn

scope={ScopeString}

**Return:** 302 Found (Redirect)

**Note**: The Green Button ESPI implementation standard **requires** the HTTP redirection message contain the optional OAuth 2.0 authorization request **redirect\_uri=**, **scope=** and **state=** query parameters.

HTTP/1.1 302 Found

Server: Apache-Coyote/1.1

Location: {AuthorizationServerAuthorizationEndpoint}

?response\_type=code

&client\_id={ThirdPartyClientID}

&redirect\_uri={ThirdPartyOAuthCodeCallbackURL}

&scope={ScopeString}

&state={State}

GET {AuthorizationPath}

?response\_type=code

&client\_id={ThirdPartyClientID}

&redirect\_uri={ThirdPartyOAuthCodeCallbackURL}

&scope={ScopeString}

&state={State}

Host: {AuthorizationServer}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: {ThirdParty}{ThirdPartyScopeSelectionScreenURL}

Cookie: JSESSIONID=F9128C4A30686004CDAFED270C9A667F; Secure

Connection: keep-alive

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Pragma: no-cache

Expires: Thu, 01 Jan 1970 00:00:00 GMT

Cache-Control: no-cache

Cache-Control: no-store

Content-Type: text/html;charset=ISO-8859-1

Content-Language: en-US

Content-Length: nnnn

Line-based text data: text/html (DataCustodianAuthorizationScreen)

1. Intermediate messaging

The Data Custodian may use any method it desires to perform the required OAuth 2.0 Retail Customer authentication and authorization process. These interactions are not part of the Green Button ESPI implementation standard. Upon completion of the authentication and authorization process, the Data Custodian sends an HTTP redirection message to the URL value contained in the redirect\_uri= parameter of the authorization request message and appends either an **error=** or **code=** query parameter based on the results of the authentication and authorization process.

**Return:** 302 Found (Redirect)

**Note:** The Green Button ESPI implementation requires the authorization request contain the **state=** parameter, the value of which MUST be returned in the HTTP redirection message.

HTTP/1.1 302 Found

Server: Apache-Coyote/1.1

Pragma: no-cache

Expires: Thu, Jan 1 1970 00:00:00 GMT

Cache-Control: no-cache

Cache-Control: no-store

Location: {ThirdPartyOAuthCodeCallbackURL}

?code={AuthorizationCode}

&state={State}

Content-Language: en-US

Content-Length: 0

GET {ThirdPartyOAuthCodeCallbackURL}

?code={AuthorizationCode}

&state={State} HTTP/1.1

Host: {ThirdParty}

Accept: text/html, application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: {AuthorizationServerAuthorizationEndpoint}

?client\_id={ThirdPartyClientID}

&redirect\_uri={ThirdPartyOAuthCodeCallbackURL}

&response\_type=code

&scope={ScopeString}

&state={State}

JSESSIONID=8C0590EFBA4003C2C720D6B8300123D0; secure;

Connection: keep-alive

POST {TokenPath} HTTP/1.1

Authorization: Basic {ThirdPartyAuthorizationCode}

Accept: application/json, application/x-www-form-urlencoded

Cache-Control: no-cache

User-Agent: Java/n.n.n\_nn

Host: {AuthorizationServer}

Connection: keep-alive

Content-Length: nnnn

Line-based text data: application/x-www-form-urlencoded

grant\_type=authorization\_code

&code={AuthorizationCode}

&redirect\_uri={ThirdPartyOAuthCodeCallbackURL}

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Cache-Control: no-store

Cache-Control: no-cache

Pragma: no-cache

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Date: Thu, 07 Mar 2013 02:41:09 GMT

{

“access\_token”:“{AccessToken}”

“token\_type”:“bearer”

“expires\_in”:{AccessTokenExpirationTime} (Recommended)

“refresh\_token”:{RefreshToken} (Recommended)

“scope”:“{ScopeString}”

“resourceURI”:“{ResourceURI}”

“authorizationURI”:“{AuthorizationURI}”

}

GET {ThirdPartyUserPortalScreen}

Host: {ThirdParty}

Accept: application/octet-stream, application/json, \*/\*

Cache-Control: no-cache

Pragma: no-cache

User-Agent: Java/n.n.n\_nn

Connection: keep-alive

HTTP/1.1 200 Found

Server: Apache-Coyote/1.1

Content-Type: application/xml

Content-Length: nnnn

Data: Thu, 07 Mar 2013 02:41:09 GMT

Line-based text data: text/html {ThirdPartyUserPortalScreen}

## Use Cases #3-12: Exchange of Resources

Once the relationships between ThirdParty and DataCustodian are established via Use Case #1, and, the individual resource authorizations made via Use Case #2, the balance of ESPI messaging consists principally of exchange of resources via HTTP requests and responses.

### REST HTTP Services

The following “CRUD” services for resource manipulation are achieved via HTTP services:

Table : CRUD Services for ESPI

|  |  |
| --- | --- |
| Abstract Service | HTTP Service |
| Create | POST |
| Read | GET |
| Update | PUT |
| Delete | DELETE |

The ESPI RESTful interface style uses HTTP messages to exchange data. The OAuth 2.0 profile of ESPI uses the bearer token form of authorization variable in the HTTP header to indicate the exchange is authorized. Depending on the exchange, different endpoints and bearer tokens are included. However, the form of the HTTP message and response is similar.

### Green Button Resources

REST describes the exchange of Resources. Resources are represented by URLs. Additionally, ESPI resources have UUIDs (as part of the Atom presentation format). The REQ.21 standard identifies the following as Resources:

* UsagePoint
* ReadingType
* IntervalBlock
* MeterReading
* ElectricPowerUsageSummary
* ElectricPowerQualitySummary
* LocalTimeParameters[[10]](#footnote-11)
* Subscription
* Authorization
* ApplicationInformation

The general form of a REST service invocation is as follows:

https://{hostname}{:portNumber}/espi/1\_1/resource/{resourceId}?{queryParameters}

where:

hostname the web site URL

portNumber an optional port number used to direct messaging to a particular service point

resourceId the RESTful “path” that identifies what the subject of the CRUD message is

queryParameters modifiers to the handling of the message

The form of the resourceId was discussed in the document “GreenButtonAtomLinks.docx”. There it was identified that the form of the resourceId is opaque as per the NAESB spec. However, an agreed upon structure provided hints as to the possible organization of such information.

The following query parameters are used to filter the resources returned by a feed. These use typical “?name=value[&…]” syntax. Only published-max/min and updated-max/min are required to be supported.

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

Rules for use of Query Parameters:

* Query parameters are optional for all Green Button ESPI standard API definitions
* The absence of query parameters supplied with Green Button ESPI standard API definitions means the Data Custodian MUST return all agreed upon available data as defined by the **scope=** parameter supplied by the Third Party during the Retail Customer Authorization process
* An HTTP status code of 202 is only valid for Green Button ESPI standard API …./Batch/… definitions and MUST be followed by a message to the Third Party notification endpoint when data is available (see section 2.4.7)
* The Data Custodian decides how much maximum data will be provided in response to a Green Button ESPI implementation API request
* A Third Party that receives less than the anticipated data in response to a Green Button ESPI implementation API request can ask for older data by using query parameters in subsequent API requests.
* Any Green Button ESPI implementation API response that contains an empty feed, indicates there is no data available. Note: If a gap in the data exists that matches the query parameters supplied in the API request, an empty feed will be returned to the Third Party. But there may be additional data. In order to avoid missing a gap when retrieving data based on a date range, always provide the data of the desired oldest data as either the published-min or updated-min query parameter.

### Error Responses

The following table summarizes the responses to resource exchange messages:

Table 6: HTTP Error Response

|  |  |
| --- | --- |
| Response Code | Description |
| 200 | The request has succeeded. |
| 202 | The request has been accepted for processing, but the processing has not been completed. A subsequent Notification will be sent when the data is ready along with the URIs to use to retrieve the data (see section 2.4.7). |
| 400 | The request could not be understood by the server due to malformed syntax. |
| 401 | The request requires user authentication. |
| 403 | The server understood the request, but is refusing to fulfill it. |

### Message Exchanges

Resource exchanges using the REST API interface have the simple pattern.

#### GET

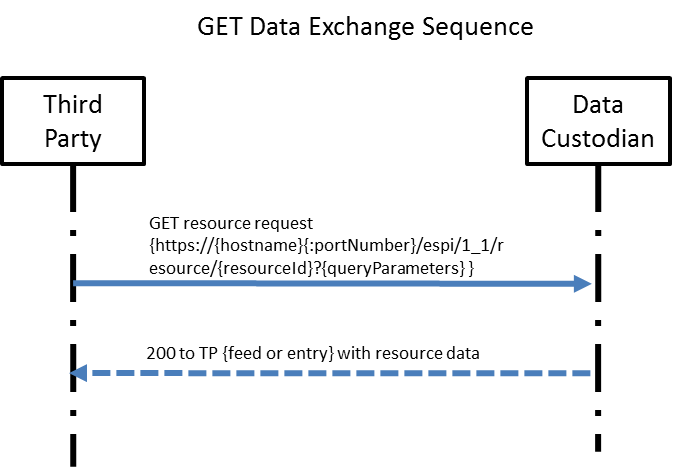


Figure : GET Exchanges with the DataCustodian

1. 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: openespivm:8443

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

1. HTTP GET Response

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: application/atom+xml

{ESPIXMLResource Representation}

Example:

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: application/atom+xml

Content-Length: 1227

Date: Sat, 11 Jul 2015 13:21:12 GMT

<entry xmlns:espi="http://naesb.org/espi" xmlns="http://www.w3.org/2005/Atom">

<id>urn:uuid:48c2a019-5598-4e16-b0f9-49e4ff27f5fb</id>

<link href="https://openespivm:8443/DataCustodian/espi/1\_1/resource/Subscription/1/UsagePoint" rel="up"/>

<link href="https://openespivm:8443/DataCustodian/espi/1\_1/resource/Subscription/1/UsagePoint/1" rel="self"/>

<link href="https://openespivm:8443/DataCustodian/espi/1\_1/resource/Subscription/1/UsagePoint/1/MeterReading" rel="related"/>

<link href="https://openespivm:8443/DataCustodian/espi/1\_1/resource/Subscription/1/UsagePoint/1/ElectricPowerQualitySummary" rel="related"/>

<link href="https://openespivm:8443/DataCustodian/espi/1\_1/resource/Subscription/1/UsagePoint/1/ElectricPowerUsageSummary" rel="related"/>

<link href="https://openespivm:8443/DataCustodian/espi/1\_1/resource/LocalTimeParameters/1" rel="related"/>

<title>Front Electric Meter</title>

<content>

<espi:UsagePoint>

<espi:ServiceCategory>

<espi:kind>0</espi:kind>

</espi:ServiceCategory>

</espi:UsagePoint>

</content>

<published>2012-10-24T00:00:00Z</published>

<updated>2012-10-24T00:00:00Z</updated>

</entry>

#### PUT or POST

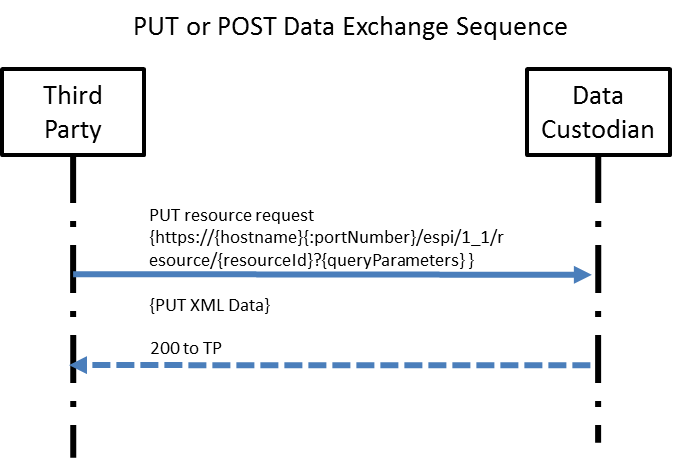
****

Figure : PUT or POST Exchanges with the DataCustodian

1. HTTP PUT Request

PUT {ResourcePath}<resource>/{ResourceID} HTTP/1.1

Host: {ResourceServer}

Content-Type: application/atom+xml

Authorization: Bearer {AccessToken}

Example:

PUT /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

Content-Length: 1273

Host: openespivm:8443

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<entry xmlns:espi="http://naesb.org/espi" xmlns ="http://www.w3.org/2005/Atom">

<id>urn:uuid:48C2A019-5598-4E16-B0F9-49E4FF27F5FB</id>

<link href="http://localhost:8080/DataCustodian/espi/1\_1/resource/RetailCustomer/1/UsagePoint/1" rel="self"/>

<link href="http://localhost:8080/DataCustodian/espi/1\_1/resource/RetailCustomer/1/UsagePoint" rel="up"/>

<link href="http://localhost:8080/DataCustodian/espi/1\_1/resource/RetailCustomer/1/UsagePoint/1/MeterReading" rel="related"/>

<link href="http://localhost:8080/DataCustodian/espi/1\_1/resource/RetailCustomer/1/UsagePoint/1/UsageSummary" rel="related"/>

<link href="http://localhost:8080/DataCustodian/espi/1\_1/resource/RetailCustomer/1/UsagePoint/1/ElectricPowerQualitySummary" rel="related"/>

<link href="http://localhost:8080/DataCustodian/espi/1\_1/resource/LocalTimeParameters/1" rel="related"/>

<title>Front Electric Meter</title>

<content>

<espi:UsagePoint>

<espi:ServiceCategory>

<espi:kind>0</espi:kind>

</espi:ServiceCategory>

</espi:UsagePoint>

</content>

<published>2012-10-24T00:00:00Z</published>

<updated>2012-10-24T00:00:00Z</updated>

</entry>

1. HTTP PUT Response

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

{ESPIXMLResource Representation}

Example:

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Length: 0

### Use Cases for Authorization Termination

TBD

### Use Case #10: Data Custodian Notifies Third Party of EUI Availability – Asynchronous

This Use Case provides a notification to the Third Party from the Data Custodian that Resources are ready for retrieval.

Upon receiving the BatchList of Resource URLs, the ThirdParty looks up the Authorization associated with each Resource URL and uses the access\_token in conjunction with the entire Resource URL from the BatchList to make the GET request to retrieve the data. Note that this BatchList may include URLs for Bulk Resources and/or individual ones.

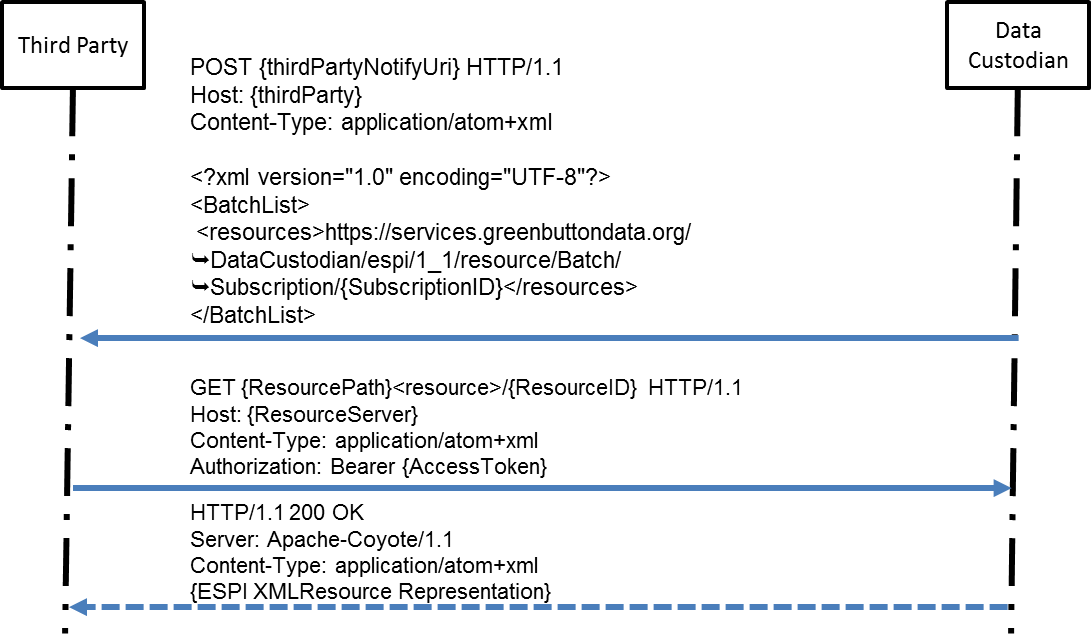


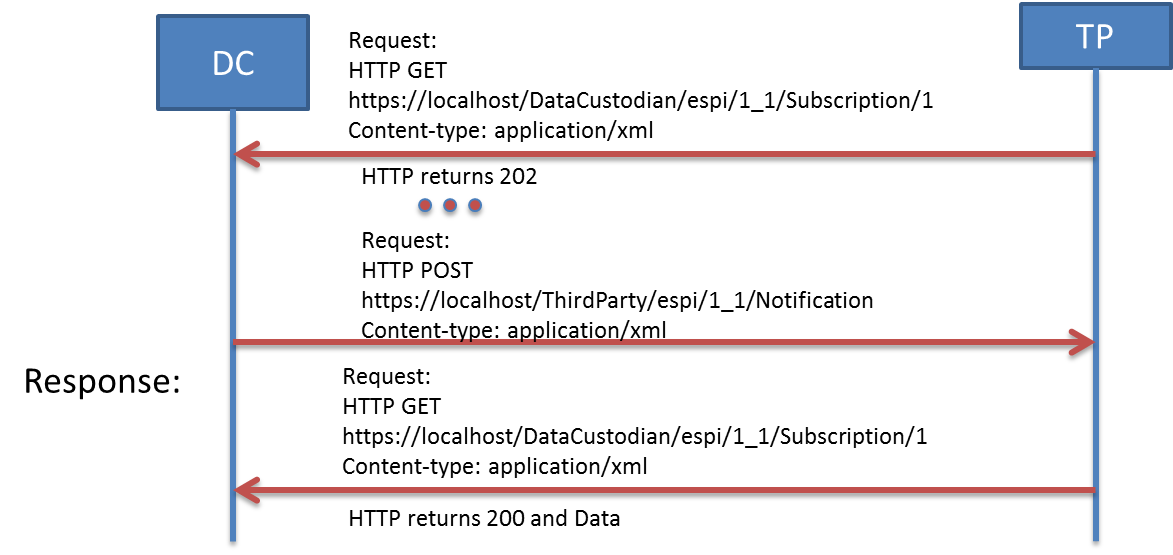
Figure : Notification

### Deferred Response when DC does not have data available

Depending on the structure of the Data Custodian, data requested may not be available. It is possible Subscriptions and Bulk Green Button ESPI implementation API requests may have large data responses that require time to assemble.

The HTTP response code 202 signifies to the Third Party a Data Custodian has accepted the request and will require some time to assemble the response. When the Data Custodian has completed assembling the data, it MUST issue a notification to the Third Party’s notification endpoint with the URI the Third Party will use to retrieve the assembled data. In many cases, this URI will be the Green Button ESPI implementation API the Third Party used in the initial request.

Since most Third Party requests seek immediate responses, the deferred response mechanism is valid only for Green Button ESPI implementation API …/Batch/… requests.



### Chunking – When the Data Custodian Needs to Provide Data in “Chunks”

For extremely large data sets, the Data Custodian may desire to have the Third Party retrieve data in defined parts called “chunks”.

In order to implement this mechanism, the Data Custodian utilizes the Green Button ESPI implementation Notification mechanism, along with query parameters to indicate to the Third Party how to retrieve results.

A message sent to the Third Party notification endpoint provides a XML formatted BatchList that contains URIs that are utilized by the Third Party in subsequent Green Button ESPI implementation API GET requests. To achieve chunking, the BatchList message contains fully distinguished URLs that constrain the amount of data sent in the subsequent responses by including query parameters that constrain the data being requested. Additionally, this provides the Third Party guidance on what to expect in the retrieved data.

For example, here is a BatchList that might be provided in a Notification message for the following request:

1. Third Party requests data

GET https://localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/1

1. Data Custodian responds with 202

202

1. Data Custodian sends Notification

<?xml version="1.0" encoding="UTF-8"?>

<BatchList xmlns="http://naesb.org/espi">

<resources>localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/1

⮩?published-min=2012-04-01T04:00:00Z

⮩&published-max=2012-04-02T04:00:00Z  
 </resources>

<resources>localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/1

⮩?published-min=2012-04-02T04:00:00Z

⮩&published-max=2012-04-03T04:00:00Z

</resources>

</BatchList>

1. Third Party requests chunks

GET https://localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/1  
 ⮩?published-min=2012-04-01T04:00:00Z

⮩&published-max=2012-04-02T04:00:00Z

GET https://localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/1  
 ⮩?published-min=2012-04-02T04:00:00Z

⮩&published-max=2012-04-03T04:00:00Z

A similar exchange pattern would result if the data were contained in files to be retrieved via SFTP with the notification URIs modified accordingly.

## Use Case #13: Bulk Transfer of Multiple Authorized Resources

Bulk transfer is for the exchange of large quantities of resources for which a Third Party holds valid authorizations with a given Data Custodian. During the “*Scope Negotiation Phase*” of the Authorization process (see section 2.3), a scope was established for each Authorization/Subscription. The **scope=** parameter of the OAuth 2.0 authorization request may optionally contain a **bulkId={bulkId}** value. The Data Custodian, at its discretion may ignore or override the value requested by the Third Party and provide it in the finalized scope provided.

The **bulkId=** parameter enables a Third Party to issue a “bulk” request for all resources that have been previously authorized with a matching **{bulkId}** as the **bulkId=** value contained in the OAuth 2.0 authorization **scope=** parameter by using the following Green Button ESPI implementation API:

GET /espi/1\_1/resource/Batch/Bulk/{bulkId}

The Data Custodian provides a response containing all resources the Third Party is authorized to access that have matching **{bulkId}** as the **bulkId=** value contained in the OAuth 2.0 authorization **scope=** parameter when the Retail Customer authorized the Third Party to access their information. The above API requires the Third Party to use the **client\_access\_token** as the value of the HTTP Authorization header field. It is the responsibility of the Data Custodian to ensure their response contains only resources covered by the collected individual authorizations that match the requested {bulkId} value.

Bulk transfer can be achieved through either a REST or SFTP transfer method. The {DataCustodianBulkRequestURI} field of the ApplicationInformation structure can indicate to the Third Party the preferred method the Data Custodian will use when responding to the above API interface.

When the Data Custodian sends a message to the Third Party notification endpoint, the retrieval mechanism they intend to use can be identified in the URI. The following are examples of how a Data Custodian might can indicate its intention to use a REST or SFTP transfer method:

REST example URI: https://localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/{bulkId}

SFTP example URI: sftp://user@localhost/DataCustodian/espi/1\_1/resource/Batch/Bulk/{bulkId}

Based on the URI provided, the Third Party client would use the appropriate messaging to retrieve the bulk data. Note that the value of the “user” in the sftp url is specified at the discretion of the Data Custodian originating it.

Bulk data is delivered as a single atom:feed that contains any number of entries. Each atom:entry can represent any authorized subscription the Third Party is authorized to receive. An atom:entry is received in no organized or sequenced order. The receiving Third Party is responsible for parsing the atom:feed into the corresponding User accounts based on the self links contained in the individual atom:entry.

## Encoding of OAuth 2.0 “scope”

The following defines a potential OAuth 2.0 “scope” parameter format. It is based on the Green Button Test Plan function blocks for the data custodian (defines what is in the file).

From the test plan (these are the individual blocks of conformance that will be tested for):

### Function Blocks Terms for Scope

| FB  Term | Function Block | FB  Term | Function Block |
| --- | --- | --- | --- |
| **1** | [FB\_1] Common Data Custodian  Required of all Data Custodian implementations. | **19** | [FB\_19] Partial update data  Support for partial updates of data consisting of only new IntervalBlock resources. |
| **2** | [FB\_2] Green Button Download My Data  Support for Retail Customer file download from Data Custodian web site. | **27** | [FB\_27] Usage Summary with Demands and Previous Day Attributes  Support for extra measurements in UsageSummary resource. |
| **3** | [FB\_3] Core Green Button Connect My Data  Support for the authorization and automated REST web services for Green Button data. | **28** | [FB\_28] Usage Summary Costs for Current Billing Period  Support for “current billing period” summary costs. |
| **4** | [FB\_4] Interval Metering  Support for Interval (time based) series of measurements. | **29** | [FB\_29] Temperature  Support for Temperature measurements readings. |
| **5** | [FB\_5] Interval Electricity Metering  Specific support for Wh interval readings. | **30** | [FB\_30] Common User Experience  Common user experience for DMD. |
| **6** | [FB\_6] Demand Electricity Metering  Support for W, VAR, and VA measurements | **32** | [FB\_32] Resource Level REST  Support for APIs that access by resource types. |
| **7** | [FB\_7] Net Metering  Support for NET metering measurements. | **33** | [FB\_33] Management REST Interfaces  Support for management REST interface that allows all APIs without further authorization. |
| **8** | [FB\_8] Forward and Reverse Metering  Support for separate forward and reverse channels in interval data. | **34** | [FB\_34] SFTP for Bulk  Support for Bulk data using SFTP to pull data from Data Custodian (single access of all resources authorized under single bulkId see section 2.4.7). |
| **9** | [FB\_9] Register Values  Support for “dial” reading. | **35** | [FB\_35] REST for Bulk  Support for Bulk data using REST GET to pull data from Data Custodian (single access of all resources authorized under single bulkId see section 2.4.7). |
| **10** | [FB\_10] Gas  Support for Gas Therm consumption readings. | **36** | [FB\_36] Third Party (Client) Dynamic Registration  Support for dynamic registration of Third Parties via API. |
| **11** | [FB\_11] Water  Support for water consumption readings. | **37** | [FB\_37] Query Parameters  Support for updated and published max and min date query parameters in REST requests. |
| **12** | [FB\_12] Cost of Interval Data  Interval data has cost-attributed to each interval. | **38** | [FB\_38] On Demand Requests  Support for On-Demand REST requests (without need for prior Notification). |
| **13** | [FB\_13] Security and Privacy classes  In CMD, required security and privacy based on section 2.7. | **39** | [FB\_39] PUSH model  Support for Data Custodian Notification of available resources followed by GET by Third Party (See section 2.4.6). |
| **14** | [FB\_14] Authorization and Authentication  Support for OAuth 2.0 and related requirements. | **40** | [FB\_40] Offline Authorization  Support for non-api based authorizations (no OAuth). |
| **15** | [FB\_15] Usage Summary  Support for the UsageSummary resource. | **41** | [FB\_41] Manage Authorization Resource  Ability to PUT and DELETE the Authorization resource to make updates to an authorization. |
| **16** | [FB\_16] Usage Summary with Cost  Support for cost elements of the UsageSummary resource. | **44** | [FB\_44] Manage ApplicationInformation Resource  Ability to PUT and DELETE the ApplicationInformation resource to make updates to the Third Party/ Data Custodian relationship. |
| **17** | [FB\_17] Power Quality Summary  Support for the PowerQualitySummary resource |  |  |
| **18** | [FB\_18] Multiple Usage Points  Support for multiple UsagePoint resources in a single file. |  |  |

### Extended Backus-Naur Form (EBNF) for Scope parameter:[[11]](#footnote-12)

| **Term** | **Expansion** |
| --- | --- |
| Scope | [ FBTerms ], [ ValueTerms ], [ ResourceTerms ]; |
| FBTerms | “FB=“, { [FBTerm], ”\_”} , FBTerm, ScopeDelimiter ; |
| FBTerm | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” | “10” | “11” | “12” | “13” | “14” | “15” | “16” | “17” | “18” | “19” | “27” | “28” | “29” | “32” | “33” | “34” | “35” | “36” | “37” | “38” | “39” | “40” | “41” | “44” |
| ValueTerms | { ( "IntervalDuration=", namedOrNumber,{“\_”, namedOrNumber}), |
|  | | ( "BlockDuration=", namedOrNumber,{“\_”, namedOrNumber}), |
|  | | ( "HistoryLength=", nonNegativeNumber), |
|  | | ( "SubscriptionFrequency=", nonNegativeNumber | namedFrequency), ScopeDelimiter }; |
| ResourceTerms | { (“AccountCollection=”, nonNegativeNumber) | “BR=”, brId), ScopeDelimiter} |
| ScopeDelimiter | “;” |
| namedFrequency | “billingPeriod” | “daily” | “monthly” | “seasonal” | “weekly” | |
| namedOrNumber | nonNegativeNumber | namedFrequency; |
| brID | Character, {Character}\*; |
| nonNegativeNumber | digit, { digit }; |
| Digit | 0 | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ; |
| Character | Digit | “-” | "A" | "B" | "C" | "D" | "E" | "F" | "G"  | "H" | "I" | "J" | "K" | "L" | "M" | "N"  | "O" | "P" | "Q" | "R" | "S" | "T" | "U"  | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" ; |

Where:

|  |  |
| --- | --- |
| ResourceTerms | If a Bulk resource is specified via the “BR” term, the value of the {bulkId} is provided after the equals sign (“=”). There could be one or more terms in this list that express the granularity of notifications about resource changes. If the Subscription has more than one UsagePoint, the AccountCollection term can indicate the number of UsagePoints included |
| FBTerms | The function blocks supported |
| ValueTerms | These are parameterized terms |
| IntervalDuration | This is the minimum default length of an interval in seconds (e.g. 900 for 15 minutes, 3600 for one hour, …) |
| BlockDuration | This is the length of a block that contains the intervals (based on enumeration of MacroPeriodKind in ESPI above as namedFrequency) |
| HistoryLength | This is the length of history buffer seconds |
| AccountCollection | Used where the DC wants to provide for the reporting of multiple UsagePoints in a single Subscription. The number of UsagePoints is represented by the value in the assignment statement – e.g. 4 UsagePoints would be AccountCollection=4. |

### Scope Examples for use In ApplicationInformation

The scope parameter in ApplicationInformation provides the scopes available from the DataCustodian. They are compound in that a particular scope for an authorization may be a subset.

Example 1:

Scope = “FB=1\_3\_4\_5\_8\_13\_18\_19\_31\_34\_35\_39;  
⮩IntervalDuration=900\_3600;BlockDuration=Daily; HistoryLength= 34128000;SubscriptionFrequency=Daily;  
⮩ AccountCollection=5;BR=1;”

Example 2:

Scope = “FB=1\_3\_4\_5\_7\_8\_13\_14\_15\_18\_19\_31\_32\_34\_35\_37\_38\_39\_40;  
⮩IntervalDuration=300\_900\_3600;BlockDuration=Daily\_BillingPeriod\_Weekly\_Monthly; HistoryLength=63072000;SubscriptionFrequency=Daily;  
⮩ AccountCollection=5;BR=1;”

Example 3:

Scope = “FB=1\_3\_4\_5\_8\_13\_14\_18\_19\_31\_34\_35\_39\_40;  
⮩IntervalDuration=300\_900\_3600;BlockDuration=Daily\_BillingPeriod\_Weekly\_Monthly; HistoryLength=94608000;SubscriptionFrequency=Daily;  
⮩ AccountCollection=5;BR=1;”

### Scope Examples for Individual Authorizations

Electricity Interval Metering of hourly load profile blocked monthly for 13 months and a usage summary:

Scope = “FB=1\_3\_4\_5\_13\_14\_15\_19\_37\_39;IntervalDuration=3600;BlockDuration=monthly;   
⮩ HistoryLength=94608000”

Monthly only electricity metering including summaries and costs for 13 months:

Scope = “FB=1\_3\_4\_5\_13\_14\_15\_16\_19\_37\_39;IntervalDuration=monthly;   
⮩ BlockDuration=monthly; HistoryLength=94608000”

## Security and Privacy for Green Button Connect My Data

### Overview

This section describes requirements for GBCMD that protect privacy and provide for secure messaging according to NIST best practices.

### Secure Messaging HTTP Headers

All messaging between Third Party and Data Custodian are via HTTPS and TLS. The following example contains suggested HTTP header contents:

The following example header shows secure headers with secure session identification that might occur in response to: {ResourceServer}{DataCustodianThirdPartySelectionScreenURL}

GET {DataCustodian[ThirdPartySelectionScreen](https://www.somedatacustodian.com/ThirdPartySelectionScreen)URL} HTTP/1.1

Host: {ResourceServer}

Accept: text/html, application/application/atom+xml

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

DNT: 1

Referer: {ResourceServer}{DataCustodianHomePageURL}

Cookie: JSESSIONID=BA353C960E0450DD2AE133469129CA49 ; secure;

Connection: keep-alive

### TLS

Transport Layer Security (TLS) is required for all exchanges for Green Button Connect My Data. NIST recommends that all implementations move to TLS version 1.2. However, most client web browsers are still at TLS 1.0 or at best TLS 1.1.

The following levels of TLS are required for the GBCMD Actors:

Data Custodian – Must implement TLS 1.2

Third Party – Must Implement TLS 1.1 or higher

Retail Customer – Can Implement TLS 1.0,1.1, or 1.2

Each party must choose the highest level of TLS mutually supported during the TLS negotiation.

The acceptable Cipher Suites are given in SP 800-57 part 3: [1]

Other than the first one (TLS\_RSA\_WITH\_NULL\_SHA, which is not acceptable because it does not offer confidentiality) they all are all good. The minimum Cipher Suite required for all implementations is TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA.

|  |  |  |  |
| --- | --- | --- | --- |
| SUITE | Key Exchange | Encryption | TLS |
| ~~TLS\_RSA\_WITH\_NULL\_SHA~~ | ~~RSA~~ | ~~NULL~~ | ~~1.0, 1.1, 1.2~~ |
| TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA \*\* | RSA | 3DES\_EDE\_CBC | 1.0, 1.1,  1.2 |
| TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA\* | RSA | AES\_128\_CBC | 1.0, 1.1, 1.2 |
| TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 | RSA | AES\_128\_CBC | 1.2 |
| TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA | RSA | AES\_256\_CBC | 1.0, 1.1, 1.2 |
| TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256 | RSA | AES\_256\_CBC | 1.2 |
| TLS\_DH\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA | DH\_DSS | 3DES\_EDE\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DH\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA | DH\_RSA | 3DES\_EDE\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA \*\*\* | DHE\_DSS | 3DES\_EDE\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA | DHE\_RSA | 3DES\_EDE\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DH\_DSS\_WITH\_AES\_128\_CBC\_SHA | DH\_DSS | AES\_128\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DH\_RSA\_WITH\_AES\_128\_CBC\_SHA | DH\_RSA | AES\_128\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA | DHE\_DSS | AES\_128\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA | DHE\_RSA | AES\_128\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DH\_DSS\_WITH\_AES\_256\_CBC\_SHA | DH\_DSS | AES\_256\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DH\_RSA\_WITH\_AES\_256\_CBC\_SHA | DH\_RSA | AES\_256\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA | DHE\_DSS | AES\_256\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA | DHE\_RSA | AES\_256\_CBC | 1.0, 1.1, 1.2 |
| TLS\_DH\_DSS\_WITH\_AES\_128\_CBC\_SHA256 | DH\_DSS | AES\_128\_CBC | 1.2 |
| TLS\_DH\_RSA\_WITH\_AES\_128\_CBC\_SHA256 | DH\_RSA | AES\_128\_CBC | 1.2 |
| TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA256 | DHE\_DSS | AES\_128\_CBC | 1.2 |
| TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 | DHE\_RSA | AES\_128\_CBC | 1.2 |
| TLS\_DH\_DSS\_WITH\_AES\_256\_CBC\_SHA256 | DH\_DSS | AES\_256\_CBC | 1.2 |
| TLS\_DH\_RSA\_WITH\_AES\_256\_CBC\_SHA256 | DH\_RSA | AES\_256\_CBC | 1.2 |
| TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA256 | DHE\_DSS | AES\_256\_CBC | 1.2 |
| TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 | DHE\_RSA | AES\_256\_CBC | 1.2 |

\* This cipher suite is named by IETF as mandatory-to-implement for TLS version 1.2 [RFC 5246].  
\*\* This cipher suite is named by IETF as mandatory-to-implement for TLS version 1.1 [RFC 4346]   
\*\*\* This cipher suite is named by IETF as mandatory-to-implement for TLS version 1.0 [RFC 2246].

### Certificates

Both the Data Custodian and the Third Party shall maintain unexpired, unrevoked RSA certificates with a public key length of at least 2048 bits. These certificates must be issued by a Certificate Authority (CA) that has been successfully audited according to the criteria of ETSI or WebTrust.

### Server Implementation

Data Custodian and Third Party implementations shall provide documentation to show that software libraries used are FIPS 140-2 level 1 or higher and listed on the CMVP website [4].

### Tokens and IDs

Tokens and IDs in this spec shall be opaque and if based on actual Customer information, must be randomized using a secure method to protect privacy. They shall consist of at least 48 bits and can be the random number part of an RFC2422 UUID [5].

## Behavior of GET UsageSummary with query parameters

# Appendix A: References

1. REQ.21 – Energy Services Provider Interface, NAESB 2010, <http://www.naesb.org/ESPI_Standards.asp>
2. SP 800-57 part 3: <http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_PART3_key-management_Dec2009.pdf>
3. The U.S. Federal PKI and the Federal Bridge Certification Authority, <http://www.idmanagement.gov/sites/default/files/documents/alterman-terena.pdf>
4. NIST Cryptographic Module Validation Program (CMVP) Website, <http://csrc.nist.gov/groups/STM/cmvp/validation.html>
5. RFC 2422, A Universally Unique IDentifier (UUID) URN Namespace, <http://www.ietf.org/rfc/rfc4122.txt>
6. The OAuth 2.0 Authorization Framework, RFC 6749, <http://www.ietf.org/rfc/rfc6749.txt>
7. The OAuth 2.0 Authorization Framework: Bearer Token Usage, RFC 6750, <http://www.ietf.org/rfc/rfc6750.txt>
8. “RESTful Web services: The basics”, IBM Developer Works, <https://www.ibm.com/developerworks/webservices/library/ws-restful/>

# 

# Appendix B: APIs

This section contains the Green Button Connect My Data REST API interfaces.

## API Type Behaviors

This section details the general behavior of the APIs based on API Type.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API Type | URI (beyond base resource – e.g. http://localhost:8080/DataCustodian) | Request Data | Response Data | Behavior |
| **Access to individual resources: ApplicationInformation, Authorization, ElectricPowerQualitySummary, UsageSummary, IntervalBlock,  MeterReading, ReadingType, LocalTimeParameters, UsagePoint** | **Note: {id} fields are locally unique to DataCustodian - retailCustomerId, usagePointId, subscriptionId, applicationInformationId, and authorizationId. They are obfuscated (at least to 48 bits) via algorithm local to DataCustodian so no account or other PII data can be deduced from the values.** |  |  |  |
| XPath GET Member | espi/1\_1/resource/RetailCustomer/{retailCustomerId}/…/{resource}/{resourceId} | none | entry containing resource in content member |  |
| XPath POST | espi/1\_1/resource/RetailCustomer/{retailCustomerId}/…/{resource} | entry containing resource in content member | 200 OK and entry containing resource in content member or ERROR | the POSTed resource becomes part of collection at the node of insertion. Ids can be linked to resource based on URI pattern. |
| XPath PUT | espi/1\_1/resource/RetailCustomer/{retailCustomerId}/…/{resource}/{resourceId} | entry containing resource in content member | 200 OK or error | the resource is updated to the new content. Note: UUID must be checked against the indicated URI. |
| XPath DELETE | espi/1\_1/resource/RetailCustomer/{retailCustomerId}/…/{resource}/{resourceId} | none | 200 OK or error | the resource is deleted |
| ROOT GET Member | espi/1\_1/resource/{resource}/{resourceId} | none | entry containing resource in content member |  |
| ROOT POST | espi/1\_1/resource/{resource} | entry containing resource in content member | 200 OK and entry containing resource in content member or ERROR | the POSTed resource becomes part of collection at the node of insertion. Note: Id associations can be made based on self link. |
| ROOT PUT | espi/1\_1/resource//{resource}/{resourceId} | entry containing resource in content member | 200 OK or error | the resource is updated to the new content. Note: UUID must be checked against the indicated URI. |
| ROOT DELETE | espi/1\_1/resource//{resource}/{resourceId} | none | 200 OK or error | the resource is deleted |
|  |  |  |  |  |
| **Collection Resources** | **Note: {id} fields are locally unique to the Data Custodian** |  |  |  |
| ROOT resource GET Collection | espi/1\_1/resource/{resource} | none | feed containing entrys with resource in content member | collection members limited by the authorization by access-token accompanying the request |
| XPath resource GET Collection | espi/1\_1/resource/Subscription/{subscriptionId}/…/{resource} | none | feed containing entrys with resource in content member | collection members limited by the authorization by access-token accompanying the request |
| Subscription GET | espi/1\_1/resource/Batch/Subscription/{subscriptionId} | none | feed containing entrys with all resources in subscription and according to the corresponding Authorization | Note: The specification of content of a subscription is not standardized but is established by the Data Custodian during the scope negotiation phase of Authorization. |
| Bulk GET | espi/1\_1/resource/Batch/Bulk/{bulkId} | none | feed containing entrys with all resources in subscriptions for all Authorizations with a scope containing the {BR=bulkId} |  |
| Download GET of data | espi/1\_1/resource/Batch/RetailCustomer/{retailCustomerId} | none | feed containing entrys with all resources in a single Retail Customer | Implements Download My Data. |
| Upload POST of data | espi/1\_1/resource/Batch/RetailCustomer/{retailCustomerId} | feed containing UsagePoints and related resources for a single RetailCustomer | 200 OK or error | Populates DataCustodian with raw data and makes association between retailCustomerId and resources under it. |

## Access Tokens that Regulate Access to Green Button Connect My Data APIs

The following table defines the OAuth 2.0 access tokens used by the Green Button Connect My Data APIs. The “Token” column of Table 8: *Green Button Connect My Data API* defines which access token is required for each API:

Table : Access Tokens used in Green Button Connect My Data APIs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Access Token Name** | **Grant Type** | **Scope** | **Abbr.** | **Description** |
| access\_token | code | FB1\_3\_... (as per authorization) | A | An authorization-specific access token used to enable access to the resources shared between Retail Customer, Data Custodian, and Third Party. This is persisted in the Authorization resource structure. When accompanying an HTTP message request, evidences the sender’s authorization to make the exchange. For example, the scope may be: FB=1\_3\_4\_5\_8\_13\_18\_19\_31\_39 |
| client\_access\_token | client\_credentials | FB\_34\_35 | C | A specific access token used by a Third Party to access collections of individually authorized resources via “Bulk Transfer” and to access Authorization resources held by the ThirdParty with the corresponding DataCustodian. This is persisted in an Authorization resource structure. |
| upload\_access\_token | client\_credentials | FB\_45 | U | A specific access token used by an AMI management application to PUSH data to a DataCustodian through a backend interface. This token can only be used for resource /Batch/RetailCustomer/{RetailCustomerID}/UsagePoint. This is persisted in the Authorization resource structure. |
| data\_custodian\_access\_token | client\_credentials | FB\_3\_19\_32\_33\_34\_35\_37\_38\_41\_44\_45 | D | A specific access token used by a trusted back-end application of the DataCustodian to populate and otherwise maintain the repository of the DataCustodian. This token enables access to any API and is persisted in the Authorization resource structure |
| registration\_access\_token | registration | FB\_36\_40 | R | A specific access token obtained during Third Party registration with the Data Custodian to enable access to the ApplicationInformation resource. This is persisted in the ApplicationInformation and Authorization resource structure. |
| refresh\_token | refresh | match "A" | - | A specific access token used to obtain a new access token when the previous access token expires. This is persisted in the Authorization resource structure. |

## APIs

The following table contains the Green Button Connect My Data API interface definitions Data Custodians are required to support. The “Tokens” column contains an abbreviation as defined in Table 7 indicating what OAuth 2.0 access token the Third Party MUST provide to the Data Custodian in the HTTP Authorization header field:

Table : Green Button Connect My Data APIs

| **Function Blocks** | **CRUD** | **API URL** | **Tokens[[12]](#footnote-13)** |
| --- | --- | --- | --- |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation | D |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation | D |
| [FB\_3] Core Green Button Connect My Data | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation/{applicationInformationId} | DR |
| [FB\_41] Manage ApplicationInformation | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation/{applicationInformationId} | DR |
| [FB\_41] Manage ApplicationInformation | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation/{applicationInformationId} | DR |
| [FB\_3] Core Green Button Connect My Data | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization | DC |
| [FB\_3] Core Green Button Connect My Data | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization/{authorizationId} | DC |
| [FB\_44] Manage Authorization | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization/{authorizationId} | DC |
| [FB\_44] Manage Authorization | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Authorization/{authorizationId} | DC |
| [FB\_35] REST for Bulk | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Bulk/{bulkId} | DC |
| [FB\_99] Future Proposed Features | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Bulk/{bulkId} | D |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/RetailCustomer/{retailCustomerId}/UsagePoint | DU |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/RetailCustomer/{retailCustomerId}/UsagePoint | DU |
| [FB\_3] Core Green Button Connect My Data | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Subscription/{subscriptionId} | DA |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Subscription/{subscriptionId}/UsagePoint/{usagePointId} | DA |
| [FB\_99] Future Proposed Features | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Bulk | D |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/IntervalBlock | D |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/IntervalBlock | D |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/IntervalBlock/{intervalBlockId} | D |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/IntervalBlock/{intervalBlockId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/IntervalBlock/{intervalBlockId} | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/LocalTimeParameters | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/LocalTimeParameters | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/LocalTimeParameters/{localTimeParametersId} | DA |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/LocalTimeParameters/{localTimeParametersId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/LocalTimeParameters/{localTimeParametersId} | D |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/MeterReading | D |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/MeterReading | D |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/MeterReading/{meterReadingId} | D |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/MeterReading/{meterReadingId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/MeterReading/{meterReadingId} | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ReadingType | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ReadingType | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ReadingType/{readingTypeId} | DA |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ReadingType/{readingTypeId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ReadingType/{readingTypeId} | D |
| [FB\_3] Core Green Button Connect My Data | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ReadServiceStatus | DCU |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId} | DA |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId} | D |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}. | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerQualitySummary | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerQualitySummary | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerQualitySummary/{electricPowerQualitySummaryId} | DA |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerQualitySummary/{electricPowerQualitySummaryId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerQualitySummary/{electricPowerQualitySummaryId} | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerUsageSumary | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerUsageSumary | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerUsageSumary/{UsageSummaryId} | DA |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerUsageSumary/{UsageSummaryId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/ElectricPowerUsageSumary/{UsageSummaryId} | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId} | DA |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId} | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId}/IntervalBlock | DA |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId}/IntervalBlock | D |
| [FB\_32] Resource Level REST | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId}/IntervalBlock/{intervalBlockId} | DA |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId}/IntervalBlock/{intervalBlockId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Subscription/{subscriptionId}/UsagePoint/{usagePointId}/MeterReading/{meterReadingId}/IntervalBlock/{intervalBlockId} | D |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/UsagePoint/ | D |
| [FB\_33] Management REST Interfaces | POST | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/UsagePoint/ | D |
| [FB\_33] Management REST Interfaces | GET | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/UsagePoint/{usagePointId} | D |
| [FB\_33] Management REST Interfaces | PUT | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/UsagePoint/{UsagePointId} | D |
| [FB\_33] Management REST Interfaces | DEL | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/UsagePoint/{UsagePointId} | D |
| [FB\_3] Core Green Button Connect My Data | POST | https://services.greenbuttondata.org/ThirdParty/espi/1\_1/Notification |  |
| [FB\_42] Third Party Core REST Services | POST | https://services.greenbuttondata.org/ThirdParty/espi/1\_1/Notification |  |
| [FB\_34] SFTP for Bulk | GETALL | sftp://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Bulk/{bulkId} | DC |

# Appendix C: Function Blocks for Green Button Connect My Data

The table below summarizes the Function Blocks for Green Button Connect My Data. They form the basis of individual testing and certification requirements for Green Button Download My Data and Green Button Connect My Data. The “DMD” and “CMD” columns indicate which Function Blocks are required to obtain Download My Data and Connect My Data certification.

Table : Function Blocks for Green Button Download My Data and Green Button Connect My Data

|  |  |  |
| --- | --- | --- |
| **Function Block** | **DMD[[13]](#footnote-14)** | **CMD[[14]](#footnote-15)** |
| **Data Custodian Role** |  |  |
| [FB\_01] Common | M | M |
| [FB\_02] Download My Data | M |  |
| [FB\_03] Connect My Data |  | M |
| [FB\_04] Interval Metering | M | M |
| [FB\_05] Interval Electricity Metering | I | I |
| [FB\_06] Demand Electricity Metering | O | O |
| [FB\_07] Net Metering | O | O |
| [FB\_08] Forward and Reverse Metering | O | O |
| [FB\_09] Register Values | O | O |
| [FB\_10] Gas | I | I |
| [FB\_11] Water | I | I |
| [FB\_12] Cost of Interval Data | O | O |
| [FB\_13] Security and Privacy Classes | O | M |
| [FB\_14] Authorization and Authentication | O | M |
| [FB\_15] Usage Summary | O | O |
| [FB\_16] Usage Summary with Cost | O | O |
| [FB\_17] Power Quality Summary | O | O |
| [FB\_18] Multiple UsagePoints | O | O |
| [FB\_19] Partial Update Data | O | O |
| [FB\_27] Usage Summary with Demands and Previous Day Attributes | O | O |
| [FB\_28] Usage Summary Costs for Current Billing Period | O | O |
| [FB\_29] Temperature | I | I |
| [FB\_30] Common User Experience | O |  |
| [FB\_32] Resource Level REST |  | O |
| [FB\_33] Management REST Services |  | O |
| [FB\_34] SFTP for Bulk |  | O |
| [FB\_35] REST for Bulk |  | O |
| [FB\_36] Third Party Dynamic Registration |  | O |
| [FB\_37] Query Parameters |  | M |
| [FB\_38] On Demand Requests |  | O |
| [FB\_39] PUSH Model |  | M |
| [FB\_40] Offline Authorization |  | O |
| [FB\_41] Authorized PUT/DELETE ApplicationInformation Resource |  | O |
| [FB\_44] PUT/DELETE Authorization |  | O |
|  |  |  |
|  |  |  |
|  |  |  |
| **Third Party Role** |  |  |
| [FB\_20] Common |  | M |
| [FB\_21] Connect My Data |  | M |
| [FB\_22] Security and Privacy Classes |  | M |
| [FB\_23] Authorization and Authentication |  | M |
| [FB\_24] Request Bulk UsagePoints |  | O |
| [FB\_25] Request Partial Update Data |  | O |
| [FB\_26] Properly Respond to Various Bad or Missing Data |  | M |
| [FB\_42] Core REST Services |  | O |
| [FB\_43] Management REST Services |  | O |

# Appendix D: ApplicationInformation and Authorization Resource Details

## ApplicationInformation

The ApplicationInformation resource exists for each Data Custodian/Third Party registration. The Data Custodian is responsible for maintaining the correct references for this resource and must provide it on demand (including the parameters that are supplied by the Third Party). Elements that are required are indicated in the table below with an asterist “\*” after the element name.

| Supplied by | Element \*Required | Description | Sample Content |
| --- | --- | --- | --- |
| DC | dataCustodianId\* | Contains the identifier for the Data Custodian. | “Sandbox Data Custodian” |
| DC | dataCustodianApplicationStatus\* | A code indicating the current status of the application. This value is provided by Data Custodian, cannot be modified by Third Party.  1 = Review  2 = Production  3 = On Hold  4 = Revoked | 2 |
| TP | thirdPartyApplicationDescription | A description of the application. | Third Party Application Description is added as provided by each TP |
| TP | thirdPartyApplicationStatus | A code indicating the current status of the application.  1 = Development  2 = ReviewTest  3 = Production  4 = Retired | 1 |
| TP | thirdPartyApplicationType | A code indicating the type of the application.  1 = Web  2 = Desktop  3 = Mobile  4 = Device | 1 |
| TP | thirdPartyApplicationUse | A code indicating the expected use of the application.  1 = EnergyManagement  2 = Comparisons  3 = Government  4 = Academic  5 = Law Enforcement  6 = Gamification | 1 |
| TP | thirdPartyPhone | The phone number of the organization to which access will be granted. (For debugging - not to be shared with customers) | +1 800 673-6377 |
| DC | authorizationServerUri | Contains the base URI link to the authorization server. | https://services.greenbuttondata.org/DataCustodian |
| TP | thirdPartyNotifyUri\* | URI used to notify ThirdParty that subscribed information is available. | https://services.greenbuttondata.org/ThirdParty/espi/1\_1/Notification |
| DC | authorizationServerAuthorizationEndpoint\* | An OAuth 2.0 URI used by the client to obtain authorization from the resource owner via user-agent redirection. {AuthorizationServer}{AuthorizationPath} | https://services.greenbuttondata.org/DataCustodian/oauth/authorize |
| DC | authorizationServerRegistrationEndpoint | A URI used by the client to register a Third Party with a Data Custodian via its {AuthorizationServer}{AuthorizationServer}{RegistrationPath} | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/register |
| DC | authorizationServerTokenEndpoint\* | An OAuth 2.0 URL used by the client to exchange an authorization grant for an access token, typically with client authentication. | https://services.greenbuttondata.org/DataCustodian/oauth/token |
| DC | dataCustodianBulkRequestURI\* | {DataCustodianBulkRequestURI} URI of DataCustodian’s Bulk Request endpoint. The value is provided by the Data Custodian and cannot be modified by the ThirdParty. | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/Batch/Bulk/{bulkId} |
| DC | dataCustodianResourceEndpoint\* | {ResourceServer}{ResourcePath} | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource |
| TP | thirdPartyScopeSelectionScreenURI\* | URI redirected from DataCustodian to ThirdParty landing page. Often a login or Scope Selection page. | https://services.greenbuttondata.org/ThirdParty/espi/1\_1/RetailCustomer/ScopeSelection |
| TP | thirdPartyUserPortalScreenURI | URI of a Third Party’s web page for use with Green Button Connect My Data | https://services.greenbuttondata.org/ThirdParty/espi/1\_1/home |
| DC | client\_secret\* | A secret to be associated with a “client\_id” provided by the Data Custodian during registration. Used as part of the HTTP basic authorization needed for OAuth authorization. | secret |
| TP | logo\_uri | The link to the logo image for the application. Size greater than 180 x 150 may be cropped or reduced (OAuth logo\_uri). | http://services.greenbuttondata.org/ThirdParty/favicon.png |
| TP | client\_name\* | The name of the application to which access will be granted (OAuth client\_name). | Green Button ThirdParty |
| TP | client\_uri | The link to the main page of the application (OAuth client\_uri). | https://services.greenbuttondata.org/ThirdParty |
| TP | redirect\_uri\* | The default redirect back to the application after authorization grant (OAuth redirect uri). | https://services.greenbuttondata.org/ThirdParty/espi/1\_1/OAuthCallBack |
| DC | client\_id\* | Contains the identifier for the Third Party (OAuth client\_id). | (ThirdParty name) |
| TP | tos\_uri | A URI that points to a human-readable Terms of Service document for the Third Party Application. The Terms of Service usually describes a contractual relationship between the Retail Customer and the Third Party Application that the Retail Customer accepts when authorizing access to the Third Party Application. | http://services.greenbuttondata.org/ThirdParty/TermsOfService |
| TP | policy\_uri | A URI that points to a human-readable Policy document for the Third Party Application. The policy usually describes how a Retail Customer's energy usage information will be used by the Third Party Application. | http://services.greenbuttondata.org/ThirdParty/UsagePolicy |
| TP | software\_id\* | An identifier for the software that comprises the Third Party Application. The software\_id is asserted by the Third Party software and is intended to be shared between all copies of the Third Party software. The value of this field MAY be a UUID [RFC4122]. | MyCoolGreenButtonAnalyzer |
| TP | software\_version\* | A version identifier for the software that comprises a Third Party Application. The value of this field is a string that is intended to be compared using string equality matching. The value of the software\_version SHOULD change on any update to the Third Party software. | Version 1.00.00 |
| DC | client\_id\_issued\_at\* | Time date stamp at which this client\_id was issued. Note the schema data type is TimeType and the presentation in OAuth message flow is xs:dateTime and requires a conversion when accessed. | "1403190000" => 2014-06-19T15:00:00Z |
| DC | client\_secret\_expires\_at\* | Date time at which this client\_secret expires -- value of 0 means the client\_secret never expires. | 0 |
| TP | contacts | Array of email addresses for people responsible for the Authorized Third Party Application. These MAY be made available to Retail Customers for support requests for the Authorized Third Party application. The Data Custodian Authorization Server MAY use the email addresses as identifiers for an Authorized Third Party application administrative page. | support@energyos.org |
| TP | token\_endpoint\_auth\_method\* | (The authentication method used by the OAuth 2.0 Token Endpoint to authenticate the Third Party Application) | client\_secret\_basic |
| TP | scope\* | List of scope values the Third Party Application may use when requesting access Tokens. | FB=1\_3\_4\_5\_8\_13\_18\_19\_31\_34\_35\_39;IntervalDuration=900\_3600;BlockDuration=Daily; HistoryLength= 34128000;SubscriptionFrequency=Daily; AccountCollection=5;BR=1; |
| TP | grant\_types\* | Grant types Third Party plans to support (all three are required). | client\_credentials |
| TP | grant\_types\* |  | authorization\_code |
| TP | grant\_types\* |  | refresh\_token |
| TP | response\_types\* | Response types Third Party plans to supported. | code |
| DC | registration\_client\_uri\* | {ClientConfigurationURI} A URI used by a registered client to manage registration information. This URI is returned by the AuthorizationServer in the “registration\_client\_uri” field of the client information response. This is used to retrieve the ApplicationInformation resource | https://services.greenbuttondata.org/DataCustodian/espi/1\_1/resource/ApplicationInformation/{applicationInformationId}/ |
| DC | registration\_access\_token\* | A credential obtained during Third Party registration with the Data Custodian to enable access to the ApplicationInformation resource. This is persisted in the ApplicationInformation resource structure. | UUID is an automatically generated 128 bit number:  example = fe82518d-e325-404e-978c-c02f9339bccc  The Third party doesn’t need to know the value for this field. |
| DC | dataCustodianScopeSelectionScreenURI\* | The URI used by the Third Party to redirect the Retail Customer to the Data Custodian Scope Selection Screen (note that this will likely involve a dialog with the Retail Customer including a log in authentication process). | http://localhost:8080/DataCustodian/RetailCustomer/ScopeSelection |

## Authorization

The Authorization resource represents the current state of the authorization by a Retail Customer of a Third Party to retrieve a Subscription from the Data Custodian. Elements that are required are indicated in the table below with an asterist “\*” after the element name.

| Element \*Required | Description | Sample Content |
| --- | --- | --- |
| authorizedPeriod | Restricts access to requests or subscriptions within this date time interval. | <authorizedPeriod>  <duration>31536000</duration>  <start>1333252800</start>  </authorizedPeriod> |
| publishedPeriod | Restricts access to the objects within the associated resource that were published within this date time interval. | <publishedPeriod>  <duration>31536000</duration>  <start>1333252800</start>  </publishedPeriod> |
| status\* | The status of this authorization.  0 - Revoked  1 - Active  2 - Denied | 1 |
| expiresAt\* | Expiration period for the accessToken (seconds)  Note: OAuth returns expiration number of seconds. This must be converted to an absolute time at which it expires for this data structure. | 1333252800 |
| grant\_type | Type of grant being negotiated for. | authorization\_code |
| refresh\_token | Used to obtain a fresh access\_token. | AA886A7A-078D-4307-A3D9-AA036796DBC4 |
| scope\* | Negotiated scope of the authorization | FB=1\_3\_4\_5\_13\_14\_15\_19\_37\_39;IntervalDuration=3600;BlockDuration=monthly;HistoryLength=94608000 |
| token\_type\* | type of token used | bearer |
| error | contains last error returned to ThirdParty | server\_error |
| error\_description | contains free text string describing last error returned to ThirdParty | No service |
| error\_uri | specific error URI for last returned error | na |
| resourceURI\* | resourceURI that represents the data set authorized. Can be used in a GET of the resource subscription. | http://localhost:8080/DataCustodian/espi/1\_1/resource/Batch/Subscription/1 |
| authorizationURI\* | URI that can be used to update or delete this Authorization | http://localhost:8080/DataCustodian/espi/1\_1/resource/Authorization/1 |

1. The management application and upload application are not yet part of ESPI but are proposed extensions [↑](#footnote-ref-2)
2. The terms in braces are the https://{hostname}{:portNumber}for these servers [↑](#footnote-ref-3)
3. Acquires registration\_access\_token [↑](#footnote-ref-4)
4. Acquires client\_access\_token [↑](#footnote-ref-5)
5. Acquires authorization [↑](#footnote-ref-6)
6. Acquires access token [↑](#footnote-ref-7)
7. This use case is recommended to be deprecated. It is not detailed in the standard, and, the implementation of subscription is now arranged in the Scope negotiation for the authorization. [↑](#footnote-ref-8)
8. Note if the resource represents a single ESPI resource, a single <entry> can be returned. Otherwise, a sequence of entry is returned in an atom <feed> [↑](#footnote-ref-9)
9. This use case is recommended to be deprecated. It is satisfied through Use Case 10 notification followed by a GET from the ThirdParty so that it can be protected by OAuth. See section 2.4.6. [↑](#footnote-ref-10)
10. *Added as part of the UCAIug OpenADE Test Plan and best practices activity and proposed for inclusion in next ESPI revision.* [↑](#footnote-ref-11)
11. Note: The current is revised from earlier syntax with comma delimiters removed due to some unpredictable behavior in some OAuth code libraries that recognized comma and space as scope delimiters and ESPI wants the scope strings to be atomic. [↑](#footnote-ref-12)
12. “Tokens” Column abbreviation meanings: D -- data\_custodian\_access\_token, R -- registration\_access\_token, U -- upload\_access\_token,  
    C -- client\_access\_token, A -- access\_token. If the API supports multiple access tokens the column contains multiple abbreviations. [↑](#footnote-ref-13)
13. DMD Column Meanings: M – mandatory, O – optional, I – at least one of these FBs [↑](#footnote-ref-14)
14. CMD Column Meanings: M – mandatory, O – optional, I – at least one of these FBs [↑](#footnote-ref-15)