

MyData API Patterns: OAUTH for Green Button

Dr. Martin Burns, Electronic Engineer, Smart Grid and Cyber-Physical Systems Program Office, National Institute of Standards and Technology

Dr. David Wollman, Deputy Director, Smart Grid and Cyber-Physical Systems Program Office, National Institute of Standards and Technology

The My Data initiatives are part of the Obama Administration's efforts to empower Americans with secure access to their own personal data, and to increase citizens' access to private-sector data-based applications and services. With its focus on personal data, the My Data initiatives complement the administration's broader set of Open Data initiatives, which are making government data and other resources more open and accessible to innovators and the public. Together, these initiatives are creating a more comprehensive data ecosystem in which citizens can receive services based on analysis of their own personal data in the context of greater information and understanding of overall trends.

For example, in 2010, the Department of Veterans Affairs created Blue Button to give patients access to their own health records. The Blue Button ecosystem has grown to include many health providers and health plans, enabling more than 88 million Americans to have digital access to their own health records. Other efforts in the Departments of Energy, Commerce, Education and other Federal entities are [helping citizens access their own information](#) as fuel for more informed, effective decision making.

Another in the My Data initiatives is Green Button, a secure way to communicate energy usage information electronically using standardized RESTful API web services and a common data format.

This series of articles discusses [the Green Button initiative](#) and how it uses [OAuth 2.0](#) to perform third-party authorization and access. We cover:

- Defining Green Button (this article)
- Understanding Green Button's Technical Elements
- Using OAuth and Green Button
- Working with Green Button's Scope

Developers who are familiar with OAuth should expect the following terminology, used differently in Green Button circles but with easy equivalents to OAuth terms. The following summarizes these key terms:

OAuth 2.0 Terms	Green Button Terms
Resource Owner	Retail Customer
Resource Server	Data Custodian Resource Server
Authorization Server	Data Custodian Authorization Server
Client	Third Party

What is Green Button?

The vision for the Green Button initiative was first announced in September 2011, through the U.S. Chief Technology Officer's "call to action" to the electricity industry to create a "Green Button" to enable consumers to download their own energy usage information from their utilities' secure websites in a standardized electronic format.

The goal of the Green Button initiative is to increase consumers' access to their own energy usage information data, supported by an ecosystem of Green Button standards, testing and certification, developer support tools, applications and services. With voluntary adoption by many utilities nationwide, over 60 million U.S. customers (representing over 100 million citizens) and over 2.5 million Canadian customers (representing over 8 million Canadian citizens) now have Green Button data access.

Many organizations and stakeholders have supported the development of Green Button, including the White House Office of Science and Technology Policy (OSTP), the National Institute of Standards and Technology (NIST), the Department of Energy (DOE), and private sector organizations which include the UCA International Users Group (UCAIug), the Smart Grid Interoperability Panel (SGIP), the North American Energy Standards Board (NAESB), utilities, industry vendors, and application developers. With this broad support, the Green Button initiative has created a growing energy usage information ecosystem over the past few years, including the recent establishment of the Green Button Alliance (GBA), a private sector nonprofit trade organization dedicated to supporting Green Button. The GBA plays a similar role for Green Button as does that of the Wi-Fi alliance in establishing branding, testing, and certification for implementations of IEEE 802.11x for wireless Ethernet.

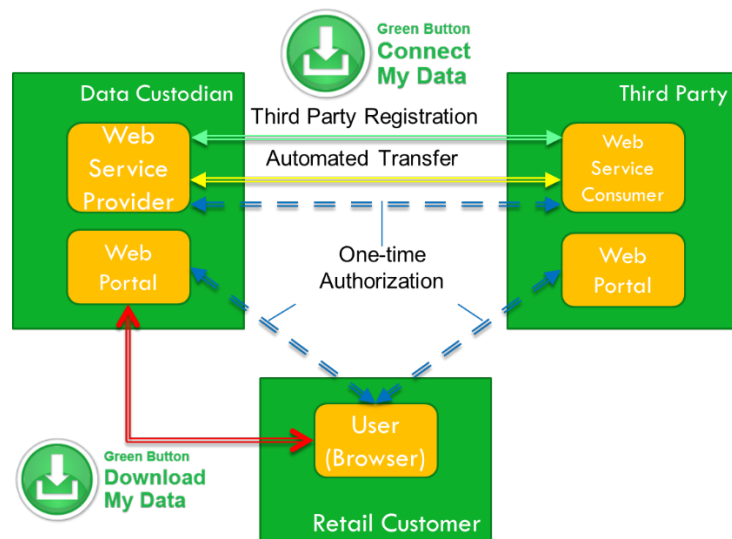
In the following articles, we describe the current state of Green Button technology, including contributions by NIST and its partners. Different stakeholders have contributed to this body of work throughout its development. Building on standards requirements contributed by the Open Automated Data Exchange Task Force (a working group under UCAIug), NIST and the SGIP initiated a priority action plan to accelerate standards development and implementation. NIST hosted an innovative information modeling "tiger team" which pulled contributions from leading standards efforts on energy usage information (EUI) to develop a key initial core UML modeling contribution. This in turn enabled subsequent development of the essential Green Button technical standards in NAESB, a standards development organization accredited by the American National Standards Institute (ANSI).

Plenty has been accomplished so far. The federal government has been a key driver of this work, working collaboratively with industry. NIST and DOE supported the U.S. CTO in key first adopter discussions and providing technical guidance. They achieved broad agreement on a standards-based XML electronic format that enabled California utilities to complete pilot implementations in less than four months. DOE supported an "Apps for Energy Challenge" focused on Green Button to kick start application development to help engage consumers. NIST and others, including Energy Open Source ([EnergyOS](#)), developed open-source/API reference implementations, testing tools, and technical materials, which support the creation of a comprehensive Green Button ecosystem that empowers consumers and entrepreneurs alike.

In creating Green Button technology, numerous innovative approaches have been developed, building on existing best practices for web services development, including Atom syndication format to provide metadata and OAuth 2.0 for third-party authorization. These advances enable Green Button to serve as a useful "pattern" for other data initiatives facing similar challenges. The novel extensions to OAuth that were derived from the requirements of the Green Button use cases are described in detail in this series.

The Green Button Download My Data and Connect My Data Use Cases

The Green Button Initiative provides for the exchange of Energy Usage Information to satisfy a number of use cases in the energy industry. The technology assembled addresses this diversity of use cases. The basic patterns of exchange are illustrated in the following figure:



Green Button technology defines three principal roles participating in a data exchange. (For Green Button, the data is energy usage information.)

Data Custodian: the entity that is the “custodian” of the data, for example the retail electric utility. OAuth refers to this role as the “resource server and authorization server.”

Third Party: the entity that wishes to provide a service to the retail customer and wants access to the data. For example, that might be a developer of advanced energy management services who is integrating cloud computing and handheld device access with consumer recommendations on how to lower a utility bill. OAuth refers to this role as the “client.”

Retail Customer: the party whom the data is about, such as a consumer who subscribed to a local electric utility for electric service. OAuth refers to this role as the “resource owner.”

What all Green Button exchanges share in common is a basic data format for exchange. In Green Button Download My Data (GBDMD) this data is exchanged via an interaction between the retail customer and data custodian on the latter’s web portal. The result is a file which is downloaded and can be used as the recipient sees fit.

Green Button Connect My Data (GBCMD), also known as Green Button Connect, involves the authorization and subsequent transfer of data via secure RESTful web services. In the next article, we cover the technical elements of Green Button, and the basics structures that developers need to understand in order to use it.

end of article 1

Understanding Green Button’s Technical Elements

This article series explains the role of [the Green Button initiative](#), a secure way to communicate energy usage information electronically using standardized RESTful API web services and a common data format. The previous

article in this series introduced the Green Button's goals and use cases. Here we continue with an examination of the architectural underpinnings.

The key requirements for the technology suite used to implement Green Button APIs include:

- **Do not reinvent the wheel.** Use existing standards whenever possible to handle things like security of data in transit, data schemas, token-based authentication, etc.
- **Support a diverse set of third party applications, from single user to complex enterprise-wide access.** For example, an enterprise use case might be a virtual audit, indicating opportunities for energy efficiency upgrades when a facility is compared to relevant baselines.
- **Allow for arbitrary complexity of data and data relationships.** Unlike “linear” technologies such as comma separated values (CSV) and electronic data interchange (EDI) files, complex data structures permit richer expression of detail and context of data. For example, relationships between measurements and the related tariffs that govern them enable “navigable” data availability beyond just the data in an initial request.
- **Provide for incremental exploration of available data rather than single batch access.** Rich data structures can be potentially large. By allowing data relationships, one data element can be acquired and understood to point to additional levels of detail. This allows a client to navigate to the data of interest rather than have to retrieve anything and everything and sift through what is wanted. This is especially beneficial for small devices that don't have or want to support large data buffers to sift through unneeded information.
- **Protect data privacy, allowing for anonymous data and aggregation.** Many applications of data about individuals do not need the individual identity to be useful. For example, a store kiosk might make recommendations based on anonymous data. Data fusion often involves “rolling up” contributing data into aggregated digests that present useful summaries.
- **Implement a high degree of security in machine-to-machine web services.** Modern communications must protect the confidentiality, integrity, and availability of data on demand.
- **Enable specification of which data (held by the data custodians) is available to third parties at a granular level.** Not every third-party service provider needs or should have access to all data for a retail customer. Enabling granular access provides maximal control of data by individuals and organizations.
- **Permit large populations of authorizations to have data retrieved by a third party in bulk.** Large third-party service providers may obtain the right to access data on large numbers of customers of a data custodian. Exclusively requiring individual access is inefficient and consumes both time and bandwidth.
- **Implement data identification and life cycle management.** Data is not static. Often it needs to be updated or enhanced with new information. Recognizing versions of the same data set and which is the “latest” is essential to managing its proper interpretation.

The Green Button technology is based on well-established existing standards that were assembled to meet the identified requirements. These requirements are presented in the subsequent sections that summarize them and elaborate their resolution. This series deals specifically with requirements for establishing third party authorization.

The following table summarizes the key standards used:

Standard	Purpose	Usage
IEC 61968-9 2nd Edition Application integration at electric utilities - System interfaces for distribution management - Part 9: Interfaces for meter reading and control	Information Model	Core information model for Green Button Energy Usage Information
NAESB REQ.21 Energy Services Provider Interface (ESPI)	Use Cases and APIs	Profile of the IEC standard used for Green Button and which elaborates the basic API use cases. (Note: the certified Green Button applications are based on extensions to the NAESB standard and not the standard alone.)
RFC 4287 Atom Syndication Format	Data Serialization	Syntactical representation of data in XML including information metadata
RFC 4122 A Universally Unique IDentifier (UUID) URN Namespace	Data Identification	Globally unique identifiers used to identify data transferred
RFC 6749 The OAuth 2.0 Authorization Framework	Third Party Authorization	Authorization of data to Third Party service provider
RFC 6750 The OAuth 2.0 Authorization Framework: Bearer Token Usage	Third Party Authorization	How to use bearer tokens to govern authorized messaging
RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2	Message Authentication	Secure messaging over TCP/IP
RFC 2818 HTTP Over TLS	Secure Messaging	Use of HTTPS for secure API messaging
RFC 3986 Uniform Resource Identifier (URI): Generic Syntax	REST Path Syntax	Format of REST URIs
espiderived.xsd Reference Schema	Schema for Green Button data	Used to describe all Green Button data structures

That's the standards upon which we built. In the next article, OAuth and Green Button, we describe its building blocks and how they respond to the project requirements.

end of article 2

OAuth and Green Button

One of the U.S. government's [My Data initiatives](#) is [Green Button](#), a secure way to communicate energy usage information electronically using standardized RESTful API web services and a common data format. In the earlier articles in this series, we defined its goals and described the technology standards on which it is built.

In this article, we describe the building blocks of Green Button technology with respect to authorization of access to data provided to third parties and show how we addressed its requirements, including how the relevant standards addressed Green Button generally along with how they were implemented for Green Button specifically.

Even if you aren't actually engaged in building or supporting software related to energy usage, most of these techniques are applicable to a broad number of data sets that might be exposed through web services. This is the case when:

- Periodic creation of data occurs, such as with sensor data streams in the Internet of Things (IoT)

- Service providers have large numbers of customers with a single data custodian
- A diversity of third-party services arise for different subsets of available customer data

We present the topics by describing the problem they solve, the specific issues for Green Button (such as like content), and finally what was used for Green Button.

So let's dive in, starting with OAuth 2.0.

OAuth 2.0 is designed for exactly the sort of use as Green Button, which involves authorization of third party access to retail customer resources held by a data custodian is orchestrated using OAuth 2.0. OAuth's key principle is that the data custodian and the third party should never exchange any private information about the retail customer. This is achieved in OAuth through the clever use of web browser redirection. The authorization process has the retail customer visit both the data custodian and third party websites, during the process of which the retail customer often is asked to authenticate himself. Non-personal Information about the authorization the third party and data custodian need to share are piggy-backed on the HTTP redirections as query parameters.

During our study of OAuth technologies, we found some minor challenges to its application among our stakeholder community. We believe these needs might be similar among some other stakeholders as well. They are specifically:

Custom response parameters. OAuth provides a limited set of details of the authorization established when providing the access-token. For example, OAuth provides the access-token itself but not a means to retrieve the authorization state or detect if it has changed.

In our case, the URI used in the data exchange phase is distinct to the authorization and can be used in subsequent requests using RESTful path navigation to details of the authorization. This URI contains a unique identifier that can be used to look up the corresponding access-token when the URI is provided in a notification (see the PUSH model discussed below).

In addition, we needed a method to allow the third party to retrieve the complete set of details about the authorization at any time using a second specific URI. This would include the overall duration of the authorization; OAuth provides the ability to convey the duration of the access-token, but not separately the life of the authorization, which can be many months. OAuth directly supports this extension mechanism. Additionally, when the retail customer seeks to change the parameters of an authorization – typically at the data custodian – the new status needs to be conveyed to the third party (see the discussion of Notification below).

Scope negotiation. In many OAuth applications, the scope of the negotiation is obvious or limited to a few options. In the case of Green Button there are many degrees of freedom in what is specifically authorized. In such a case, the chances are that an “asked for” scope is not acceptable or practical for a specific combination of data custodian and retail customer and third party.

We encountered a need for a scope negotiation protocol to ensure that only valid scopes were presented to and on behalf of consumers in the authorization grant service. Once the scope is agreed to, the standard OAuth sequence is used providing the appropriate scope. For example, consider a situation in which a gas-and-electric utility company's gas-only customer goes to a third-party site that analyzes electricity usage, and picks her utility to authorize, when this customer has no electricity service with that provider. It is important to discover this case programmatically in order to perform a graceful dialog with the customer.

Notification for PUSH model. *Curated data* is data that has been verified to some extent, as opposed to raw data. Green Button data available through utilities is curated and typically is acquired daily. As a result, data is ready

when it is ready. The natural model for providing such data to a Third Party is to use a “PUSH model.” However, OAuth only supports a “PULL” model from the third party. That is, the third party “GET”s resource data by providing an access token as evidence of the right to inquire. We added a notification service to accommodate this Green Button data distribution to provide secure distribution of resource URIs of data that is new to be retrieved. Once received, the third party uses the URIs and the proper access tokens to retrieve the data using the normal OAuth flow. Thus an effective “PUSH” pattern is achieved.

Bulk Transfer. Utility companies may have millions of customers, and third party service providers may serve a significant fraction this number. It would be extremely inefficient for a third party to make potentially millions of daily requests for the new data using OAuth 2.0 for regulating access to these resources. For example, that might require a utility to support millions of transactions per day, usually in a fairly narrow time window.

For this reason, a bulk transfer mechanism was devised to allow data retrieval to occur with a single daily request.

The relationship requirements among the retail customer, data custodian, and third party are not symmetrical. Typically, the data custodian has a responsibility to maintain the privacy and access to the retail customer’s data and therefore must strongly authenticate the customer’s requests to authorize. On the other hand, the third party may have a very short term or casual relationship to the retail customer, so does not need strong authentication. And, of course, in some cases the relationship may be long term and require substantial trust between the third party and retail customer.

Let’s consider a specific example. Imagine a kiosk in a big box store that markets a solar panel installation. A retail customer might want a cost estimate for such an installation. She might provide general information about her house’s kind and shape and perhaps its orientation. To provide the estimate, the kiosk may request access to the customer’s energy usage information at her utility company, since this is a key factor in determining the economic payback. In order to render this analysis, the kiosk needs the data, but it does not need to know who the customer actually is. By routing the customer to the utility that has the data (data custodian), she can authorize the data transfer. If the data itself is anonymous (it contains no Personally Identifiable Information (PII)) the kiosk (here it’s the third party) need never know the consumer’s identity to render the results of the cost analysis. Yet, the utility requires the retail customer’s clear authentication before it is willing to provide the data to the kiosk.

For another example: An energy services company may provide a virtual audit of a client’s commercial facility. Opportunities for great energy savings might be discovered after a review of previous usage in comparison with other data sets such as weather patterns. In this case, it is essential for the third party to understand the location and other PII for the property. Such a relationship might persist during the deployment of remedial solutions to track performance.

Custom Parameters

The OAuth authorization sequence provides the third party (which OAuth calls the “client”) with a minimum set of parameters used to administer the authorization (access_token, token_type, expires_in, refresh_token, scope). This information is extended in Green Button to provide critical additional parameters. OAuth provides the ability to return customized data when authorizations have been established. This capability has been used for Green Button to provide the following additional parameters:

resourceUri	The URI to retrieve the authorized data subscription. For example: https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Batch/Subscription/9B6C7065
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

authorizationUri A URI to retrieve the entire state of the authorization as a Green Button resource. This resource contains additional Green Button specific data including the actual period of the authorization. For example:
https://services.greenbuttondata.org/DataCustodian/espi/1_1/resource/Authorization/016a234f

Scope Negotiation

OAuth 2.0 typically is concerned with access to single or few different data at a time. Often the nature of the data custodian makes the *scope* (OAuth's term for what access is being authorized for) obvious, such as in providing access to one's photos stored in a cloud. This scope is used in the third party request for authorization, according to the protocol.

For Green Button, however, scope is more complicated, and goes beyond the simple case of access to time series of Energy Usage Information. Many variables and options can pertain for a specific retail customer / data custodian pair. For example, a customer may be an electricity customer, a gas customer, or both. The customer may have five minute, hourly, or monthly data available. Additionally, the third party may have data preferences. For example, although monthly data might be available for a customer, a third party service may be designed to only exploit fine-grained data that is stored hourly in resolution or better.

For this reason, Green Button has a fairly detailed scope description language that allows for the third party, data custodian, and retail customer to discover the compatible data service that works for all three. This "scope negotiation" is outside the OAuth protocol and occurs prior to the operation of the OAuth process. Its aim is to arrive at the acceptable scope string the third party can use to execute the authorization.

During scope negotiation, similar to the OAuth method of browser redirects, the retail customer visits both sites (the process can start at either data custodian or third party). During this process, the customer identifies himself to the respective parties, and those parties identify themselves to one other. The customer, properly authenticated at the data custodian stop, can identify the data available for him to share and select what to offer the third party. Note that both the data custodian or the retail customer may vary the availability of data based on any mutually-agreed basis, which may include what might be shared with a specific third party. The third party can determine if the data allows its service to be successfully provided (that is: "Do I have permission? Great!"), whereupon it can begin the OAuth authorization service.

The resulting scope or scopes are shared with the third party. These are shared as one or more query parameters, each of which identifies a scope the data custodian and retail customer is willing to share with the third party.

For example, let's assume third party that provides energy management services for a residential customer. This third party requires access to electricity and gas data. Optimally the data includes an hourly resolution for electricity and monthly resolution for gas in order for their software to provide the best strategy for energy efficiency and cost savings. Further, assume the customer has Municipal Gas and Electric (MG&E, a fictitious name) as his utility company. When the retail customer is sent to the data custodian, MG&E, from the third party website, the user is provided with the ability to authenticate himself as a MG&E customer and accept sharing with the third party. MG&E is willing to provide, for this specific customer for whom they actually provide gas and electric service, any of the following:

- Monthly gas usage data
- Hourly electric usage data

- Monthly electric usage data
- Hourly electric and monthly gas data

Once the customer sends the browser on its way back to the third party, these alternatives are conveyed. The third party recognizes that only one of these scopes will be optimal for its services and thus uses the “Hourly electric and monthly gas data” scope in the OAuth 2.0 authorization sequence. In the example described, the retail customer did not need to be burdened with details or dialogs to determine the appropriate scope. Yet, the third party and data custodian can negotiate behind the scenes (aka redirects) to find the suitable scope for this specific customer.

Before you work with Green Button, however, it’s important to understand how scope is used. That’s the subject of our next article in the series.

end of article 3 here

Working with Green Button’s Scope

OAuth uses the term scope to describe the depth or level of authorization permitted or required in a data-centric transaction. As applied to Green Button, an open data project to manage the interchange of information for energy data between utilities, third-party providers, and consumers, scope has specific needs. Start with the Green Button background [link here to the start of the series], and follow along as we define its underlying technologies.

In this article, we explain the structure of Green Buttons’ scope parameters and illustrate the data exchanges and protocol used to implement Green Button’s scope negotiation.

Note that there are two scenarios:

- **Retail customer starts at data custodian:** This may occur when the retail customer is looking for a service and asks the energy service provider what third party providers are available. For instance, the customer might ask the utility company to suggest a solar company who is equipped to service his house.
- **Retail customer starts at third party:** This may occur when the retail customer is pursuing a specific service provider. In this scenario, the customer and solar company might discuss the options available, and need access to information from the utility company in order to negotiate rates and cost savings.

The exchanges described here utilize the same web browser redirections used by OAuth 2.0. They share the express goal of identifying the third party and data custodian to one other and determining which available scopes the third party can use in the OAuth 2.0 authorization sequence after scope negotiation is completed.

Green Button SCOPE Syntax

As shown in the scope negotiations that follow, the data custodian is obligated to provide one or more scope strings to the third party. Usually, this is often only one option. However, it is possible to offer more than one from which a third party can choose.

OAuth defines the scope parameter as a string (see [OAuth RFC 6749](#), section 3.3 “space-delimited, case sensitive strings”).

The following tables define [how to encode a scope string using Extended Backus–Naur Form](#) (EBNF). Note that the function block terms (FBTerms) represent sets of behavioral and data requirements that allow for different provisions by data custodians. For example, function block 5 indicates support for electricity metering; function

block 10 indicates support for gas metering. These tables illustrate the potential diversity of Green Button data and how its availability can be conveyed to the third party. Note that there is no personally identifiable information (PII) in the scope string so it is inherently anonymous.

Here are two examples of a scope parameter for Green Button assembled with the syntax description that follows:

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"
```

If the data custodian and retail customer offer both strings to the third party, the redirect (see below) contains:

```
.../Scope = "FB=1_3_4_5_13_14_15_19_37_39;IntervalDuration=3600;BlockDuration=monthly;
↳ HistoryLength=94608000"
↳ &
↳ Scope = "FB=1_3_4_5_13_14_15_16_19_37_39;IntervalDuration=monthly;
↳ BlockDuration=monthly; HistoryLength=94608000"
```

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	The minimum default length of an interval in seconds (e.g. 900 for 15 minutes, 3600 for one hour, and so on)
BlockDuration	The length of a block that contains the intervals (based on enumeration of MacroPeriodKind in ESPI above as namedFrequency)
HistoryLength	The length of history buffer seconds
AccountCollection	Used where the data custodian 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; for example 4 UsagePoints would be AccountCollection=4.

The function block referenced above works with these values:

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 website.	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	32	[FB_32] Resource Level REST

	Support for W, VAR, and VA measurements		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).
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).
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.	39	[FB_39] PUSH model Support for data custodian notification of available resources followed by GET by third party
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.		
----	-------------------------------------------------------------------------------------------------	--	--

Retail Customer Starts at Data Custodian

Remember, there are two scenarios: The customer starts with the data custodian or the third party.

Let's walk through each process graphically, starting where the retail customer begins with the utility company. Or in formal terms, at the web portal of the data custodian. Note that in each scenario, what is required is that the retail customer browser and at both the data custodian and retail customer site. Depending on the needs of the services being provided, these steps may require retail customer interaction, or may be "click-through" without additional need for interaction.

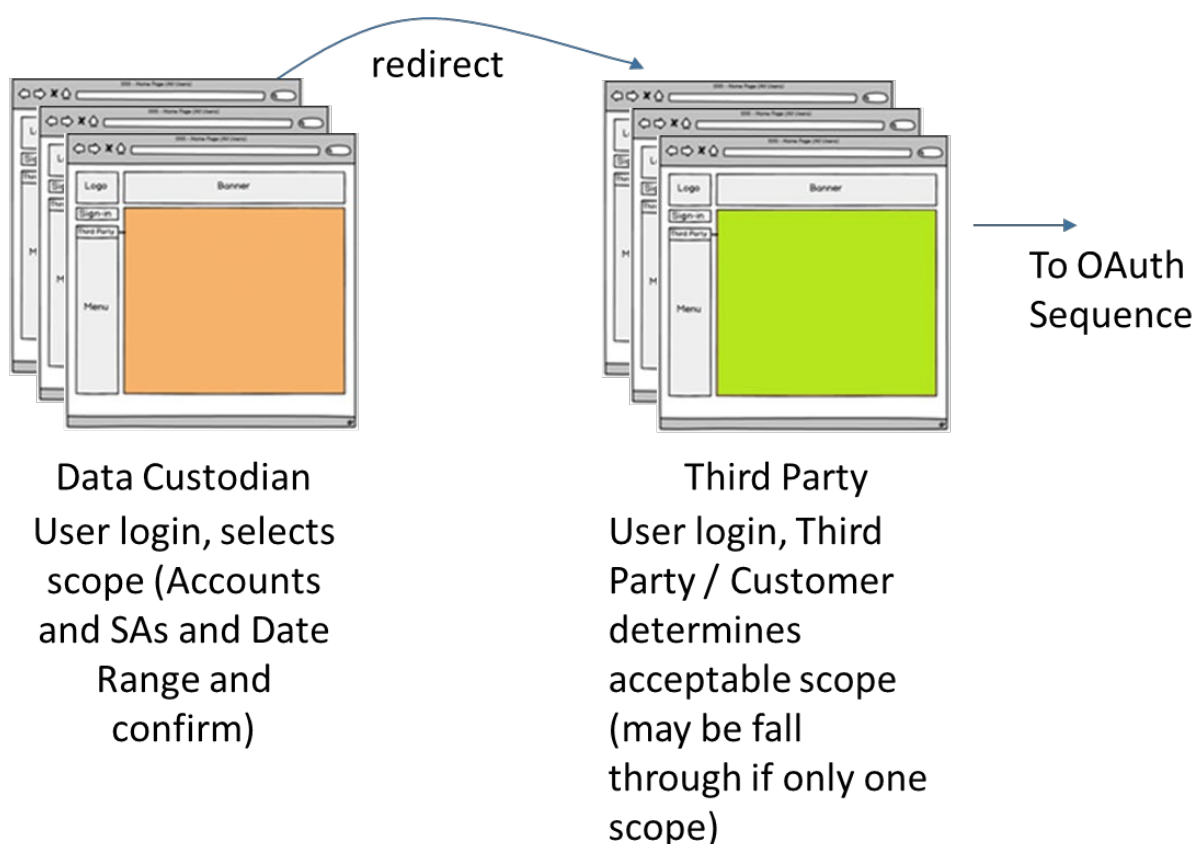


Figure 1: Data Custodian Point of View

The following steps represent "transitions" between the data custodian and third party application initiated by the data custodian's retail customer. The steps below enumerate the actions taken by the retail customer, starting with her initial browser selection to the start of the "OAuth 2.0 Authorization Phase" sequence:

- The retail customer selects the data custodian's website.
- The retail customer completes her login and completes all the data custodian's required documents.

For example, the data custodian may require the retail customer to navigate through its webpages to determine the availability of customer data the third party may access. These interactions are *not* a requirement of the Green Button implementation.

- At an appropriate step, determined by the data custodian, the customer is redirected to the selected third party. Contained within the HTTP redirection message are query parameters that identify the data custodian and the resources the retail customer granted to the third party during the “OAuth 2.0 Authorization Phase.”

(Scope={ScopeString}&[scope={ScopeString}&])

- At the third party site, the retail customer completes her login (as needed). Based on scopes contained within the data custodian’s redirect message, the third party provides the retail customer with a webpage or dialog, allowing her to select which data custodian resource scopes she wants to share. If there is only one acceptable choice, the third party may proceed to the next step without displaying those choices.
- After that selection is complete, the next step is a HTTP redirection message to the data custodian’s OAuth 2.0 authorization endpoint.

UML Sequence Diagram

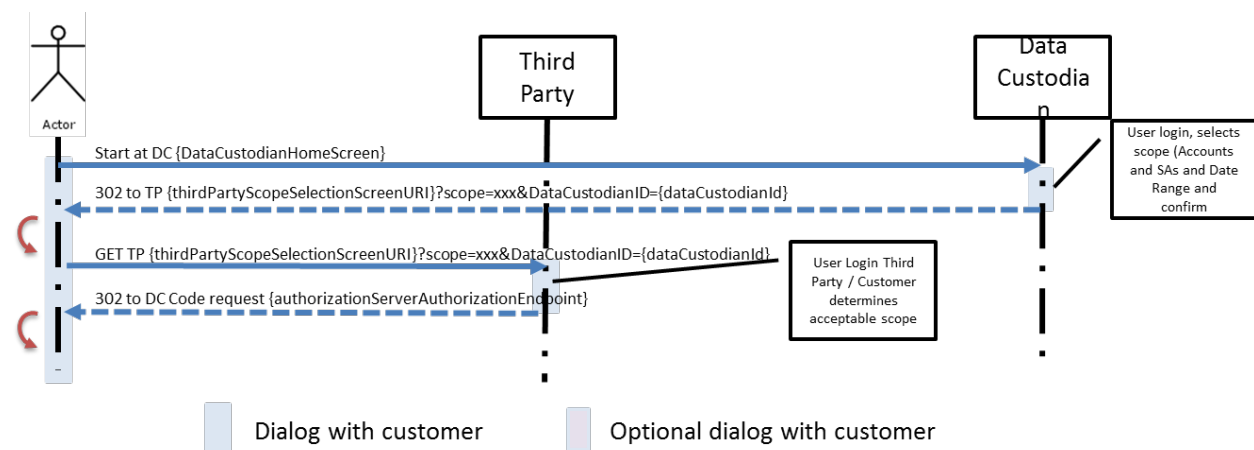


Figure 2: Data Custodian Point of View UML Model

Retail Customer Starts at Third Party

In the second scenario, the customer begins the process with the third party provider. Let’s step through each part of the process.

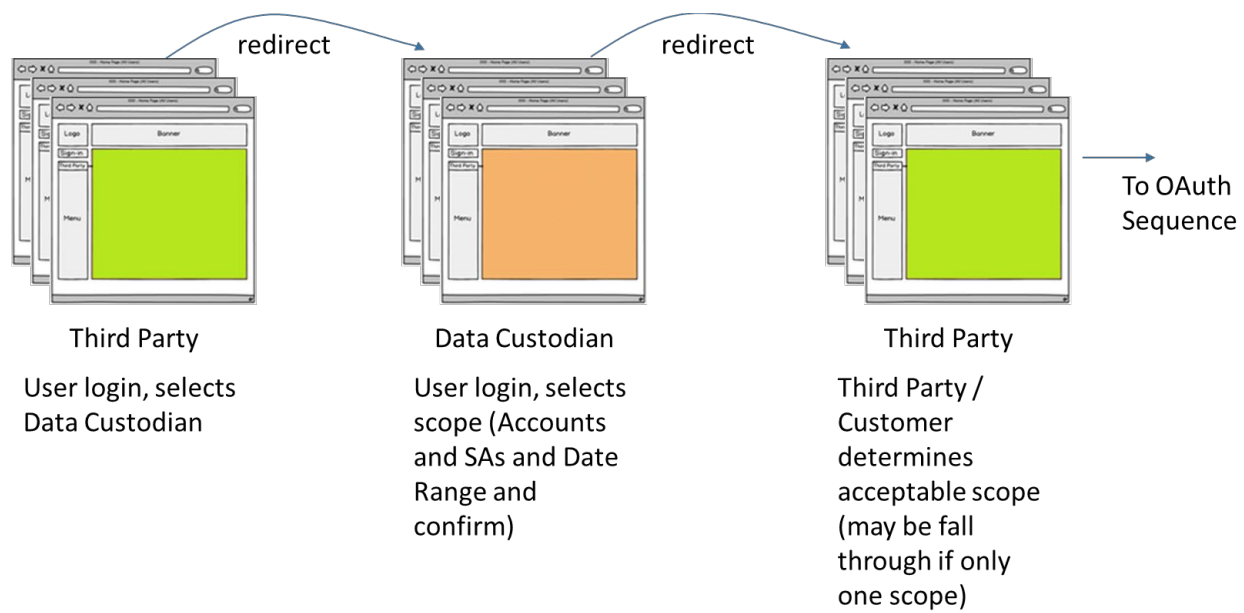


Figure 3: Third Party Point of View

The following steps represent “transitions” between the data custodian and the third party application, initiated by the third party’s retail customer.

Note that if the third party seeks to obtain authorization from multiple data custodians, it can repeat portions of this process to maximize the quality of the customer experience.

Here’s the actions taken by the retail customer, starting with his initial browser selection:

- Retail customer begins at the third party’s website.
- The retail customer completes his login and completes any documents required by the third party.

The third party may have the retail customer navigate through any number of webpages to meet its own needs. These interactions are *not* a requirement of the Green Button ESPI implementation.

- The third party presents the retail customer with a webpage or dialog box in which he selects which data custodian the third party is authorized to access. After he does so, the retail customer’s browser is redirected to the selected data custodian’s scope selection endpoint. Contained within the HTTP redirection message is a query parameter that identifies the third party to the data custodian.
- Once directed to the data custodian site by his chosen third party, the retail customer completes the login process, as well as fills in any documents the data custodian requires.

Again, that process may include as many webpages as the data custodian deems are necessary to determine the availability of customer data or permit access. These interactions are *not* a requirement of the Green Button ESPI implementation.

- Once that’s done, at an appropriate step the data custodian determines, the customer is redirected (back) to the selected third party. Contained within the HTTP redirection message are query parameters that identify the data custodian and the resources the third party may be granted access to by the Retail Customer during the “Oauth 2.0 Authorization Phase.”

(Scope={ScopeString}&[scope={ScopeString}&])

- Based on scopes contained within the data custodian's redirect message, the third party displays a webpage to the retail customer with a web screen, wherein he is asked to select which data custodian resource scopes he is willing to share. (If there is only one acceptable choice, that step may be skipped.)
- With all permissions granted, the next step is a HTTP redirection message to the data custodian's OAuth 2.0 authorization endpoint.

UML Sequence Diagram

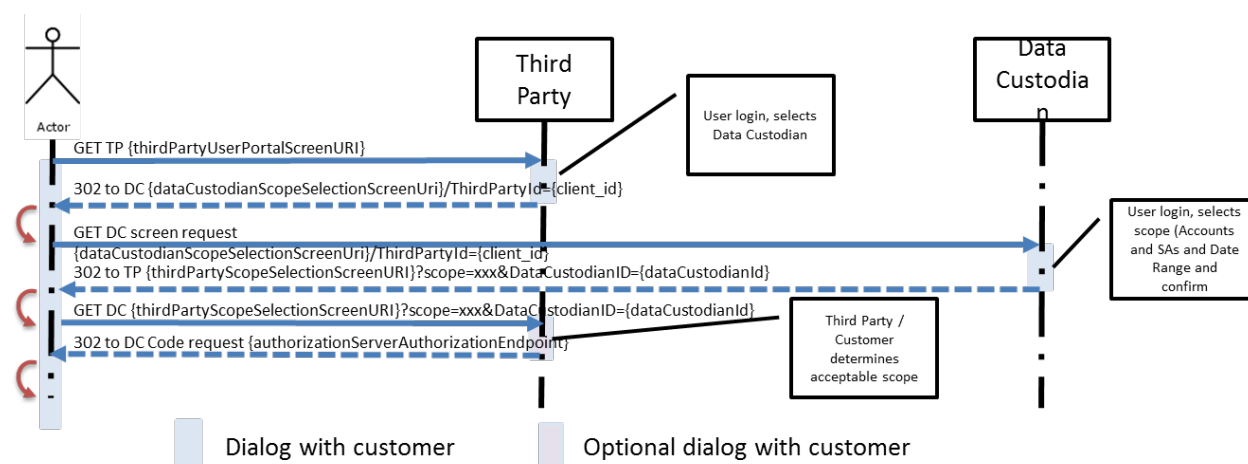


Figure 4: Retail Customer Starts at Third Party

PUSH Model

As described earlier, by its nature Green Button data is renewed regularly, typically daily or on a repeatable periodic basis. The new data is made available according to the data custodian's internal business processes. Thus it is not efficient for a third party to ask for the data whenever it desires it, since the data may not be ready. The obvious answer to consider, as in other such data transfer situations, likely is a PUSH model.

OAuth supports a non-symmetrical protection mechanism for access to authorized resources. The third party must GET resources and offer the access-token as evidence that it is authorized to make the request.

For these reasons, Green Button developed a "pseudo PUSH model." The model is consistent with the OAuth resource data exchange model, which is the mechanism by which all Green Button data is exchanged. Specifically, when the data custodian wants to "PUSH" data to the Third Party, it securely sends the third party a Notification POST message. The contents of the Notification is a *BatchList*, which is a sequence of resource URIs that changed and that the third party is recommended to GET via the normal mechanism. These resource URIs have no PII and are useless without a valid access_token to retrieve data.

This Notification can be used for various purposes:

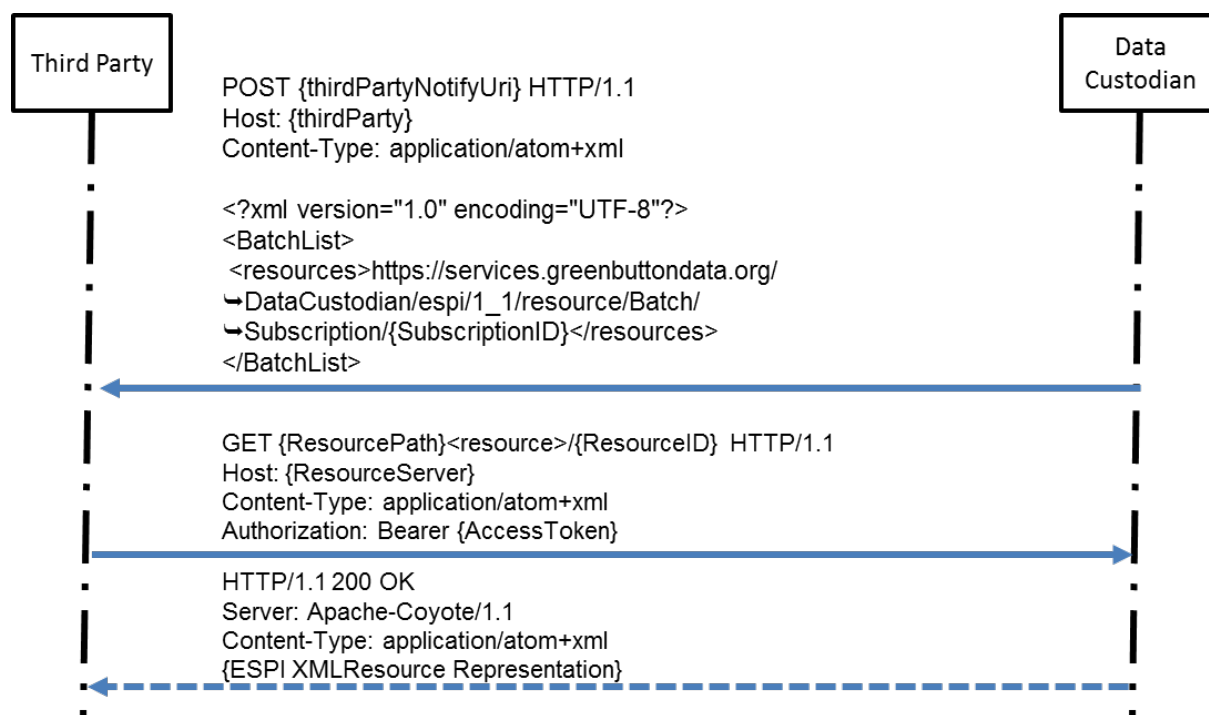
- To notify the third party of newly-available resources to retrieve
- To notify the third party of revised scopes and authorization status
- To notify the third party of revised parameters of the bi-lateral relationship between data custodian and third party

For example, URIs that a BatchList might include:

- the URI for bulk data retrieval indicating the bulk data is ready
- the URI for an individual subscription indicating that it has been updated
- the URI for an authorization indicating that its status or parameters have changed, such as when a customer terminates the authorization or changes its duration
- the URI for the relationship between the data custodian and the third party for a specific resource named "ApplicationInformation"
- the URI for an updated specific resource. For example, electricity meter readings are often provided daily as "raw" data and may be corrected at some later time.

Upon receiving the BatchList of resource URIs, the third party looks up the authorization associated with each Resource URL. It uses the access_token in conjunction with the entire resource URL from the BatchList to instruct the GET request to retrieve the data.

This BatchList may include URIs for bulk resources (see below) and/or individual resources.



Bulk Transfer

There's a lot of data involved, many energy utilities, a huge number of third party providers, and a lot of individual customers. Sometimes that can be daunting. For data to be retrieved daily, the Green Button community needs an efficiency mechanism to enable this transfer.

As a result, Green Button allows the data custodian and the third party to establish one or more “Bulk Transfer” URIs. This mechanism makes use of the Notification method described above. It notifies the third party that data is ready by providing a Bulk Transfer URI in the notification.

The third party can use this Bulk Transfer URI along with a client access token obtained during the third party registration process. The data custodian is responsible for maintaining the list of valid authorizations that may be included in the resulting data set returned by this request. As a result, with a single request, a third party can retrieve a data set of all new and changed resources for which authorization has been previously obtained.

Summary: Are You Energized?

This article series describes the Green Button Initiative as a set of technologies that enable utility customers to share their usage information with third party service providers using the OAuth 2.0 protocol. To satisfy the Green Button use case, several small but significant extensions to the OAuth standard were needed, as described herein. With these enhancements the “scope” of OAuth applications has been substantially increased to the benefit of the data services that may be provided.

We hope that similar use cases in other domains might benefit from these techniques as well.

The following references are provided for enriched additional information for the reader about technologies and choices made in the design of the Green Button architecture.

- [RESTful Service Best Practices](#)
- [URI Conventions \(OData Version 2.0\)/](#)
- [How to Build Green Button Applications](#)
- [API Sandbox](#)
- [UML OMG Unified Modeling Language](#) (OMG UML), Superstructure, V2.1.2, OMG, 2007-11-02
- IEC 61968-9 2nd Edition Application integration at electric utilities - System interfaces for distribution management: Interfaces for meter reading and control, [IEC 61968-9 2nd Edition](#)
- NAESB REQ.21 Energy Services Provider Interface (ESPI), [NAESB REQ.21](#)
- RFC 6750 The OAuth 2.0 Authorization Framework: [Bearer Token Usage](#)