

Testing Web Services

August 2016

Introduction

This document is designed to help you test exchanging data with EPA's Portfolio Manager via web services. Portfolio Manager is the industry standard for benchmarking energy performance, with more than 450,000 buildings benchmarked and new accounts added every day.

To learn more about Portfolio Manager and the associated web services visit: <http://portfoliomanager.energystar.gov/webservices>. Use this document to learn how you can begin testing our suite of web services to connect with Portfolio Manager.

Technical Tools and Requirements

Before you begin testing, you need to make sure you have the right tools. In particular:

1. Basic System and Software Requirements – Portfolio Manager uses Representational State Transfer (REST)-based web services. This allows you to choose any programming language that supports the HTTP protocol to develop your implementation.
2. Preparing XML Schema – Although it is not required, we strongly suggest that you use an XML editor to help author your XML requests. This will help you ensure that data is correctly formatted so that your calls will succeed.
3. Sending Data to Portfolio Manager – Web services are intended to be a software-to-software communication. Therefore, typically when you are testing or using web services your own custom software will send and receive your calls directly to EPA (i.e. there is no “user interface”). Because the primary use case is automated (not manual) transmission, EPA does not have its own custom REST client or submission window. However, when you are beginning testing, you may find it useful to submit individual XML files manually. If you are interested in manual submission, you can use one of any number of free REST clients available in the market such as the *RESTClient* for Mozilla Firefox (<https://addons.mozilla.org/en-US/firefox/addon/restclient/>) or *Advanced REST client Application* for Google Chrome (<https://chrome.google.com/webstore/detail/hqmlfoofddfdnphfgcellkdfbfjeloo>).

Testing Overview

The Test environment (<https://portfoliomanager.energystar.gov/wstest/>) is designed to help you test the web services only. EPA offers a test server with which you can exchange data via all of our web services, so that you can build the correct calls and logs into your software. The Test environment does not include a user interface as there is no test version of the Portfolio Manager graphical user interface. You can, however, use the live user interface with a Portfolio Manager account that is separate from your test account, simultaneously with your testing. This

means you can log in to see what a connection request looks like, and to see how EPA displays properties and property metrics.

The first step for any organization interested in exchanging data with Portfolio Manager will be to create a Test Account. The creation of this account is accomplished via an XML call, described below (see Testing Part A). Once your account is created, your next step will depend on which web services you need to test, based on your use case. There are two primary use cases:

1. Set up new properties and meters for your customers – This scenario will typically apply to third-party utility bill management and energy information services companies. These organizations aggregate and exchange all property and meter data into Portfolio Manager in order to supply their customers with the ENERGY STAR score and other metrics. If you are looking to perform this type of service, your next step will be to test creating customer accounts, properties, and meters. You can refer to Testing Part B, below.
2. Submit meter data only for existing properties – This scenario will typically apply to utilities, who will be authorized by a customer to submit utility data for their property. In this case, you will need to use the test environment to simulate what these requests will look like when they come in from your customers. To test, you will have to begin by creating test customer accounts and properties. You will use these test accounts to send your own account a connection request. This step is like your customer sending you a connection request in the live environment. After you accept the connection, you can also generate both property and meter share requests from your sample customers. These share requests are the way in which your customer will authorize you to manage data for a property or meter. You will have the opportunity to accept or reject each property or meter. Refer to Testing Part C, below, for more information on testing these share features.

Once you have completed your testing, you will be ready to move to the Live environment. To begin exchanging data with the live Portfolio Manager, you will need approval from EPA. Within your Account Settings in Portfolio Manager, there is an online form you will complete to request EPA approval to offer live web services.

When you are using testing, you can create as many accounts and properties as you need for testing. However, the purpose of testing is to help you build your software, not to populate your actual data. Therefore, when you are ready to move from the Test to the Live environment, please note that none of the data you have entered into Test is migrated into the Live Portfolio Manager.

Testing Part A: Creating an Account

Anyone wishing to test the system can create a test account following these simple steps using any REST client:

- 1) Select **POST** for the HTTP Method
- 2) Set the URL to **<https://portfoliomanager.energystar.gov/wstest/account>**

- 3) Add the following HTTP Header fields and values to the HTTP request:

Header Field Name	Value
Content-Type	application/xml

- 4) Set the request body to the following XML. The XML conforms to the account.xsd schema which can be referenced in the *XML Schemas* section at <https://portfoliomanager.energystar.gov/webservices/home/test/api/account/account/post>

Sample XML 1

```
<account>
  <username>acme dx user</username>
  <password>PA$sw0r6</password>
  <webserviceUser>true</webserviceUser>
  <searchable>true</searchable>
  <contact>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <address address1="123 Main St" city="Arlington"
      state="VA" postalCode="22201" country="US"/>
    <email>john_doe@acme.com</email>
    <jobTitle>Building Administrator Data Exchange User</jobTitle>
    <phone>703-555-2121</phone>
  </contact>
  <organization name="ACME Corporation">
    <primaryBusiness>Other</primaryBusiness>
    <otherBusinessDescription>other</otherBusinessDescription>
    <energyStarPartner>true</energyStarPartner>
    <energyStarPartnerType>Service and Product Providers</energyStarPartnerType>
  </organization>
</account>
```

- 5) Submit the request.
- 6) The XML response will indicate whether or not your account was created successfully.
- 7) Manage your account settings. For the purposes of Testing, EPA provides a test service for managing these account settings, including updating your terms and conditions, creating Custom fields, and specifying the fuel types that you support.

The following example shows how you would update your data exchange settings through a web service.

Sample XML 2

```
PUT /dataExchangeSettings

<dataExchangeSettings>
  <termsOfUse>
    <text>Terms and Conditions: You may use the ACME's services to
      interface with EPA's Data Exchange Platform only if you are a
      customer of ACME and agree to the detailed Terms and Conditions
      set forth on our company's website.</text>
  </termsOfUse>
  <supportedMeterTypes>
    <meterType>Electric</meterType>
    <meterType>Natural Gas</meterType>
    <meterType>Propane</meterType>
  </supportedMeterTypes>
</dataExchangeSettings>
```

```
</dataExchangeSettings>
```

You can also retrieve your data exchange settings to verify that your updates were successful.

```
GET /dataExchangeSettings
```

Sample XML 3

Testing Part B: Creating Customer Accounts, Properties, and Meters

If you plan to set up accounts, properties, and meters, for your customers this is the next important step for you, as these test services will be identical to what you will use in the Live environment.

If you are an organization that typically updates meter data on already existing properties, you will not need to use these services in the Live environment, but you will need to use them in the Test environment in order to simulate accounts and properties for your customers. That is, although you will not create accounts or properties in the live system, you need to create them in the Test environment to simulate what the pending connection and share requests will look like.

For either use case scenario, you will need to test the following web services:

1. **Create Customers** – There are a set of services that will enable you to create Portfolio Manager accounts on behalf of your customers. These services will include POST, GET, and PUT forms, which allow you to create the customer, retrieve their information, and edit their information, respectively. There is no ability to delete a Portfolio Manager account once it has been created.

To create a customer, you would use this call:

```
POST /customer
```

```
<account>
  <username>my_cust_user</username>
  <password>PA$sw0r6</password>
  <webserviceUser>true</webserviceUser>
  <searchable>true</searchable>
  <contact>
    <address country="US" postalCode="22201" state="VA" city="Arlington"
address1="123 Main St"/>
    <email>jane_doe@test.com</email>
    <firstName>Jane</firstName>
    <lastName>Doe</lastName>
    <jobTitle>Building Administrator</jobTitle>
  </contact>
  <phone>703-555-2121</phone>
  </account>
```

Sample XML 4

2. **Create Properties** – When you create a property you will be required to enter basic information, such as a property name and address. This information is submitted via a

POST call, may be edited via a PUT call, and may be deleted with a DELETE call. Finally, once property information has been uploaded to Portfolio Manager, it may be accessed via a GET call. Once you run a POST call, the response will indicate that the POST was successful, but there is no user interface to see the values. If you want to see the values, you can run a GET call to query the database.

To create a property, use this call:

Sample XML 5

```

POST /account/100/property

<property>
  <name>Broadway School</name>
  <constructionStatus>Existing</constructionStatus>
  <primaryFunction>K-12 School</primaryFunction>
  <grossFloorArea temporary="true" units="Square Feet">
    <value>10000</value>
  </grossFloorArea>
  <yearBuilt>2000</yearBuilt>
  <address postalCode="22201" address1="12321 Main Street" city="Arlington"
state="VA" country="US"/>
  <numberOfBuildings>5</numberOfBuildings>
  <isFederalProperty>false</isFederalProperty>
  <occupancyPercentage>55</occupancyPercentage>
</property>

```

3. **Create Meters** – Once you have created a property, you can create meters for that property, including water meters, energy meters, IT energy meters, and water/wastewater plant flow rate meters. Meter information is submitted via a POST call, may be edited via a PUT call, and may be deleted with a DELETE call. Finally, once meter information is exchanged into Portfolio Manager, it may be accessed via a GET call. Once you run a POST call, the response will indicate that the POST was successful, but there is no user interface to see the values. If you want to see the values, you can run a GET call to query the database.

To create an energy meter, use this call:

Sample XML 6

```

POST /property/200/meter

<meter>
  <type>Electric</type>
  <name>Electric Main Meter</name>
  <unitOfMeasure>kBtu (thousand Btu)</unitOfMeasure>
  <firstBillDate>2010-01-01</firstBillDate>
  <inUse>true</inUse>
</meter>

```

Associate Meters – Newly created meters are not automatically included in the energy or water use metrics for a property. In order for that to happen, a meter must be “associated” with a property. You can associate a meter to a property with a POST call. Note: The current POST meter association call is a “replace” function, not an “add” function, which means that it replaces the existing meter configuration for a property with a completely new meter configuration. If other meters have been previously associated to a property, they must be included in the POST call when you add the additional meter or the previous associations will be lost.

To associate the meters, use one of the following calls:

Sample XML 7 is a full meter association call: it associates a specified list of meters to a given property and overwrites the list of meters that were previously associated. This call treats energy, water, and waste meters as separate categories, so running this call for energy meters without including water or waste meters will not affect any standing water or waste meter associations.

Sample XML 7

```
POST /association/property/200/meter

<meterPropertyAssociationList>
  <energyMeterAssociation>
    <meters>
      <meterId>300</meterId>
    </meters>
    <propertyRepresentation>
      <propertyRepresentationType>Whole Property</propertyRepresentationType>
    </propertyRepresentation>
  </energyMeterAssociation>
</meterPropertyAssociationList>
```

Sample XML 8 is an individual meter association call: it associates just one meter to a property without affecting the other associated meters.

Sample XML 8

```
POST /association/property/100/meter/321
```

4. Add Meter Consumption Data – Once a meter is created, you can add use (or consumption) data to the meter. Consumption data typically includes a start date, an end date, a consumption value, and a unit. It is possible to submit a maximum of 120 periods (e.g. months) of consumption for a single meter within one XML call.

Meter consumption information is submitted via a POST call, may be edited via a PUT call, and may be deleted with a DELETE call. If you need to retrieve meter consumption data, you can use a GET call. Once you run a POST call, the response will indicate that the POST was successful, but there is no user interface to see the values. If you want to see the values, you can run a GET call to query the database.

To add meter consumption data, use this call:

Sample XML 9

```
POST /meter/300/consumptionData

<meterData>
  <meterConsumption>
    <usage>639573</usage>
    <startDate>2011-01-01</startDate>
    <endDate>2011-01-31</endDate>
  </meterConsumption>
  <meterConsumption>
    <usage>613123</usage>
```

```

        <startDate>2011-01-01</startDate>
        <endDate>2011-02-28</endDate>
    </meterConsumption>
</meterData>

```

5. Add Property Use Data – In addition to meter data, you may also manage property characteristics and use data for your customers. You would use these services to enter and update property usage information. You will use a POST call to enter data, a PUT call to edit data, and a DELETE call to remove data. If you need to access property use data, you can use a GET call. Once you run a POST call, the response will indicate that the POST was successful, but there is no user interface to see the values. If you want to see the values, you can run a GET call to query the database.

To add Property Use data, use this call:

Sample XML 10

```

POST /property/200/propertyUse

<k12School>
  <name>Broadway School</name>
  <useDetails>
    <totalGrossFloorArea units="Square Feet" currentAsOf="2010-08-13"
temporary="false">
      <value>333</value>
    </totalGrossFloorArea>
    <openOnWeekends currentAsOf="2010-08-13" temporary="false">
      <value>Yes</value>
    </openOnWeekends>
    <percentCooled currentAsOf="2010-08-13" temporary="false">
      <value>0</value>
    </percentCooled>
    <percentHeated currentAsOf="2010-08-13" temporary="false">
      <value>0</value>
    </percentHeated>
    <numberOfComputers currentAsOf="2010-08-13" temporary="false">
      <value>3</value>
    </numberOfComputers>
    <cookingFacilities currentAsOf="2010-08-13" temporary="false">
      <value>Yes</value>
    </cookingFacilities>
    <isHighSchool currentAsOf="2010-08-13" temporary="false">
      <value>Yes</value>
    </isHighSchool>
    <schoolDistrict currentAsOf="2010-08-13" temporary="false">
      <value>String</value>
    </schoolDistrict>
  </useDetails>
</k12School>

```

6. Retrieve Metrics – Once you have property, meter, and property use data entered, you can start receiving calculated metrics, such as the ENERGY STAR score. The GET Metrics service allows you to request and receive specific Metrics. That is, you will specify exactly which metrics you would like to see, rather than receiving a pre-determined set from EPA. The example below specifies the ENERGY STAR score, source energy use intensity, and total greenhouse gas emissions.

Header Field Name	Value
Content-Type	application/xml
PM-Metrics	score, sourceIntensity, totalGHGEmissions

Sample XML 11

```
GET /property/200/metrics?year=2012&month=12&measurementSystem=EPA
```

Please note that the seven services listed above are not the only services available. For example, the Meter Services listed are the basic services for adding and editing individual meters. There is also a Get Meter List service, which will return a full list of meters for a given property. Please refer to our complete API documentation for a description of all available web services: <http://portfoliomanager.energystar.gov/webservices/home/api>.

Testing Part C: Simulating Connection and Share Requests

If you have customers who already have properties in the Live environment of Portfolio Manager, then you will access their data via the connection and sharing functions of Portfolio Manager and the web services. In this process, your customers will request a connection to you and after you have accepted that connection, they will be able to share their properties and meters, which gives you authorization to manage their data.

These sharing actions are initiated by the customer in Portfolio Manager, but because the Test environment does not have a front-end web interface that simulates what the customer typically sees, you will have to simulate the generation of these connection and sharing requests via a web service.

EPA has provided a set of web services (available only in the Test environment) that you can use to mimic a customer submitting connection and share requests to you.

You should follow these steps for testing:

1. Create Sample Accounts and Properties – In order to begin to test the process of sharing, you will need to create sample accounts, properties, and meters. These accounts will eventually be your customers, who will send you connection and sharing requests. This process will be analogous to the process of creating accounts, properties, and meters, as discussed in Testing Part B, above.
2. Generate & Accept a Customer Connection Request – The first thing your customer will do is send you a connection request. To simulate this request, follow these steps:
 - a. *Determine the ID and password of the customer account that you want to generate a connection request from.* For the purposes of this example, assume the customer account has an ID of 500.
 - b. *Login to the customer account you have created – **You will need to use the customer’s login credentials to send yourself a connection request.***

- c. *Provide data for any custom fields.* If you have defined account-level custom fields, then you will assign custom field values to them in the connection request.

This simulates the customer providing data for any customer-level custom fields that you defined as they initiate the request to you. The following example assigns a custom field value of 5788 to the connection request that will be sent to your account. For the purposes of this example, assume your account has an ID of 900. Again, the web service must be called using the customer's login credentials.

```
PUT /account/customFieldList/recipient/900

<customFieldList>
  <customField name="Customer Number">5788</customField>
</customFieldList>
```

Sample XML 12

- d. *Submit the connection request to your account.* To simulate sending the connection request to you, you would run the following web service using the customer's login credentials. For the purposes of this example, assume your account has an ID of 900.

```
POST /invite/account/900
```

Sample XML 13

- e. *Logout as your customer and login as yourself.* Once the request has been sent, you need to **logout/close your customer's credentials and provide your own so that you can retrieve your pending requests.**
- f. *Retrieve the connection request.* To retrieve the pending connection request, you would run the following web service using your login credentials.

```
GET /connect/account/pending/list
```

Sample XML 14

The response for this GET call will look like sample xml 14 below.

```
<pendingList>
  <account>
    <accountId>87267</accountId>
    <customFieldList>
      <customField name="Username">Yes</customField>
    </customFieldList>
    <accountInfo>
      <address address1="45324 Labor Street" city="Fairfax" state="FL"
        postalCode="33843" country="US"/>
      <email>john_doe@acme.com</email>
      <firstName>John</firstName>
      <phone>123-456-7891</phone>
      <lastName>Doe</lastName>
      <jobTitle>Operations Manager</jobTitle>
      <organization>ACME Corp</organization>
    </accountInfo>
```

Sample XML 15

```

    <connectionAudit>
      <createdBy>Chris Jones123</createdBy>
      <createdByAccountId>75489</createdByAccountId>
      <createdDate>2014-12-03T08:49:29.000-05:00</createdDate>
      <lastUpdatedBy>Chris Jones123</lastUpdatedBy>
      <lastUpdatedByAccountId>75489</lastUpdatedByAccountId>
      <lastUpdatedDate>2014-12-03T13:49:29.000-05:00</lastUpdatedDate>
    </connectionAudit>
  </account>

  ...

  <links>
    <link linkDescription="next page" link="/connect/account/pending/list?page=2"
    httpMethod="GET"/>
  </links>
</pendingList>

```

In sample xml 14, the <accountInfo> and <accountID> fields will always contain the information of the account with which you are connecting. The <connectionAudit> field will contain the information for the account that *created the connection request*. These two fields do not necessarily need to be the same account. For instance, if a middleman account shares a property forward that is owned by another account, the middleman's information will appear in the <connectionAudit> field, and the information of the account that owns the building will appear in the <accountInfo> field. For more information, please see page 9 of [our Connection and Sharing document](#).

- g. *Accept the connection request.* To accept the pending connection request, you would run the following web service using your login credentials. For purposes of this example, assume the customer has an ID of 500.

```

POST /connect/account/500

```

Sample XML 16

```

<sharingResponse>
  <action>Accept</action>
  <note>Your connection request has been verified and accepted.</note>
</sharingResponse>

```

- h. *Verify the connection has been established.* To verify the connection, you would run the following web service using your login credentials to retrieve your list of customers.

```

GET /customer/list

```

Sample XML 17

3. **Generate & Accept a Property Share** – In the preceding step you connected with your customer **at the account level**. Once you and your customer are connected, your customer can share properties and meters with you. As noted above, when testing you will have to first create properties within your customer's account, as described in Testing Part B, above. To simulate a property share from the customer, you would follow these steps:

- a. *Determine the ID of the property for which you want to generate a share request.* For the purposes of this example, assume the property has an ID of 400.
- b. *Login to the customer account you have created – **You will need to use the customer’s login credentials to share the property with your account.***
- c. *Provide data for any custom fields.* If you have defined property-level custom fields, then you will assign custom field values to them in the share request. This simulates the customer providing data for any property-level custom fields that you defined as they initiate the request to you. The following example assigns a custom field value of BH971 to the share request that will be sent to your account. For the purposes of this example, assume your account has an ID of 900.

```

PUT /property/400/customFieldList/recipient/900

<customFieldList>
  <customField name="Lot Number">BH971</customField>
</customFieldList>
```

Sample XML 18

- d. *Submit the property share request to your account.* To simulate sending the share request to you, you would run the following web service using the customer’s login credentials. For the purposes of this example, assume the property has an ID of 400 and your account has an ID of 900.

```

POST /invite/account/900/property/400
```

Sample XML 19

- e. *Logout of your customer account and login to your own account.* Once the property share has been sent, **you need to logout/close your customer’s credentials and provide your own so that you can retrieve your pending property shares.**
- f. *Retrieve the property share request.* To retrieve the pending property share request, you would run the following web service using your login credentials.

```

GET /share/property/pending/list
```

Sample XML 20

The response for this GET call will look like the sample xml here:

```

GET /share/property/pending/list

<pendingList>
  <property>
    <propertyId>400</propertyId>
    <customFieldList>
      <customField name="Lot Number">BH971</customField>
    </customFieldList>
    <accessLevel>Read Write</accessLevel>
    <accountId>900</accountId>
    <propertyInfo>
      <name>Broadway School</name>
      <address address1="12321 Main Street" city="Arlington" state="VA"
        postalCode="22201" country="US"/>
      <numberOfBuildings>3</numberOfBuildings>
      <constructionStatus>Test</constructionStatus>
```

Sample XML 21

```

    <primaryFunction>K-12 School</primaryFunction>
    <yearBuilt>2000</yearBuilt>
    <grossFloorArea units="Square Feet" temporary="false" default="N/A">
      <value>10000</value>
    </grossFloorArea>
    <occupancyPercentage>80</occupancyPercentage>
    <isFederalProperty>false</isFederalProperty>
    <audit>
      <createdBy>acme_dx_user</createdBy>
      <createdByAccountId>400</createdByAccountId>
      <createdDate>2014-04-01T13:51:15.000-04:00</createdDate>
      <lastUpdatedBy>acme_dx_user</lastUpdatedBy>
      <lastUpdatedByAccountId>400</lastUpdatedByAccountId>
      <lastUpdatedDate>2014-09-03T23:29:50.000-04:00</lastUpdatedDate>
    </audit>
  </propertyInfo>
  <shareAudit>
    <createdBy>my_cust_user</createdBy>
    <createdByAccountId>100</createdByAccountId>
    <createdDate>2015-02-10T13:01:37.000-05:00</createdDate>
    <lastUpdatedBy>my cust user</lastUpdatedBy>
    <lastUpdatedByAccountId>100</lastUpdatedByAccountId>
    <lastUpdatedDate>2015-02-10T13:01:37.000-05:00</lastUpdatedDate>
  </shareAudit>
</property>

...

  <links>
    <link linkDescription="next page" link="/share/property/pending/list?page=2"
httpMethod="GET"/>
  </links>
</pendingList>

```

The `createdBy` tag toward the bottom of the xml response notes which account created the share request. For share requests made directly to your account by the PDA account, the account identified here will be the same as the account specified in the `accountId` tag toward the top of the xml response. However, the `createdBy` tag will instead note a middleman account if the property was shared with you by an account that is not the PDA. For more information, please see page 9 of our [Connection and Sharing document](#).

- g. *Accept the property share request.* To accept the pending property share request, you would run the following web service using your login credentials. For the purposes of this example, assume the property has an ID of 400.

Sample XML 22

```

POST /share/property/400

<sharingResponse>
  <action>Accept</action>
  <note>Your share request has been verified and accepted.</note>
</sharingResponse>

```

- h. *Verify the share has been established.* To verify the share, you would run the following web service using your login credentials to retrieve the property. For purposes of this example, assume the property has an ID of 400.

Sample XML 23

```

GET /property/400

```

4. Generate & Accept a Meter Share Request – In the preceding step you accepted your customer’s property. There is one more step required to provide you with access at the individual meter level. When your customer sent you the property request, they also would have sent you meter share requests for any meters they want you to service.
5. In order to simulate this meter share, you will have to make sure your customer has created meters (see Testing Part B to create meters). To simulate a meter share from the customer, you would follow these steps:
 - a. *Determine the ID of the meter for which you want to generate a share request.* For the purposes of this example, assume the meter has an ID of 450 and the property has an ID of 400.
 - b. *Login to the customer account you have created – **You will need to use the customer’s login credentials to share the meter with your account.***
 - c. *Provide data for any custom fields.* If you have defined meter-level custom fields, then you will assign custom field values to them in the share request. This simulates the customer providing data for any meter-level custom fields that you defined as they initiate the request to you.
 - d. The following example assigns a custom field value of M324 to the share request that will be sent to your account. For the purposes of this example, assume your account has an ID of 900.

Sample XML 24

```
PUT /meter/450/customFieldList/recipient/900

<customFieldList>
  <customField name="Lot Number">M324</customField>
</customFieldList>
```

- e. *Submit the meter share request to your account.* To simulate sending the share request to you, you would run the following web service using the customer’s login credentials. For the purposes of this example, assume the meter has an ID of 450 and your account has an ID of 900.

Sample XML 25

```
POST /invite/account/900/meter/450
```

- f. *Logout of your customer account and login to your own account.* Once the meter share has been sent, **you need to logout/close your customer’s credentials and provide your own so that you can retrieve your pending meter shares.**
- g. *Retrieve the meter share request.* To retrieve the pending meter share request, you would run the following web service using your login credentials.

Sample XML 26

```
GET /share/meter/pending/list
```

Note that the response from this call will be similar to Sample XML 19, above.

- h. *Accept the meter share request.* To accept the pending meter share request, you would run the following web service using your login credentials. For purposes of this example, assume the meter has an ID of 450.

```
POST /share/meter/450
```

Sample XML 27

```
<sharingResponse>
  <action>Accept</action>
  <note>Your share request has been verified and accepted.</note>
</sharingResponse>
```

- i. *Verify the share has been established.* To verify the share, you would run the following web service using your login credentials to retrieve the meter. For purposes of this example, assume the meter has an ID of 450.

```
GET /meter/450
```

Sample XML 28

Data Migration

When you move from the Test environment to the Live environment, no data is migrated from your test account. In fact, you must create a new Portfolio Manager account at (<https://portfoliomanager.energystar.gov/pm/>) to access the Live environment. This new account has no relation to your test account.

Speed/Performance

The Test environment's infrastructure consists of its own separate database and application servers from the Live environment. Since this is a Test environment, it is configured on a smaller scale and does not intend to reflect the actual level of speed and performance that you would experience in the Live environment.