

Testing Web Services

August 2020

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.

There is also a companion video tutorial that walks through the testing procedures outlined in this document. The full video can be viewed [here](#), and there are timestamps to specific sections noted **in bold** throughout this document. Click on the timestamps in bold throughout this document to refer to a specific section of the online video.

Testing Tools and Requirements **(Video starts at 00:00:06)**

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 calls directly to and from 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/hgmlloofddffdnphfgcellkdfbfjeloo>).

Testing Overview

The web services 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. This will help you to build the correct calls and logs into your software. It is important to note that 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 to another account 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 begin by creating test customer accounts and properties. You will use these test accounts to send your own account a simulated connection request. This step is analogous to your customer sending you a connection request in the Live environment. After you accept the connection, you can also generate simulated 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. In the Test environment, you will have the opportunity to accept or reject each property or meter share request, just as you would in the Live environment. 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, and 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 the Test environment will be migrated into the Live Portfolio Manager.

Testing Part A: Create an Account ([Video Step 1; starts at 00:02:04](#))

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

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

```

POST /account

<account>
  <username>SampleJohnDoeAccount</username>
  <password>Sample123</password>
  <webserviceUser>true</webserviceUser>
  <searchable>true</searchable>
  <contact>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <address address1="123 Main St" city="Edmonton"
      state="AB" postalCode="T5G 2S7" country="CA"/>
    <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>
  <emailPreferenceCanadianAccount>true</emailPreferenceCanadianAccount>
</account>

```

4. Submit the request.
5. The XML response will indicate whether or not your account was created successfully. When the response indicates that your account was created successfully, it will also return a unique account ID. Make sure to save and store this ID, along with your username and password, in order to reference your account in subsequent web service calls.

6. Manage your account settings, as needed. 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 [\(Video Step 2; starts at 00:10:42\)](#).

The following example shows how you would update your data exchange settings through web services:

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>Municipally Supplied Potable Water - Indoor</meterType>
    <meterType>Disposed - Trash</meterType>
  </supportedMeterTypes>
</dataExchangeSettings>

```

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

Sample XML 3

```

GET /dataExchangeSetting

```

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 Customer Accounts [\(Video Step 3; starts at 00:14:37\)](#). 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 methods, 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 account, use this call:

Sample XML 4

```

POST /customer

<account>
  <username>ENTER_CUSTOMER_USERNAME</username>
  <password>ENTER_CUSTOMER_PASSWORD</password>
  <webserviceUser>true</webserviceUser>
  <searchable>>false</searchable>
  <contact>
    <firstName>Jack</firstName>
    <lastName>Brown</lastName>
    <address address1="123 South St" city="Edmonton"
      state="AB" postalCode="T5G 2S7" country="CA"/>
    <email>jack_brown@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>
  <emailPreferenceCanadianAccount>true</emailPreferenceCanadianAccount>
</account>

```

The XML response will indicate whether or not the customer account was created successfully. When the response indicates that the account was created successfully, it will also return a unique account ID. Make sure to save and store this ID, along with your username and password, in order to reference your account in subsequent web service calls.

2. Create Properties within your Customer Account ([Video Step 4; starts at 00:18:21](#)). 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/(customer_accountId)/property

<property>
  <name>Broadway School</name>
  <primaryFunction>K-12 School</primaryFunction>
  <address address1="123 South St" city="Edmonton"
    state="AB" postalCode="T5G 2S7" country="CA"/>
  <yearBuilt>2000</yearBuilt>
  <constructionStatus>Existing</constructionStatus>
  <grossFloorArea temporary="false" units="Square Feet">
    <value>10000</value>
  </grossFloorArea>
  <occupancyPercentage>80</occupancyPercentage>
  <isFederalProperty>>false</isFederalProperty>

```

```
</property>
```

The XML response will indicate whether or not the customer property was created successfully. When the response indicates that the property was created successfully, it will also return a unique property ID. Make sure to save and store this ID in order to reference this property in subsequent web service calls.

3. Add Property Use Data ([Video Step 5; starts at 00:24:08](#)). 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 6

```
POST /property/(propertyID)/propertyUse

<k12School>
  <name>Broadway K-12 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>
    <monthsInUse currentAsOf="2010-08-13" temporary="false">
      <value>8</value>
    </monthsInUse>
    <schoolDistrict currentAsOf="2010-08-13" temporary="false">
      <value>String</value>
    </schoolDistrict>
  </useDetails>
</k12School>
```

The XML response will indicate whether or not the property use was created successfully. When the response indicates that the property use was created successfully, it will also return a unique property use ID. Make sure to save and store this ID in order to reference this property use in subsequent web service calls.

4. Create Meters. Once you have created a property, you can create meters for that property, including energy meters, water meters, waste 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 ([Video Step 6a; starts at 00:29:10](#)), use this call:

```

Sample XML 7
POST /property/(propertyId)/meter

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

```

The XML response will indicate whether or not the energy meter was created successfully. When the response indicates that the energy meter was created successfully, it will also return a unique energy meter ID. Make sure to save and store this ID in order to reference this meter in subsequent web service calls.

To create a water meter ([Video Step 6b; starts at 00:31:40](#)), use this call:

```

Sample XML 8
POST /property/(propertyId)/meter

<meter>
  <type>Municipally Supplied Potable Water - Indoor</type>
  <name>Potable Indoor Meter</name>
  <metered>true</metered>
  <unitOfMeasure>Gallons (US)</unitOfMeasure>
  <firstBillDate>2014-01-01</firstBillDate>
  <inUse>true</inUse>
</meter>

```

The XML response will indicate whether or not the water meter was created successfully. When the response indicates that the water meter was created successfully, it will also return a unique water meter ID. Make sure to save and store this ID in order to reference this meter in subsequent web service calls.

To create a waste meter ([Video Step 6c; starts at 00:32:58](#)), use this call:

```

Sample XML 9
POST /property/(propertyId)/meter

<wasteMeter>
  <name>regularDisposedWasteEstimation</name>

```

```

<type>Disposed - Trash</type>
<unitOfMeasure>Cubic yards</unitOfMeasure>
<dataEntryMethod>regular container</dataEntryMethod>
<containerSize>2</containerSize>
<firstBillDate>2000-01-01</firstBillDate>
<inUse>true</inUse>
</wasteMeter>

```

The XML response will indicate whether or not the waste meter was created successfully. When the response indicates that the waste meter was created successfully, it will also return a unique waste meter ID. Make sure to save and store this ID in order to reference this meter in subsequent web service calls.

5. Associate Meters ([Video Step 7; starts at 00:34:30](#)). Newly created meters are not automatically included in the calculation of energy, water, or waste use metrics for a property. In order for that to happen, a meter must be “associated” with a property.

You can associate a meter with a property using a POST call. Note that the POST /association/property/(propertyId)/meter 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(s), or the previous associations will be lost. This POST 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.

If you are only seeking to associate a single, specific meter to a property, you would use the POST /association/property/(propertyId)/meter/(meterId) call. The status of any other meters currently associated with the property will be unaffected.

Sample XML 10 provides an example of a “full meter” association call. In other words, it associates a specified list of meters to a given property and overwrites the list of meters that were previously associated with this property. Again, this call affects only meters within the category, so this example call demonstrates associations for energy, water and waste meters.

Sample XML 10

```

POST /association/property/(propertyId)/meter

<meterPropertyAssociationList>
  <energyMeterAssociation>
    <meters>
      <meterId>(energy_meterId)</meterId>
    </meters>
    <propertyRepresentation>
      <propertyRepresentationType>Whole Property</propertyRepresentationType>
    </propertyRepresentation>
  </energyMeterAssociation>

  <waterMeterAssociation>
    <meters>
      <meterId>(water_meterId)</meterId>
    </meters>

```



```

        <propertyRepresentation>
            <propertyRepresentationType>Whole Property</propertyRepresentationType>
        </propertyRepresentation>
    </waterMeterAssociation>

    <wasteMeterAssociation>
        <meters>
            <meterId>(waste_meterId)</meterId>
        </meters>
    </wasteMeterAssociation>
</meterPropertyAssociationList>

```

Sample XML 11 shows an individual meter association call. In this case, we would be associating just one meter to this property, without affecting other associated meters.

Sample XML 11

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

The XML response will indicate whether or not the meter association was performed successfully. To see/confirm the complete list of meters associated with this property, you would use the GET /association/property/(propertyId)/meter call.

6. 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. Cost associated with a specific consumption record can be provided as well, although it is optional. 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 energy meter consumption data ([Video Step 8a; starts at 00:40:55](#)), use this call:

Sample XML 12

```

POST /meter/(energy_meterId)/consumptionData

<meterData>
    <meterConsumption estimatedValue="true">
        <cost>21</cost>
        <startDate>2018-12-01</startDate>
        <endDate>2018-12-31</endDate>
        <usage>130</usage>
    </meterConsumption>
    <meterConsumption estimatedValue="false">
        <cost>20</cost>
        <startDate>2018-11-01</startDate>
        <endDate>2018-11-30</endDate>
        <usage>120</usage>
    </meterConsumption>

```

```
</meterData>
```

The XML response will indicate whether or not the meter consumption data were created successfully. When the response indicates that the meter consumption data were created successfully, it will also return a unique meter consumption data ID for each individual consumption record (e.g., each monthly entry). Make sure to save and store these IDs in order to reference these meter consumption records in subsequent web service calls.

To add water meter consumption data ([Video Step 8b; starts at 00:45:44](#)), use this call:

Sample XML 13

```
POST /meter/(water_meterId)/consumptionData

<meterData>
  <meterConsumption estimatedValue="true">
    <cost>21</cost>
    <startDate>2018-12-01</startDate>
    <endDate>2018-12-31</endDate>
    <usage>130</usage>
  </meterConsumption>
  <meterConsumption estimatedValue="false">
    <cost>20</cost>
    <startDate>2018-11-01</startDate>
    <endDate>2018-11-30</endDate>
    <usage>120</usage>
  </meterConsumption>
</meterData>
```

The XML response will indicate whether or not the meter consumption data were created successfully. When the response indicates that the meter consumption data were created successfully, it will also return a unique meter consumption data ID for each individual consumption record (e.g., each monthly entry). Make sure to save and store these IDs in order to reference these meter consumption records in subsequent web service calls.

To add waste meter consumption data ([Video Step 8c; starts at 00:47:38](#)), use this call:

Sample XML 14

```
POST /meter/(water_meterId)/consumptionData

<wasteDataList>
  <wasteData>
    <startDate>2018-01-01</startDate>
    <endDate>2018-01-31</endDate>
    <timesEmptied>4</timesEmptied>
    <averagePercentFull>75</averagePercentFull>
    <cost>50</cost>
    <disposalDestination>
      <incinerationPercentage>25</incinerationPercentage>
      <landfillPercentage>25</landfillPercentage>
      <unknownDestPercentage>5</unknownDestPercentage>
      <wasteToEnergyPercentage>45</wasteToEnergyPercentage>
    </disposalDestination>
  </wasteData>
```

```

<wasteData>
  <startDate>2018-02-01</startDate>
  <endDate>2018-02-28</endDate>
  <timesEmptied>4</timesEmptied>
  <averagePercentFull>100</averagePercentFull>
  <cost>50</cost>
  <disposalDestination>
    <incinerationPercentage>25</incinerationPercentage>
    <landfillPercentage>25</landfillPercentage>
    <unknownDestPercentage>5</unknownDestPercentage>
    <wasteToEnergyPercentage>45</wasteToEnergyPercentage>
  </disposalDestination>
</wasteData>
</wasteDataList>

```

The XML response will indicate whether or not the meter consumption data were created successfully. When the response indicates that the meter consumption data were created successfully, it will also return a unique meter consumption data ID for each individual consumption record (e.g., each monthly entry). Make sure to save and store these IDs in order to reference these meter consumption records in subsequent web service calls.

- Retrieve Metrics ([Video Step 9; starts at 00:50:51](#)). 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 receive, from a full list of more than 1,600 metrics available [here](#). In the example call below, the calculated metrics being requested are the ENERGY STAR score, source energy use intensity, and total greenhouse gas emissions for the period ending December 31, 2018.

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

Sample XML 15

```
GET /property/{propertyId}/metrics?year=2018&month=12&measurementSystem=EPA
```

Please note that the 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 similar to social media platforms, 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 represent 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. Terminate the existing connection/shares between your Test account and your customer's Test account/property/meter records ([Video Step 10; starts at 00:59:00](#)).

When creating simulated customer accounts, properties, and meters under Testing Part B, above, account-level connections and property/meter-level shares are automatically carried out. In order to continue simulating customer-initiated connection and share requests, you will need to manually remove these connections and shares. To do this, please use the POST `/disconnect/account/{accountId}?keepShares={keepShares}` call, and set the `?keepShares` parameter to "false." This step only needs to be taken at the account/disconnect level, since an existing property and meter shares between your account and the customer account will be removed.

This service can be run while authenticated as the customer OR as your main account (both users are set as "web services" users and can initiate a disconnect). The "accountId" specified in the URL will be the account from which the authenticated user wishes to disconnect. In example XML 16 below, we are disconnecting from the customer's account while authenticated using the main account's credentials

Sample XML 16

```
POST /disconnect/account/{customer_accountId}?keepShares=false
<terminateSharingResponse>
  <note>Disconnecting to test a simulated connection request and property share request.</note>
</terminateSharingResponse>
```

Please note that this step is only required within the Test environment, for the purposes of placing and processing simulated customer connection/share requests. This is not a "typical" business process that will need to be performed within the Live environment.

3. 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 from which you wish to generate a connection request.
- 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 via your Test provider account, then you will need to provide custom field values as part of the simulated customer connection request.

Sample XML 17 below simulates the process by which the customer would provide data for an account-level custom field that you (as the provider) have required. In the Live environment, this step would be required in order for the customer to initiate the connection request. The following example assigns a custom field value of 5788 to the connection request that will be sent to your account. Again, the web service must be called using the customer's login credentials, and so the accountId that you specify in the URL would be the ID for your Test provider account.

Sample XML 17

```
PUT /account/customFieldList/recipient/{accountId}

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

- d. Submit a Simulated Connection Request from your Customer's Account to your Main Account ([Video Step 11; starts at 01:02:00](#)). To simulate a customer sending a connection request to you, you would run the following web service, while still using the test customer's login credentials.

Sample XML 18

```
POST /invite/account/{accountId}
```

- e. Log out 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 ([Video Step 12; starts at 01:04:15](#)). To retrieve the pending connection request, you would run the following web service using your login credentials.

Sample XML 19

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

The response for this GET call will look like sample XML 20 below.

Sample XML 20

```
<pendingList>
  <account>
    <accountId>87267</accountId>
    <username>JohnDoeTestAccount</username>
    <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>
    <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 20, the `<accountInfo>` and `<accountId>` fields will always contain the information for the account to 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* ([Video Step 13; starts at 01:06:04](#)). To accept the pending connection request, you would run the following web service using your login credentials.

Sample XML 21

```
POST /connect/account/(customer_accountId)

<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 connected customers.

```
GET /customer/list
```

Sample XML 22

You could also run the GET Pending Connections call again, to confirm that this customer account is no longer showing up on the list.

4. **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:
- Determine the ID of the property for which you want to generate a share request.*
 - 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.***
 - Provide data for any custom fields.* If you have defined property-level custom fields via your Test provider account, then you will need to provide custom field values as part of the simulated property share request.

Sample XML 23 below simulates the process by which the customer would provide data for a property-level custom field that you (as the provider) have required. In the Live environment, this step would be required in order for the customer to initiate the property share request. The following example assigns a custom field value of BH971 to the property share request that will be sent to your account. Again, the web service must be called using the customer's login credentials, and so the accountId that you specify in the URL would be the ID for your Test provider account.

```
PUT /property/(propertyId)/customFieldList/recipient/(accountId)

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

Sample XML 23

- d. *Submit the property share request to your account* ([Video Step 14: starts at 01:09:09](#)). To simulate the customer sending a property share request to you, you would run the following web service while still using the test customer's login credentials.

```
POST /invite/account/(accountId)/property/(propertyId)
```

Sample XML 24

- e. Log out 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 ([Video Step 15: starts at 01:14:31](#)). 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 25

The response for this GET call will look like sample XML 26 below:

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

Sample XML 26

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


The <createdBy> tag toward the bottom of the XML response notes which account initiated the share request. For share requests made directly to your account by the Property Data Administrator (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, if the property was shared with you by an account that is not the PDA, the <createdBy> tag will instead note a “middleman account.” For more information, please see page 9 of our [Connection and Sharing document](#).

- g. *Accept the property share request* ([Video Step 16; starts at 01:16:31](#)). To accept the pending property share request, you would run the following web service using your login credentials.

<pre>POST /share/property/(propertyId) <sharingResponse> <action>Accept</action> <note>Property share request has been verified and accepted.</note> </sharingResponse></pre>	Sample XML 27
--	----------------------

- h. *Verify the share has been established* ([Video Step 17; starts at 01:18:55](#)). To verify the property share, you would run the following web service using your login credentials to retrieve the list of properties to which you have shared access.

<pre>GET /property/(propertyId)</pre>	Sample XML 28
---------------------------------------	----------------------

You could also run the GET Pending Property Shares call again, to confirm that this property is no longer showing up on the list.

5. **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. In the Live Environment, when your customer sends you the property request, they also have the opportunity to send you share requests for any specific meters they want you to service. In order to simulate this customer-initiated meter sharing process, you will have to make sure that you have created meters at the simulated/test property that you have set up for your test customer (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.*
 - 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 via your Test provider account, then you will need to provide custom field values as part of the simulated meter share request.

Sample XML 28 below simulates the process by which the customer would provide data for a meter-level custom field that you (as the provider) have required. In the Live environment, this step would be required in order for the customer to initiate the meter share request. The following example assigns a custom field value of M324 to the meter share request that will be sent to your account. Again, the web service must be called using the customer's login credentials, and so the accountId that you specify in the URL would be the ID for your Test provider account.

```
PUT /meter/(energy_meterId)/customFieldList/recipient/(accountId)

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

Sample XML 29

- d. *Submit the meter share request to your account* ([Video Step 14a; starts at 01:14:16](#)). To simulate the customer sending a meter share request to you, you would run the following web service while still using the customer's login credentials.

```
POST /invite/account/(accountId)/meter/(energy_meterId)
```

Sample XML 30

- e. *Log out 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.**
- f. *Retrieve the meter share request* ([Video Step 18; starts at 01:19:40](#)). To retrieve the pending meter share request, you would run the following web service using your login credentials.

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

Sample XML 31

Note that the response from this call will be similar to Sample XML 26, above.

- g. *Accept the meter share request* ([Video Step 19; starts at 01:19:47](#)). To accept the pending meter share request, you would run the following web service using your login credentials.

```
POST /share/meter/(energy_meterId)

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

Sample XML 32

```
</sharingResponse>
```

- h. *Verify the share has been established* ([Video Step 20; starts at 01:20:00](#)). To verify the meter share, you would run the following web service using your login credentials to retrieve the list of meters to which you have shared access.

```
GET /meter/{energy_meterId}
```

Sample XML 33

You could also run the GET Pending Meter Shares call again, to confirm that this meter is no longer showing up on the list.

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.