# Use Summary

**http://simon.ist.rit.edu:8080/Services/resources/ESD...**
try these links and see the results:

- http://simon.ist.rit.edu:8080/Services/resources/ESD/Organizations
- http://simon.ist.rit.edu:8080/Services/resources/ESD/Organizations?type=&name=&town=&state=NY&zip=14534&county=

To use with the proxy server:

```
$.ajax({
        type: "GET",
        async: true,
        cache:false,
        url: "proxy.php",
        data: {path: "/Organizations?type="+escape($('#orgType').val())+"&name="+escape($('#orgName').val())+"&town="+escape($('#city
        dataType: "xml",
        success: function(data, status){
                //do something with data (like a each() iterator!)
        }
});
```

It is that simple!
Half of these you wont use/need, but you need to figure that out!

# Method Detail

## getOrgs

Returns organizations matching specified input criteria. Access is via GET.

**Path:** ...ESD/Organizations

**Parameters:**
type - (optional) The organization type (e.g., "Ambulance", "Hospital").
name - (optional) The name of the organization. A match will be generated if the passed value matches any part of the organizatio
town - (optional) The town where the organization is located. Only the main office location will be checked.
state - (optional) The state where the organization is located. Only the main office location will be checked.
zip - (optional) The zip code where the organization is located. Only the main office location will be checked.
county - (optional) The county where the organization is located. Only the main office location will be checked.

**Returns:**
An XML string containing the organizations that meet the specified criteria. The XML structure is as follows:>

```
<data>
  <row>
    <OrganizationID\>
    <type\>
    <Name\>
    <Email\>
    <city\>
    <state\>
    <zip\>
    <CountyName\>
  </row>
</data>
```

If a Type of "Physician" is specified, the return XML structure contains the following:

```
<data>
  <row>
    <fName\>
    <mName\>
    <lName\>
    <OrganizationID\>
    <type\>
    <Name\>
    <city\>
    <state\>
    <zip\>
    <CountyName\>
    <phone\>
  </row>
</data>
```

Order of elements is not guaranteed. Multiple organizations will be represented in multiple <row> elements.

---

## getOrgTypes

Lookup function that returns a list of possible organization types.

**Path:** ...ESD/OrgTypes

**Parameters:**
none -
**Returns:**
An XML string containing a list of possible organization types. The XML structure is as follows:>

```
<data>
  <row>
    <typeId\>
    <type\>
  </row>
</data>
```

Order of elements is not guaranteed. Multiple organization types will be listed via multiple row elements.

---

## getTabs

Returns the tabs for a specific orgId. Access is via GET.

**Path:** ...ESD/Application/Tabs

Possible Tabs are: General, Locations, Treatment, Training, Facilities, Equipment, Physicians, People

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.
**Returns:**
An XML string containing the tabs for a specific organization. The XML structure is as follows:

```
<data>
        <row>
                <Tab>General</Tab>
        </row>
        <row>
                <Tab>Locations</Tab>
        </row>
        <row>
                <Tab>Treatment</Tab>
        </row>
        <row>
                <Tab>Training</Tab>
        </row>
        <row>
                <Tab>Facilities</Tab>
        </row>
        <row>
                <Tab>Equipment</Tab>
        </row>
        <row>
                <Tab>Physicians</Tab>
        </row>
        <row>
                <Tab>People</Tab>
        </row>
</data>
```

---

## getGeneralInfo

Returns general information about specified organization. Access is via GET.

**Path:** ...ESD/{orgId}/General

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.
**Returns:**
An XML string containing the organization's general information. The XML structure is as follows:>

```
<data>
```

```
            <name\>
            <email\>
            <website\>
            <description\>
            <nummembers\>
            <numcalls\>
            <serviceArea\>
          </data>
```

Order of elements is not guaranteed. Not all elements will be returned for all organization types.

---

**getPhysiciansInfo**

Returns information regarding all physicians who have admitting privleges at the specified hospital. Access is via GET.

**Path:** ...ESD/{orgId}/Physicians

**Parameters:**
orgId - (required in path) The hospital's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.

**Returns:**
An XML string listing the authorized physicians. The XML structure is as follows:>

```
       <data>
         <count>
         <physician>
           <personId\>
           <fName\>
           <mName\>
           <lName\>
           <suffix\>
           <phone\>
           <license\>
         </physician>
       </data>
```

Order of elements is not guaranteed. Multiple physicians will be listed via multiple physician elements. The number of physician

---

**getAssetsInfo**

Returns information regarding available facilities at the specified organization. Access is via GET.

**Path:** ...ESD/{orgId}/Facilities

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are view (default if not specified) and edit.

**Returns:**
An XML string containing the organization's facilities information. The XML structure is as follows:>

```
       <data>
         <count>
         <facility>
           <typeId\>
           <type\>
           <quantity\>
           <description\>
         </facility>
       </data>
```

If mode is set to edit, all possible facility values will be returned and an additional element named "checked" will indicate wh

Order of elements is not guaranteed. Multiple facilities will be listed via multiple facility elements. The number of facility el

---

**getTreatmentsInfo**

Returns information regarding available treatment at the specified organization. Access is via GET.

**Path:** ...ESD/{orgId}/Treatments

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.

**Returns:**
An XML string containing the organization's treatment information. The XML structure is as follows:>

```
<data>
  <count>
  <treatment>
    <typeId\>
    <type\>
    <abbreviation\>
  </treatment>
</data>
```

If mode is set to edit, all possible treatment values will be returned and an additional element named "checked" will indicate w

Order of elements is not guaranteed. Multiple treatments will be listed via multiple treatment elements. The number of treatment

---

**getLocationInfo**

Returns information regarding all locations of the specified organization. Access is via GET.

**Path:** ...ESD/{orgId}/Locations

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.

**Returns:**
An XML string containing the organization's locations. The XML structure is as follows:>

```
<data>
  <count>
  <location>
    <type\>
    <address1\>
    <address2\>
    <city\>
    <state\>
    <zip\>
    <phone\>
    <ttyphone\>
    <fax\>
    <latitude\>
    <longitude\>
    <countyId\>
    <countyName\>
    <siteId\>
  </location>
</data>
```

Order of elements is not guaranteed. Multiple locations will be listed via multiple location elements. The number of location ele

---

**getTrainingInfo**

Returns information regarding available training at the specified organization. Access is via GET.

**Path:** ...ESD/{orgId}/Training

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.

**Returns:**
An XML string containing the organization's training information. The XML structure is as follows:>

```
<data>
  <count>
  <training>
    <typeId\>
    <type\>
    <abbreviation\>
  </training>
</data>
```

If mode is set to edit, all possible training values will be returned and an additional element named "checked" will indicate wh

Order of elements is not guaranteed. Multiple training services will be listed via multiple training elements. The number of trai

---

## getEquipmentInfo

Returns information regarding available equipment at the specified organization. Access is via GET.

**Path:** ...ESD/{orgId}/Equipment

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.
**Returns:**
An XML string containing the organization's equipment. The XML structure is as follows:>

```
<data>
  <count>
  <equipment>
    <typeId\>
    <type\>
    <quantity\>
    <description\>
  </equipment>
</data>
```

If mode is set to edit, all possible equipment types will be returned and an additional element named "checked" will indicate wh

Order of elements is not guaranteed. Multiple equipment entries will be listed via multiple equipment elements. The number of equ

---

## getPeopleInfo

Returns information regarding personnel associated with the specified organization. Each of the organization's locations is listed wit

**Path:** ...ESD/{orgId}/People

**Parameters:**
orgId - (required in path) The organization's id.
mode - (optional) How the information will be used. Possible values are View (default if not specified) and Edit.
token - (optional) Security token obtained at login.
ip - (optional) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
**Returns:**
An XML string containing the organization's personnel. The XML structure is as follows:>

```
<data>
  <siteCount>
  <site address= siteId= siteType= >
    <personCount\>
    <person\>
      <personId\>
      <honorific\>
      <fName\>
      <mName\>
      <lName\>
      <suffix\>
      <role\>
      <contactMethods>
        <method>
          <type\>
          <value\>
        <\method>
      <\contactMethods>
```

```
              <person\>
            </site>
          </data>

         Order of elements is not guaranteed. Multiple locations will be listed via multiple site elements. The number of site elements is
          Similarly, multiple people at a given site will be listed via multiple person elements with the number of person elements availa
          Finally, each person may have multiple contact methods. Each manner of contact is listed in a method element. There may be multi
```

---

You will definitely use the methods above - might use the methods below...

---

**getTreatments**

Lookup function that returns a list of possible treatments.

**Path:** ...ESD/Treatments

**Parameters:**
none -
**Returns:**
An XML string containing a list of possible treatments. The XML structure is as follows:>

```
 <data>
   <row>
     <typeId\>
     <type\>
   </row>
 </data>
```

Order of elements is not guaranteed. Multiple treatment entries will be listed via multiple row elements.

---

**getTraining**

Lookup function that returns a list of possible training services.

**Path:** ...ESD/Training

**Parameters:**
none -
**Returns:**
An XML string containing a list of possible training services. The XML structure is as follows:>

```
 <data>
   <row>
     <typeId\>
     <type\>
   </row>
 </data>
```

Order of elements is not guaranteed. Multiple training entries will be listed via multiple row elements.

---

**getEquipment**

Lookup function that returns a list of possible equipment types.

**Path:** ...ESD/Training

**Parameters:**
none -

**Returns:**
An XML string containing a list of possible equipment types. The XML structure is as follows:>

```
<data>
  <row>
    <typeId\>
    <type\>
  </row>
</data>
```

Order of elements is not guaranteed. Multiple equipment types will be listed via multiple row elements.

---

## getFacilities

Lookup function that returns a list of possible types of facilities.

**Path:** ...ESD/Facilities

**Parameters:**
none -
**Returns:**
An XML string containing a list of possible types of facilities. The XML structure is as follows:>

```
<data>
  <row>
    <typeId\>
    <type\>
  </row>
</data>
```

Order of elements is not guaranteed. Multiple types of facilities will be listed via multiple row elements.

---

## getLocationTypes

Lookup function that returns a list of possible location types.

**Path:** ...ESD/LocationTypes

**Parameters:**
none -
**Returns:**
An XML string containing a list of possible location types. The XML structure is as follows:>

```
<data>
  <type\>
</data>
```

Order of elements is not guaranteed. Multiple location types will be listed via multiple type elements.

---

## getContactTypes

Lookup function that returns a list of possible contact types.

**Path:** ...ESD/OrgTypes

**Parameters:**
none -
**Returns:**

An XML string containing a list of possible location types. The XML structure is as follows:>

```
<data>
  <type\>
</data>
```

Order of elements is not guaranteed. Multiple location types will be listed via multiple type elements.

---

**getCities**

Lookup function that returns a list of possible cities.

**Path:** ...ESD/Cities

**Parameters:**
state - (optional) Results are limited to specified state.
county - (optional) Results are limited to specified county.

**Returns:**
An XML string containing a list of possible cities. The XML structure is as follows:>

```
<data>
  <row>
    <city\>
  </row>
</data>
```

Order of elements is not guaranteed. Multiple cities will be listed via multiple row elements.

---

**getStates**

Lookup function that returns a list of possible states (2-letter abbreviation).

**Path:** ...ESD/States

**Parameters:**
none -

**Returns:**
An XML string containing a list of possible states. The XML structure is as follows:>

```
<data>
  <row>
    <State\>
  </row>
</data>
```

Order of elements is not guaranteed. Multiple states will be listed via multiple row elements.

---

**getCounties**

Lookup function that returns a list of possible counties.

**Path:** ...ESD/Counties

**Parameters:**
state - (optional) Results are limited to specified state.

**Returns:**
An XML string containing a list of possible counties. The XML structure is as follows:>

```
<data>
```

```
            <row>
              <CountyID\>
              <CountyName\>
            </row>
          </data>
```

Order of elements is not guaranteed. Multiple counties will be listed via multiple row elements.

---

**getCounty**

Lookup function that return county associated with a specified zipcode

**Path:** ...ESD/County

**Parameters:**
zip - (required) 5-digit zip code whose county is being sought.
**Returns:**
An XML string containing county of specified zip code. The XML structure is as follows:>

```
 <data>
   <row>
     <CountyID\>
     <CountyName\>
   </row>
 </data>
```

Order of elements is not guaranteed. If the zip code spans multiple counties, multiple row elements will be returned.

---

**You wont use the ones below, wanted you to see what sets would look li**

---

**setGeneralInfo**

Sets organization's general information. Access is via POST.

**Path:** ...ESD/{orgId}/General

**Parameters:**
orgId - (required in path) The organization's id. Use 0 to create a new organization.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
data - (required) String of information to be set. Data string must conform to the following specification:

- Each piece of information should be expressed as a tag/value pair in the format: tag=value
- tag/value pairs must be separated by the pipe (|) symbol
- tag/value pairs may appear in any order

**Returns:**

- If an new organization was indicated, return string will be an XML string containing a single element named data whose value
- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

**setTreatmentInfo**

Sets treatments offered by an organization. Access is via POST. This is a "replace" operation. All existing treatment information for

**Path:** ...ESD/{orgId}/Treatments

**Parameters:**
orgId - (required in path) The organization's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
data - (required) String of information to be set. Data string is made up of treatmentIds separated by a tilda (~).

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## setTrainingInfo

Sets training services offered by an organization. Access is via POST. This is a "replace" operation. All existing training informati

**Path:** ...ESD/{orgId}/Training

**Parameters:**
orgId - (required in path) The organization's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
data - (required) String of information to be set. Data string is made up of trainingIds separated by a tilda (~).

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## setEquipmentInfo

Sets equipment available from an organization. Access is via POST. This is a "replace" operation. All existing equipment information

**Path:** ...ESD/{orgId}/Equipment

**Parameters:**
orgId - (required in path) The organization's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
data - (required) String of information to be set. Data string must conform to the following specification:

- Each piece of information should be expressed as a tag/value pair in the format: tag=value
- tag/value pairs must be separated by the pipe (|) symbol
- tag/value pairs may appear in any order
- Multiple sets of information must be separated by a tilda (~).

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## setFacilitiesInfo

Sets available facilities at an organization. Access is via POST.  This is a "replace" operation. All existing facilities information

**Path:** ...ESD/{orgId}/Facilities

**Parameters:**
orgId - (required in path) The organization's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
data - (required) String of information to be set. Data string must conform to the following specification:

- Each piece of information should be expressed as a tag/value pair in the format: tag=value
- tag/value pairs must be separated by the pipe (|) symbol
- tag/value pairs may appear in any order
- Multiple sets of information must be separated by a tilda (~).

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## setLocationInfo

Sets location information for an organization. Access is via POST.  This is a "replace" operation. All existing information for this l

**Path:** ...ESD/{orgId}/Locations/{siteId}

**Parameters:**
orgId - (required in path) The organization's id.
siteId - (required in path) The locations's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
mode - (optional) Defines what should be done with this location. Possible values are Delete and Edit(default if not specified).
data - (required) String of information to be set. Data string must conform to the following specification:

- Each piece of information should be expressed as a tag/value pair in the format: tag=value
- tag/value pairs must be separated by the pipe (|) symbol
- tag/value pairs may appear in any order

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## setPeopleInfo

Sets information regrding the people at this site for the organization. Access is via POST.  This is a "replace" operation. All existi

**Path:** ...ESD/{orgId}/People/{siteId}

**Parameters:**
orgId - (required in path) The organization's id.
siteId - (required in path) The locations's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
mode - (optional) Defines what should be done with this group of people. Possible values are Delete (not implemented) and Edit(de
data - (required) String of information to be set. Data string must conform to the following specification:

- Each piece of information should be expressed as a tag/value pair in the format: tag=value
- tag/value pairs must be separated by the pipe (|) symbol

- tag/value pairs for the person's basic information may appear in any order
- tag/value pairs for contact methods must appear at the end of the record for any particular person
- Multiple records (persons) must be separated by a tilda (~)

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## insertPhysicianInfo

Sets information regarding the physicians that have admitting privleges for an organization. Access is via POST.  This is a "replace" (

**Path:** ...ESD/{orgId}/Physicians

**Parameters:**

orgId - (required in path) The organization's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
mode - (optional) Defines what should be done with this location. Possible values are Insert and Load.
csv - (optional) File name containing physician information for use when mode=load.
data - (optional) String of information containing physician information for use when mode=insert. Data string must conform to th

- Each piece of information should be expressed as a tag/value pair in the format: tag=value
- tag/value pairs must be separated by the pipe (|) symbol
- tag/value pairs may appear in any order

**Returns:**

- If information was set for an existing organization, return string will be an XML string containing a single element named s
- If the information could not be set for some reason, return string will be an XML string containing a single element named e

---

## setPhysicianInfo

Removes or adds a single physician from an organization. Access is via POST.

**Path:** ...ESD/{orgId}/Physicians/{selector}

**Parameters:**

orgId - (required in path) The organization's id.
token - (required) Security token obtained at login.
ip - (required) IP address (with periods) of the client machine. Must match the IP address used when obtaining the security token
mode - (required) Defines what should be done with this physician. Possible values are Insert and Delete.
selector - (required in path) If mode is Delete, this should be set to the personId. If mode is Insert, this should be set to the

**Returns:**

An XML string  as follows:

- If an error occured the string will consist of the single element "error" and a message.
- If mode is Delete, the string will consist of a single element "success" with a value of 1.
- If mode is Insert, the string will contain information about the physician formatted as follows:

```
<data count="1">
    <personId>
    <fName/>
    <mName/>
    <lName/>
    <suffix/>
    <license/>
    <phone/>
  </data>
```