

# Getting Started

## Making a REST API call

The simplest way to get started is by making an actual API call. For example, to list all the services on your Gateway:

<https://127.0.0.1:8443/restman/1.0/services>

The above link makes a REST API call to list all the services in the Gateway.

## The URL

All the available API calls have a similar URL format:

*https://[gateway-host]/[restman-service]/[api-version]/[api-resource]*

<b>[gateway-host]</b>	Host and port of the Gateway.
<b>[restman-service]</b>	Published URI of the REST management service. Default is "restman".
<b>[api-version]</b>	Version of the API that you are accessing; all API calls require an API version. Current version is 1.0.
<b>[api-resource]</b>	Resource being accessed. This can either map to an actual resource available on the Gateway, like a policy or JDBC connection, or it can reference an available service, like the migration service.

## The Response

All API calls will return the results wrapped. The wrapper will be either a `I7:List` or `I7:Item`. These are both based on the same underlying object and so they both will have:

<b>Name</b>	A human readable name describing the contents
<b>Type</b>	Type of the item that this contains
<b>TimeStamp</b>	When this item was created
<b>Link</b>	0 or more links relating to the containing object

The List wrapper will also contain 0 or more Items. These items are the listing results.

The Item wrapper will contain an ID element containing the ID of the contained object. The Item wrapper will also contain a single resource. This resource will be the object that this item wraps.

# Setting Up

## Set Up from the Policy Manager

To install the Gateway REST Management Service from the Policy Manager:

1. Select **Tasks** -> **Publish Internal Service** from the main menu (on the browser client, from the Manage menu). The Publish Internal Service Wizard appears.
2. Choose "Gateway REST Management Service" from the drop-down list as the service to publish.
3. Optionally, change the routing URI if necessary. Note that the routing URI should always end with a "/"
4. Click **[Finish]** to publish the internal service. This will create a new service in the root folder called "Gateway REST Management Service".

## Set Up Using Auto-Provisioning

The Gateway REST Management service can be installed automatically on a new Gateway using the auto-provisioning service.

A Gateway can automatically install an internal service on startup if it is licensed and there are no other services installed (as would be the case for a new Gateway). This is done by providing a file in a folder on the Gateway machine specifying the internal service to install. By default, running this command on the Gateway should set it up:

```
touch  
/opt/SecureSpan/Gateway/node/default/etc/bootstrap/services/restman
```

## Basic Entity Operations

Most of the entities support the same set of similar operations. These include the basic CRUD (Create, Read, Update and Delete) operations, Listing/Searching operations, and retrieving template (example) entities. The general use case for these operations is described below.

### Listing and Searching Entities

You can list all entities of a given type by making a GET request to that entity's URI. For example, to list all Folders on the Gateway, make a request to:

<https://127.0.0.1:8443/restman/1.0/folders>

*Listing Example*

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/folders">https://127.0.0.1:8443/restman/1.0/folders</a>
Http Method	GET

Response	
Http Status	200
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:List xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"&gt;   &lt;l7:Name&gt;FOLDER list&lt;/l7:Name&gt;   &lt;l7:Type&gt;List&lt;/l7:Type&gt;   &lt;l7:TimeStamp&gt;2014-04-24T15:27:30.386-07:00&lt;/l7:TimeStamp&gt;   &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders"/&gt;   &lt;l7:Link rel="template" uri="https://127.0.0.1:8443/restman/1.0/folders/te   &lt;l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/folders"/&gt;   &lt;l7:Item&gt;     &lt;l7:Name&gt;Root Node&lt;/l7:Name&gt;     &lt;l7:Id&gt;00000000000000000000000000000000&lt;/l7:Id&gt;     &lt;l7:Type&gt;FOLDER&lt;/l7:Type&gt;     &lt;l7:TimeStamp&gt;2014-04-24T15:27:30.384-07:00&lt;/l7:TimeStamp&gt;     &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders/00     &lt;l7:Resource&gt;       &lt;l7:Folder id="00000000000000000000000000000000" version="0"&gt;         &lt;l7:Name&gt;Root Node&lt;/l7:Name&gt;       &lt;/l7:Folder&gt;     &lt;/l7:Resource&gt;   &lt;/l7:Item&gt;   &lt;l7:Item&gt;     &lt;l7:Name&gt;My Services&lt;/l7:Name&gt;     &lt;l7:Id&gt;78457446b2cea4de5244873fe56a9f31&lt;/l7:Id&gt;     &lt;l7:Type&gt;FOLDER&lt;/l7:Type&gt;     &lt;l7:TimeStamp&gt;2014-04-24T15:27:30.385-07:00&lt;/l7:TimeStamp&gt;     &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders/78     &lt;l7:Resource&gt;       &lt;l7:Folder folderId="00000000000000000000000000000000" id="7845744         &lt;l7:Name&gt;test&lt;/l7:Name&gt;       &lt;/l7:Folder&gt;     &lt;/l7:Resource&gt;   &lt;/l7:Item&gt; &lt;/l7:List&gt;</pre>

The returned message above shows two folders:

- Root Node: This is the root folder.
- My Services

The listing API call can also be used for searching and sorting entities. For example, many entities have a name and in most cases you can find an entity by its name using the listing API call and providing a name parameter. Specifically, you can find the folder named "My Services" using the following API call:

#### Listing Example 2

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/folders?name=My%20Services">https://127.0.0.1:8443/restman/1.0/folders?name=My%20Services</a>
Http Method	GET
Response	

Http Status	200
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:List xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"&gt;   &lt;l7:Name&gt;FOLDER list&lt;/l7:Name&gt;   &lt;l7:Type&gt;List&lt;/l7:Type&gt;   &lt;l7:TimeStamp&gt;2014-04-24T15:27:30.386-07:00&lt;/l7:TimeStamp&gt;   &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders"/&gt;   &lt;l7:Link rel="template" uri="https://127.0.0.1:8443/restman/1.0/folders/te   &lt;l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/folders"/&gt;   &lt;l7:Item&gt;     &lt;l7:Name&gt;Root Node&lt;/l7:Name&gt;     &lt;l7:Id&gt;00000000000000000000000000000000&lt;/l7:Id&gt;     &lt;l7:Type&gt;FOLDER&lt;/l7:Type&gt;     &lt;l7:TimeStamp&gt;2014-04-24T15:27:30.384-07:00&lt;/l7:TimeStamp&gt;     &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders/00     &lt;l7:Resource&gt;       &lt;l7:Folder id="00000000000000000000000000000000" version="0"&gt;         &lt;l7:Name&gt;Root Node&lt;/l7:Name&gt;       &lt;/l7:Folder&gt;     &lt;/l7:Resource&gt;   &lt;/l7:Item&gt; &lt;/l7:List&gt;</pre>

The returned message is very similar to the one returned when listing all folders, however only the folder with the name "My Services" is returned. You can specify multiple search parameters to create more complex searches. For example, to list all enabled Gateway connectors using the HTTP or FTP protocols:

[https://127.0.0.1:8443/restman/1.0/ssgconnectors?  
enabled=true&protocol=HTTP&protocol=FTP](https://127.0.0.1:8443/restman/1.0/ssgconnectors?enabled=true&protocol=HTTP&protocol=FTP)

Listing results can also be sorted and ordered using the "sort" and "order" query parameters. Refer to the [WADL](#) or the [Resources](#) to find what you can sort and search by for an entity. For example, use this to sort all email listeners in descending order by their host name:

<https://127.0.0.1:8443/restman/1.0/emailListeners?sort=host&order=desc>

If the returned list does not contain any items, then none could be found with the specified search fields or you do not have the correct permissions to view the items. See <https://127.0.0.1:8443/restman/1.0/doc/authentication.html> for more information on authentication and permissions.

## Retrieving an Entity

An individual entity can be returned by using its ID. For example you can retrieve the secure password with ID "b0f109679d51ca3d3be3f08a13052f72" by making the following API call:

### Retrieving Example 1

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/passwords/b0f109679d51ca3d">https://127.0.0.1:8443/restman/1.0/passwords/b0f109679d51ca3d</a>
Http Method	GET
Response	
Http Status	200
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"&gt;   &lt;l7:Name&gt;MySecurePassword&lt;/l7:Name&gt;   &lt;l7:Id&gt;b0f109679d51ca3d3be3f08a13052f72&lt;/l7:Id&gt;   &lt;l7:Type&gt;SECURE_PASSWORD&lt;/l7:Type&gt;   &lt;l7:TimeStamp&gt;2014-04-24T16:12:44.383-07:00&lt;/l7:TimeStamp&gt;   &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/passwords/b0f1   &lt;l7:Link rel="template" uri="https://127.0.0.1:8443/restman/1.0/passwords/   &lt;l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/passwords"/&gt;   &lt;l7:Resource&gt;     &lt;l7:StoredPassword id="b0f109679d51ca3d3be3f08a13052f72" version="1"&gt;       &lt;l7:Name&gt;MySecurePassword&lt;/l7:Name&gt;       &lt;l7:Properties&gt;         &lt;l7:Property key="usageFromVariable"&gt;           &lt;l7:BooleanValue&gt;true&lt;/l7:BooleanValue&gt;         &lt;/l7:Property&gt;         &lt;l7:Property key="description"&gt;           &lt;l7:StringValue&gt;&lt;/l7:StringValue&gt;         &lt;/l7:Property&gt;         &lt;l7:Property key="lastUpdated"&gt;           &lt;l7&gt;DateValue&gt;2014-04-24T16:12:22.960-07:00&lt;/l7&gt;DateValue&gt;         &lt;/l7:Property&gt;         &lt;l7:Property key="type"&gt;           &lt;l7:StringValue&gt;Password&lt;/l7:StringValue&gt;         &lt;/l7:Property&gt;       &lt;/l7:Properties&gt;     &lt;/l7:StoredPassword&gt;   &lt;/l7:Resource&gt; &lt;/l7:Item&gt;</pre>

## Creating an Entity

In order to create an entity, send a POST request, with message body containing the entity description, to the entity's uri. All properties that are not in the entity description will be set to null. The following example shows how to create a new folder.

### Creating Example 1

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/folders">https://127.0.0.1:8443/restman/1.0/folders</a>
Http Method	POST
Content-Type	application/xml
Message Body	

<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:Folder xmlns:l7="http://ns.17tech.com/2010/04/gateway-management" folderId   &lt;l7:Name&gt;My New Folder&lt;/l7:Name&gt; &lt;/l7:Folder&gt;</pre>	
<b>Response</b>	
Http Status	201
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:Item xmlns:l7="http://ns.17tech.com/2010/04/gateway-management"&gt;   &lt;l7:Name&gt;My New Folder&lt;/l7:Name&gt;   &lt;l7:Id&gt;b0f109679d51ca3d3be3f08a1305300c&lt;/l7:Id&gt;   &lt;l7:Type&gt;FOLDER&lt;/l7:Type&gt;   &lt;l7:TimeStamp&gt;2014-04-24T16:24:48.353-07:00&lt;/l7:TimeStamp&gt;   &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders/b0f109   &lt;l7:Link rel="template" uri="https://127.0.0.1:8443/restman/1.0/folders/te   &lt;l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/folders"/&gt;   &lt;l7:Link rel="dependencies" uri="https://127.0.0.1:8443/restman/1.0/folder   &lt;l7:Link rel="parentFolder" uri="https://127.0.0.1:8443/restman/1.0/folder &lt;/l7:Item&gt;</pre>

You can create an entity with a specific ID by using the PUT HTTP method. See below for more details.

## Updating an Entity

In order to update an entity, send a PUT request with the entity description to the entity's uri. All properties that are not in the entity description will be set to null. For example, to update the folder created above and change its name:

### Updating Example

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/folders/b0f109679d51ca3d3be3f08a1305300c">https://127.0.0.1:8443/restman/1.0/folders/b0f109679d51ca3d3be3f08a1305300c</a>
Http Method	PUT
Content-Type	application/xml
Message Body	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <l7:Folder xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management" folderId="0000000000000000ffffffffffffec76" id="b0f109679d51ca3d3be3f08a1305300c" version="0"> <l7:Name>My Policies</l7:Name> </l7:Folder>
Response	
Http Status	200
Message Body	

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
  <l7:Name>My Policies</l7:Name>
  <l7:Id>b0f109679d51ca3d3be3f08a1305300c</l7:Id>
  <l7:Type>FOLDER</l7:Type>
  <l7:TimeStamp>2014-04-24T16:31:05.844-07:00</l7:TimeStamp>
  <l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders/b0f109
  <l7:Link rel="template" uri="https://127.0.0.1:8443/restman/1.0/folders/te
  <l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/folders"/>
  <l7:Link rel="dependencies" uri="https://127.0.0.1:8443/restman/1.0/folder
  <l7:Link rel="parentFolder" uri="https://127.0.0.1:8443/restman/1.0/folder
</l7:Item>
```

You can create an entity with a specific ID by sending a PUT request to that entity's URI. For example, to create a folder with the ID "b0f109679d51ca3d3be3f08a130530bf", send the following request:

### Creating Example 2

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/folders/b0f109679d51ca3d3be">https://127.0.0.1:8443/restman/1.0/folders/b0f109679d51ca3d3be</a>
Http Method	PUT
Content-Type	application/xml
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:Folder xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management" folderId   &lt;l7:Name&gt;My Encass Policies&lt;/l7:Name&gt; &lt;/l7:Folder&gt;</pre>
Response	
Http Status	201
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"&gt;   &lt;l7:Name&gt;My Encass Policies&lt;/l7:Name&gt;   &lt;l7:Id&gt;b0f109679d51ca3d3be3f08a130530bf&lt;/l7:Id&gt;   &lt;l7:Type&gt;FOLDER&lt;/l7:Type&gt;   &lt;l7:TimeStamp&gt;2014-04-24T16:34:48.521-07:00&lt;/l7:TimeStamp&gt;   &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/folders/b0f109   &lt;l7:Link rel="template" uri="https://127.0.0.1:8443/restman/1.0/folders/te   &lt;l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/folders"/&gt;   &lt;l7:Link rel="dependencies" uri="https://127.0.0.1:8443/restman/1.0/folder   &lt;l7:Link rel="parentFolder" uri="https://127.0.0.1:8443/restman/1.0/folder &lt;/l7:Item&gt;</pre>

Note that when updating an entity, all properties of that entity must be specified, even those that are not changing, otherwise they will be set to null.

## Deleting an Entity

An entity can be deleted by sending a DELETE request to that entity's URI. For example, to delete a policy with ID "b0f109679d51ca3d3be3f08a130531a0" send the following request:

### *Deleting Example*

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/policies/b0f109679d51ca3d3be3f08a130531a0">https://127.0.0.1:8443/restman/1.0/policies/b0f109679d51ca3d3be3f08a130531a0</a>
Http Method	DELETE
Response	
Http Status	204

## Retrieving an Entity Template

Most entities have a template uri that allows you to retrieve a template entity object. This can be used to then create a new entity. For example, the template for cluster properties can be retrieved as follows:

### *Template Example*

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/clusterProperties/template">https://127.0.0.1:8443/restman/1.0/clusterProperties/template</a>
Http Method	GET
Response	
Http Status	200
Message Body	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"&gt;   &lt;l7:Name&gt;CLUSTER_PROPERTY Template&lt;/l7:Name&gt;   &lt;l7:Type&gt;CLUSTER_PROPERTY&lt;/l7:Type&gt;   &lt;l7:TimeStamp&gt;2014-04-24T16:44:56.440-07:00&lt;/l7:TimeStamp&gt;   &lt;l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/clusterPropert   &lt;l7:Link rel="list" uri="https://127.0.0.1:8443/restman/1.0/clusterPropert   &lt;l7:Resource&gt;     &lt;l7:ClusterProperty&gt;       &lt;l7:Name&gt;Template Cluster Property Name&lt;/l7:Name&gt;       &lt;l7:Value&gt;Template Cluster Property Value&lt;/l7:Value&gt;     &lt;/l7:ClusterProperty&gt;   &lt;/l7:Resource&gt; &lt;/l7:Item&gt;</pre>

## Other Operations

### Dependency Analysis



Entities may depend on each other in many different ways. For example, a service may require a secure password and a JDBC connection.

You can view the dependencies for a service, folder, policy, or encapsulated assertion by using the dependency analysis sub-resource.

For example, to display all the dependencies for a service, make the following API call:

#### *Dependency Example*

Request	
Uri	<a href="https://127.0.0.1:8443/restman/1.0/services/b0f109679d51ca3d3b">https://127.0.0.1:8443/restman/1.0/services/b0f109679d51ca3d3b</a>
Http Method	GET
Response	
Http Status	200
Message Body	

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
  <l7:Name>MyService dependencies</l7:Name>
  <l7:Type>DEPENDENCY</l7:Type>
  <l7:TimeStamp>2014-04-24T16:59:14.373-07:00</l7:TimeStamp>
  <l7:Link rel="self" uri="https://127.0.0.1:8443/restman/1.0/services/b0f10">
  <l7:Resource>
    <l7:DependencyList>
      <l7:Reference>
        <l7:Name>LookupInDatabase</l7:Name>
        <l7:Id>b0f109679d51ca3d3be3f08a13053261</l7:Id>
        <l7:Type>SERVICE</l7:Type>
        <l7:Dependencies>
          <l7:Dependency>
            <l7:Name>jdbc1</l7:Name>
            <l7:Id>08d65a5a4449f13594a9379d1bb3820d</l7:Id>
            <l7:Type>JDBC_CONNECTION</l7:Type>
          </l7:Dependency>
        </l7:Dependencies>
      </l7:Reference>
      <l7:Dependencies>
        <l7:Dependency>
          <l7:Name>pass1</l7:Name>
          <l7:Id>08d65a5a4449f13594a9379d1bb38215</l7:Id>
          <l7:Type>SECURE_PASSWORD</l7:Type>
        </l7:Dependency>
        <l7:Dependency>
          <l7:Name>jdbc1</l7:Name>
          <l7:Id>08d65a5a4449f13594a9379d1bb3820d</l7:Id>
          <l7:Type>JDBC_CONNECTION</l7:Type>
          <l7:Dependencies>
            <l7:Dependency>
              <l7:Name>pass1</l7:Name>
              <l7:Id>08d65a5a4449f13594a9379d1bb38215</l7:Id>
              <l7:Type>SECURE_PASSWORD</l7:Type>
            </l7:Dependency>
          </l7:Dependencies>
        </l7:Dependency>
      </l7:Dependencies>
    </l7:DependencyList>
  </l7:Resource>
</l7:Item>
```



The Dependency list example above returns all the dependencies for the entity identified under "Reference". This reference contains:

<b>Name</b>	Human-readable name of the entity.
<b>Id</b>	The ID of the entity
<b>Type</b>	Type of the entity
<b>Dependencies</b>	A list of the direct dependencies for this entity

Following the Reference is a list of dependencies. This list includes all dependencies for the Reference, both direct and indirect. Each dependency contains:

<b>Name</b>	Human-readable name of the entity.
<b>Id</b>	The ID of the entity
<b>Type</b>	Type of the entity
<b>Dependencies</b>	A list of the direct dependencies for this entity

## Migration

The Migration resource can be used to migrate folders, services, and policies between Gateways. For more information about using the Migration resource see:

<https://127.0.0.1:8443/restman/1.0/doc/migration.html>

