

## 5 Publish REST Service

### 5.1 Description

This tutorial shows you how to publish a gateway REST service (or proxy) for the tutorial Warehouse REST service deployed in **Tutorial 1 - Deploy Tutorial Services** which in practice would be hosted on an application server behind the gateway. This is one of the most common tasks performed by users of our gateway, and it is the starting point for enforcing additional policy on traffic sent through the gateway to a REST service or API as demonstrated in other tutorials.

### 5.2 Prerequisites

#### 5.2.1 Environment

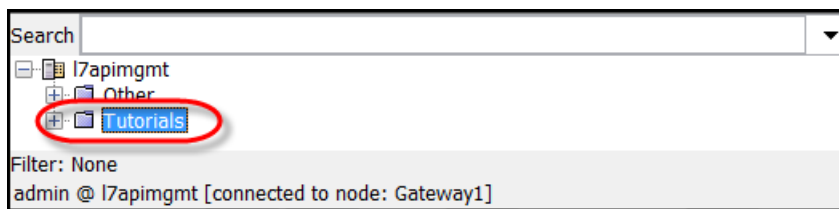
1. Layer 7 SecureSpan Gateway (*this tutorial was designed using a version 7.0 gateway; it may or may not work with earlier versions; it should work with later versions*)
2. Layer 7 Policy Manager (*this tutorial uses the Policy Manager software installation; the software installation version must match the gateway version; alternatively, users can use the Policy Manager browser-based version which always matches the gateway version that is connected to*)
3. soapUI (*this tutorial was designed using the free soapUI version 4.5.1; it may or may not work with other versions of soapUI; other clients can be used for this and other tutorials, but specific steps will not be provided for those other clients*)

#### 5.2.2 Tutorials

1. Layer 7 Tutorials - Getting Started
2. Tutorial 1 - Deploy Tutorial Services
3. Tutorial 3 - Test Tutorial REST Service

### 5.3 Tutorial Steps

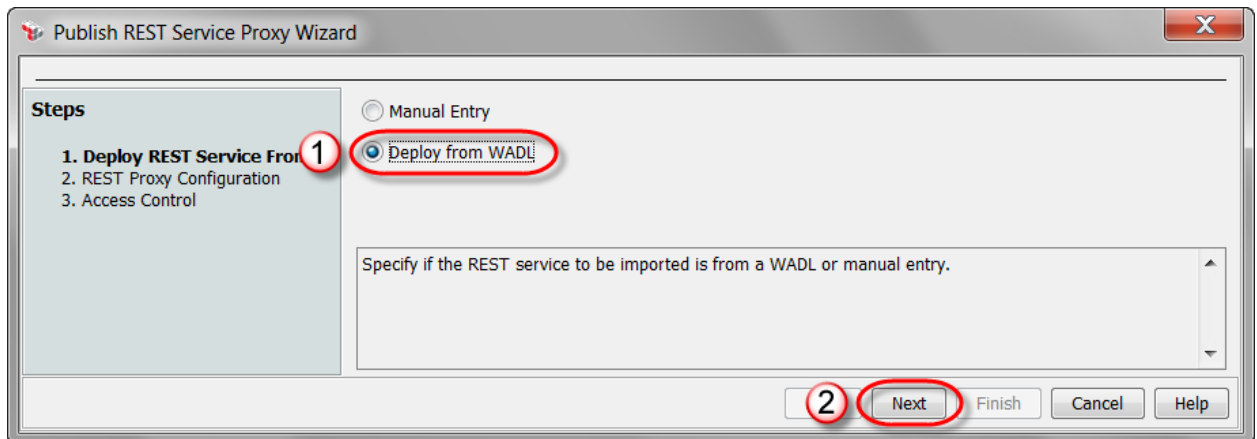
1. Connect to your gateway using Policy Manager (see tutorial **Layer 7 Tutorials - Getting Started**).
2. In the service and policy tree, right click on the new **Tutorials** folder.



3. In the context menu, select the **Publish RESTful Service Proxy** context menu item.

*Note: You can also launch the Publish RESTful Service Proxy wizard from the Policy Manager Home screen and the Tasks menu. However, when starting from those locations, published services are located at the root level of the service and policy tree. When starting from the context menu of a folder in the tree, published services are located in that folder.*

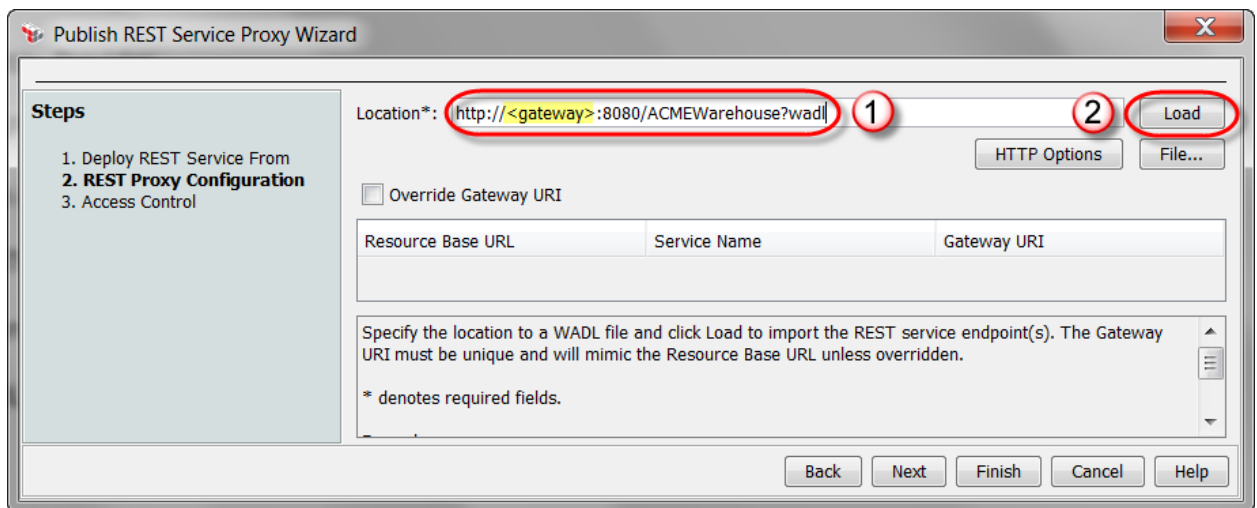
4. In the Publish REST Service Proxy Wizard, select the **Deploy from WADL** option, and click the **Next** button.



5. In the Publish REST Service Proxy Wizard dialog, copy this URL into the Location field:

**http://<gateway>:8080/ACMEWarehouse?wadl**

Replace <gateway> with the host name or IP address of your gateway, and click the **Load** button.



6. In the Publish REST Service Proxy Wizard dialog, select the **Override Gateway URI** option, highlight the current service name and type **Warehouse REST Tutorials**, highlight the current gateway URI and type **gateway**, and then click the **Next** button.

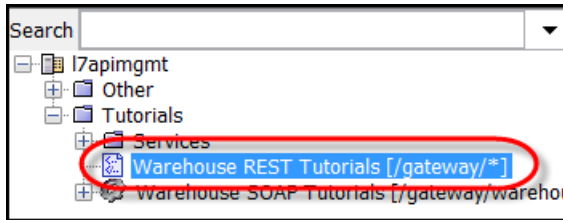
The screenshot shows the 'Publish REST Service Proxy Wizard' dialog, Step 2: REST Proxy Configuration. The 'Steps' pane on the left lists: 1. Deploy REST Service From, 2. REST Proxy Configuration (highlighted with a red circle and '1'), and 3. Access Control. The 'Location\*' field contains 'http://l7apimgmt:8080/ACMEWarehouse?wadl'. The 'Override Gateway URI' checkbox is checked (highlighted with a red circle and '1'). Below this is a table with three columns: 'Resource Base URL', 'Service Name', and 'Gateway URI'. The 'Resource Base URL' is 'http://l7apimgmt:8080/ACME' (highlighted with a red circle and '2'), the 'Service Name' is 'Warehouse REST Tutorials' (highlighted with a red circle and '2'), and the 'Gateway URI' is 'gateway' (highlighted with a red circle and '3'). Below the table is a text box with instructions: 'Specify the location to a WADL file and click Load to import the REST service endpoint(s). The Gateway URI must be unique and will mimic the Resource Base URL unless overridden.' and '\* denotes required fields.' At the bottom, the 'Next' button is highlighted with a red circle and '4'. Other buttons include 'Load', 'HTTP Options', 'File...', 'Finish', 'Cancel', and 'Help'.

7. In the Publish REST Service Proxy Wizard dialog, click the **Finish** button.

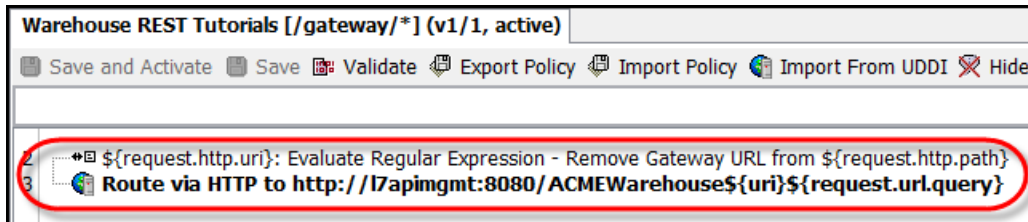
The screenshot shows the 'Publish REST Service Proxy Wizard' dialog, Step 3: Access Control. The 'Steps' pane on the left lists: 1. Deploy REST Service From, 2. REST Proxy Configuration, and 3. Access Control (highlighted with a red circle and '1'). The 'Require SSL/TLS Encryption' checkbox is unchecked. The 'Allow Anonymous Access' radio button is selected. The 'Require Users to Authenticate' radio button is unselected. Below this are dropdowns for 'Authentication Method' (set to 'HTTP Basic') and 'Identity Provider' (set to 'Internal Identity Provider'). There are two list boxes: 'No Permission' and 'Have Permission', both empty. Between them are buttons: 'Add', 'Add All', 'Remove', and 'Remove All'. Below the list boxes is a text box with instructions: 'Specify non-SOAP application access security and permissions.' At the bottom, the 'Finish' button is highlighted with a red circle. Other buttons include 'Back', 'Next', 'Cancel', and 'Help'.

*Note: This step would allow you to select some easy to configure options to require a secure transport (i.e. SSL/TLS) and/or some form of identification, authentication and authorization. Ultimately, selections made here result in more initial policy assertions added to the policy of the service being published. This tutorial will skip this step, and other tutorials will show you how to require these things in many different ways later in policy.*

8. In the service and policy tree, under the Tutorials folder, you should now see a REST service named Warehouse REST Tutorials.

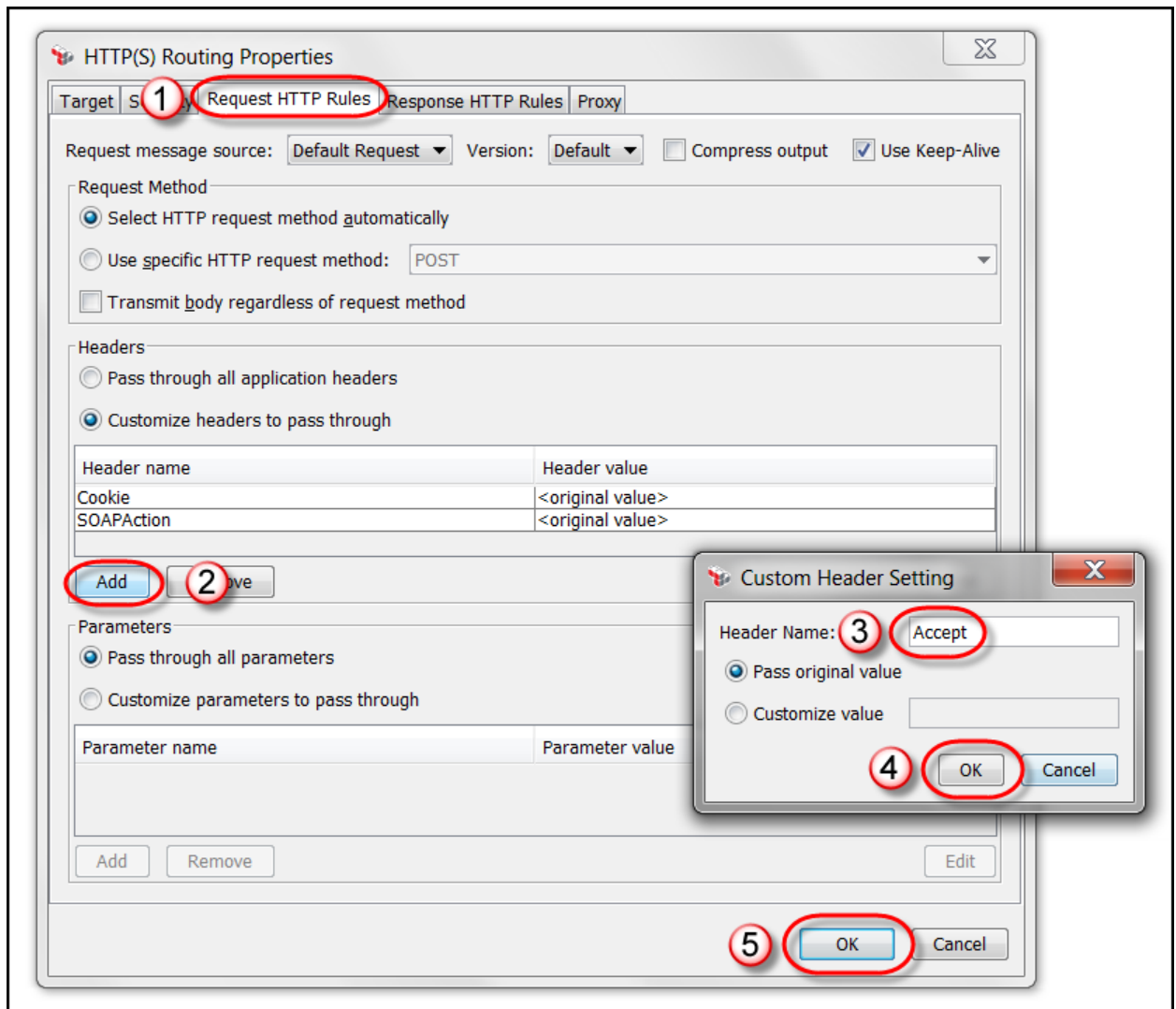


9. In the policy editor, notice the **Evaluate Regular Expression** and the **Route via HTTP** policy assertion added to the active policy of the new service by the wizard.

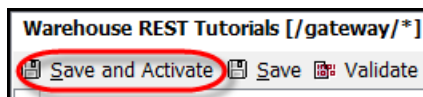


10. To allow the consumer to request a JSON response, we need to allow the HTTP Accept header through the gateway. Currently, the HTTP Accept header is not one of the HTTP headers that we pass through by default. To allow the HTTP Accept header, double-click on **assertion #3, the Route via HTTP assertion**.

11. In the HTTP(S) Routing Properties dialog, select the **Request HTTP Rules** tab. Under the Headers frame, click the **Add** button. In the Custom Header Setting dialog, in the Header Name field, type **Accept**, and click the **OK** button. In the HTTP(S) Routing Properties dialog, click the **OK** button.



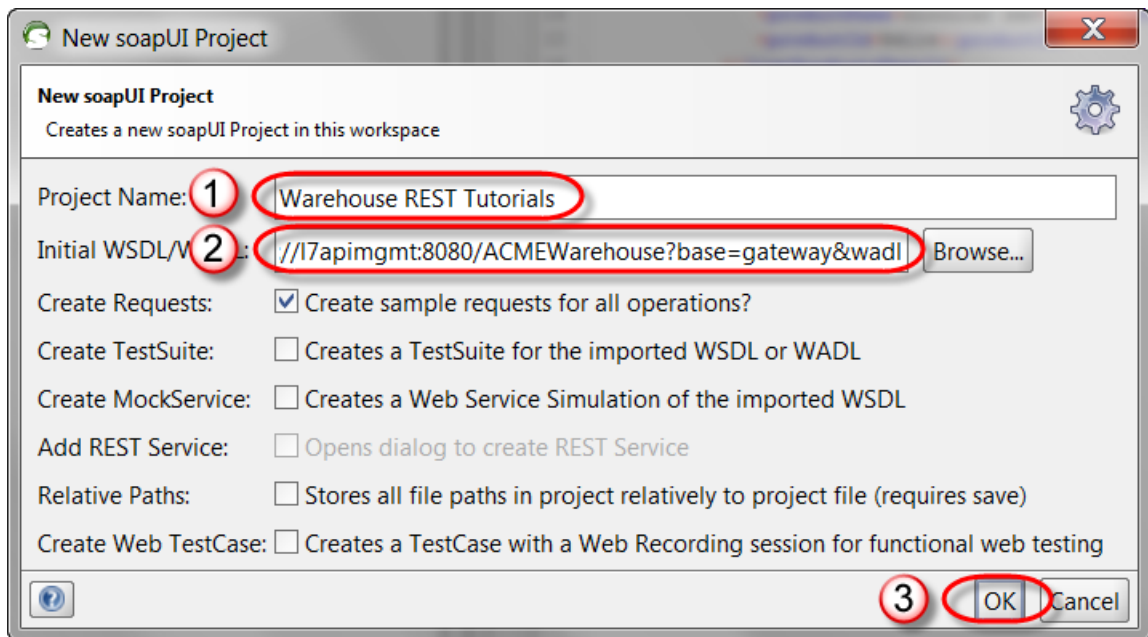
12. On the policy editor toolbar, click the **Save and Activate** button.



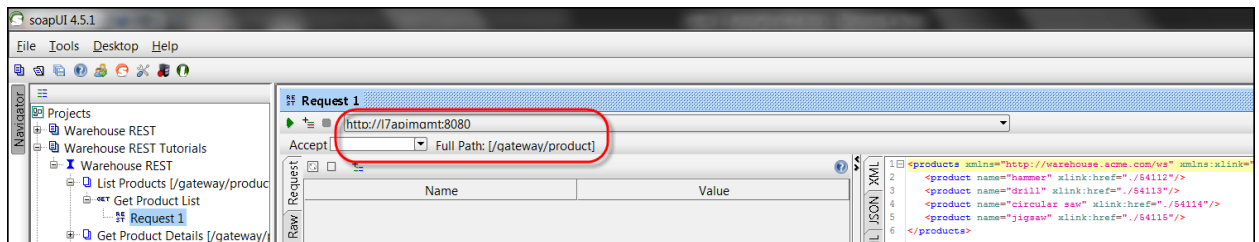
13. Test the new Warehouse REST Tutorials service by creating a new soapUI project as demonstrated in **Tutorial 3 - Test Tutorial REST Service**. This time, when following that tutorial to create a new project you'll want to call it **Warehouse REST Tutorials** and use the following WADL URL (with <gateway> replaced by the host name or IP address of your gateway):

**http://<gateway>:8080/ACMEWarehouse?base=gateway&wadl**

For example:



And:



14. Per **Layer 7 Tutorials - Getting Started/Basic Policy Concepts/Policy Authoring/Policy Revisions**, and as demonstrated at the end of **Tutorial 1 - Deploy Tutorial Services**, comment the active policy revision of the **Warehouse REST Tutorials** service with the comment, **Tutorial 5 Complete**.
15. You are done with this tutorial.

## 5.4 Additional Context

In their current state, the Warehouse REST Tutorials service and policy are referred to as a pass through proxy and a pass through policy respectively. In other words, they're simply passing through request and response messages. They're not enforcing any more interesting policy (besides the URI transformation which is an aspect of service virtualization that happens to some degree when fronting any service with a gateway).

However, Layer 7 customers already gain some value from pass through services. Namely, pass through services virtualize the actual service input on a secure gateway (possibly deployed in the DMZ) while providing some automatic network layer and message layer protection (without requiring additional policy assertions).

Also, once traffic is routed through a pass through proxy on the gateway, it's easy to begin enforcing additional value added policies upon that traffic as demonstrated by other tutorials in this document.