FORUM SYSTEMS HANDS-ON TRAINING

LAB 9. INTRODUCTION TO TASK LISTS

Published: September 2014
Forum Systems Hands-on Training – Lab 9. Introduction to Task Lists
D-ASF-SE-010029

Contents

Introduction

Lab 9. Introduction to Task Lists

Skill Level

This lab is Intermediate Skill Level. Some existing experience with Forum Sentry and SOAPSonar or completion of the Beginner Level labs is assumed.

Prerequisites

This lab requires the Forum Sentry Training Image, with a licensed copy of SOAPSonar Enterprise Edition.

Refer to the "FS_Training_Labs_v8-1_Introduction" document for information on the Forum Sentry Training Image and licensing SOAPSonar Enterprise Edition.

Part of this lab will utilize the OpenWeatherMap REST Policy built in Lab 6. This lab will apply conditional processing (content based routing) to this REST Policy using a task list.

Lab Overview

This lab focuses on building Task Lists and Task List Groups in Sentry. Task Lists are used to process request and response data that is brokered by Sentry. Task Lists are used for security processing, integration (data and attribute mapping), and identity management / access control.

There are many different things you can do with Task Lists, some of the common task operations (not a full list) are below:

- Conditional Identification
- Add, strip, or modify HTTP headers
- Change the HTTP method
- Add, modify, or remove XML nodes
- Convert data from JSON to/from XML, XML to/from SOAP, etc.
- Base64 or URL encode and decode data
- Modify and map query parameter values
- Create, filter on, and inject User Attribute values
- Content based routing
- WS Security – Encryption, Digital Signatures, WS Tokens
- User Identity and Access Control – SAML, OAuth, custom
- Database queries
- Custom Logging

This lab will provide instructions for generating and testing Task Lists with Forum Sentry. Topics include:

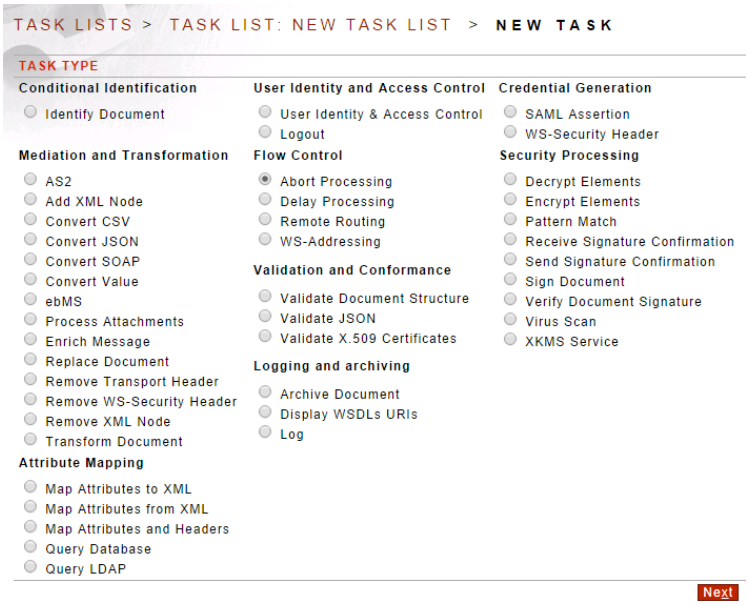
1. Types of Tasks
2. Building and Copying Task Lists
3. Sample Documents
4. Task List Errors
5. Task List Groups
6. Conditional Task List Groups – The Identify Document Task
7. Content Based Routing – The Remote Routing Task

Task Lists

Task Lists provide a comprehensive set of document processing rules that can be created to map, transform, identify, and otherwise manipulate transactions.

Types of Tasks

Task lists are comprised of individual tasks that perform processing on a request or response. The tasks in the list are processed in order starting with the first task in the list. A task list can have one or many tasks.



Tasks are grouped into Categories. The 8 categories are:

1. Conditional Identification – Identify Document, used to match a request or response based on a variety of criteria, including: header value, parameter value, XPath expression, attribute value, source IP, etc. Once the request or response is matched, subsequent tasks in the list can process accordingly. The Identify Document task is special in that it is used as a trigger for conditional processing. This is a very common use case.
2. Mediation and Transformation – Several tasks for manipulating data, including: add/remove XML node, remove HTTP header, XSLT transformation, convert JSON to/from XML/SOAP, encode and decode data, call another web service for enrichment, etc.
3. Attribute Mapping – Several tasks for mapping data to/from: headers, xml documents, parameters, databases, LDAP servers, user attributes, etc.
4. User Identity and Access Control – Tasks for consuming credentials, enabling OAuth, enabling SAML SSO, logging out session cookies, etc.
5. Flow Control – Tasks for controlling the traffic flow, including: setting and following WS Addressing headers, content based routing, abort processing, etc.
6. Validation – Tasks for XML and JSON schema validation.
7. Credential Generation – Tasks to generate WS Security headers/tokens and SAML Assertions.
8. Security Processing – Tasks for encryption, signatures, pattern matching, virus scanning, etc.

Sample Documents

Tasks that process SOAP/XML data often require an XPath expression and so require a Sample Document that is used to build the XPath expression.

First Example: If the requirement is to ensure that a specific element in an incoming SOAP request exists, a sample of the expected SOAP message will need to be selected as the Sample Document, so that an XPath expression can be used in the task.

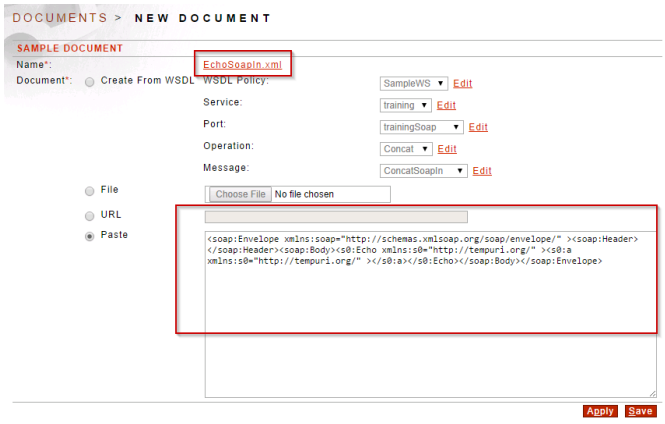
Second Example: If the requirement is to map a URI query parameter value from the incoming request into an XML document that Sentry will be sending forward, a sample of the XML Sentry is mapping the parameter value into is required as a sample as an XPath expression will be used in the mapping task.

Sentry includes several Documents that are representative of basic SOAP and WS Trust messages. See the image below for the default Documents.

Documents can be created from a loaded WSDL file, imported from file or URL, or manually entered via copy/paste or direct input.

Follow the instructions below to create a new Document in Sentry via file import.

1. Navigate to the Resources→Documents→Documents page. Click New.
2. Select File and browse to the Samples folder of the Desktop on the Sentry Training Image.
3. Select the file named “EchoSoapIn.xml”.
4. The name of the document will be filled in automatically to match the imported file. Click Apply to import the document.



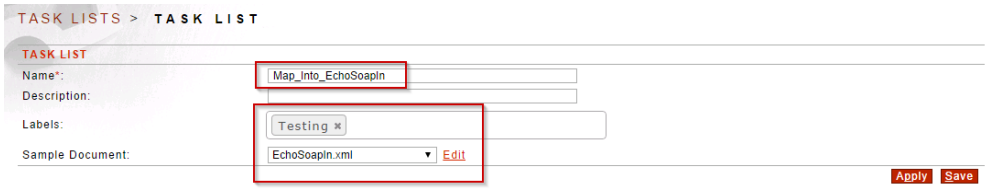
Building a Task List

This step will build a simple task list that maps a constant value into an XML document. It will not require sending a request into Sentry. Instead we will accomplish this by building the task list and running it within Sentry.

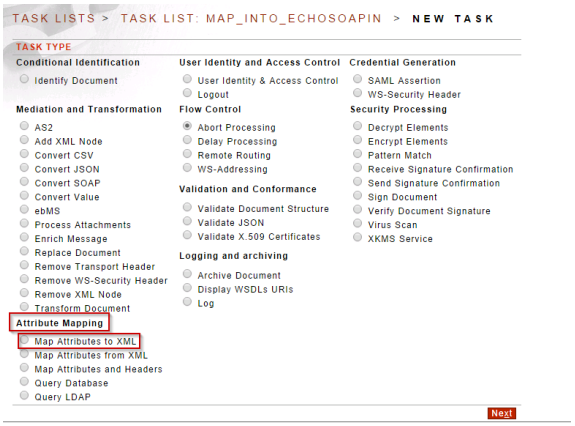
***Hint** – when working with Task Lists, if you click the Apply button it will save the Task List and leave you on the same page. If you click Save, it will save the list and bring you back out to the Task Lists screen. Typically using Apply is recommended so you can review the saved changes.*

Follow the steps below to build a new task list.

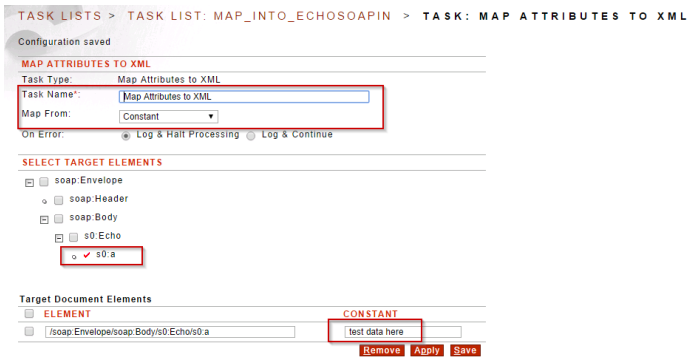
1. Navigate to the Gateway→Task Policies→Task Lists page. Click New.
2. On the following screen, enter the following information:
    a. Name: Map_Into_EchoSoapIn
    b. Description: leave blank
    c. Labels: Testing
    d. Sample Document: select the EchoSoapIn.xml file created earlier in this lab.



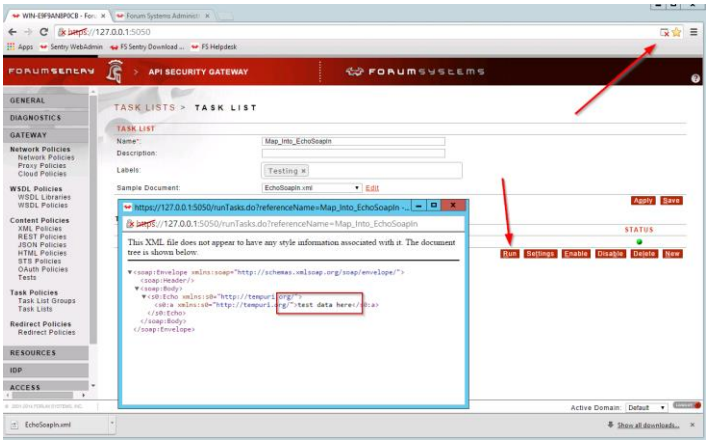
    e. Click Apply to build the Task List and stay on the same page (within the Task List). If you click Save, you will build the Task List and be returned to the main Task List page.
3. Click New.
4. Under the Attribute Mapping category, select “Map Attributes to XML”.



5. On the following page, make the changes listed below:
    a. Map From: Constant
    b. Select Target Elements: select the bottom node of the document
    c. Click Apply
    d. A new XPath expression will show up, with a Constant field populated with “a”
    e. Change the Constant value from “a” to “test data here”
    f. Click Apply



6. Click Save to exit the Task List.
7. You have now built a simple Task List that will map a constant value to an XML document using an XPath expression.
8. To test the Task List, click the Run button.
9. A new browser window should pop-up (ensure pop-ups are allowed) showing you the EchoSoapIn.xml with the constant value “test data here” inside the s0:a element.



Task List Errors

When a new task is configured, all preceding tasks in the Task List are run in the background against the Sample Document. This is the default behavior. If a failure in one of the preceding tasks is encountered, an error is displayed when building the new task, and the new task may not save.

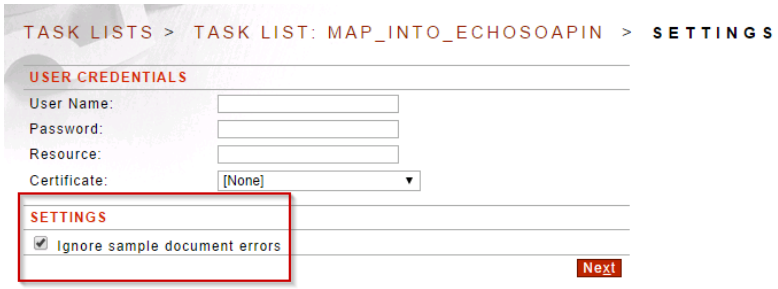
Running of the Task List against the Sample Document does not need to succeed in order for the Task List to be valid for runtime messages.

First Example: The first task in the list is Verify Document Signature, but the signature in the Sample Document is not valid - perhaps because it is expired or it uses a different key. So this Verify Document task will fail when run against the Sample Document. When you build a second task, you will see the signature verification failure listed in red at the top of the screen.

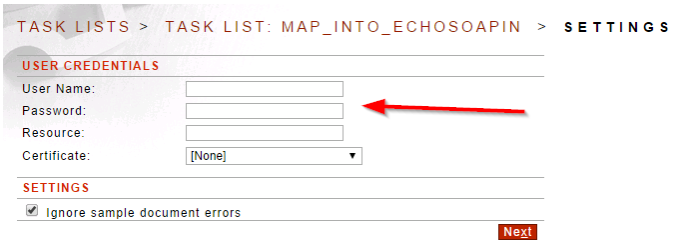
Second Example: There is a User Identity and Access Control task in the list and you receive a "User not identified" error at the top of the screen when building a subsequent task list.

Solutions

To disable the background task processing while building task lists, click the Settings button on the task list and enable the "Ignore sample document errors" option.



For failures related to missing user credentials or a missing certificate on the Task List Settings page, you can enter the credentials or browse to a cert in Sentry.



Lastly, you can disable the preceding tasks in the list so that they are not run when you are building the next task. However, this is only viable if the preceding task is not required for some processing.

For example, if the first task is decryption and the second task is to map from the decrypted document, you need the decryption task to succeed first. For this example, if it is not possible to get the decryption working (because you don't have the key), disable the decryption task and load a different Sample Document (a decrypted version) so you can build the next task. When you are all finished building all tasks, enable all that should be enabled for runtime.

The Run option on the task list will run all tasks in the list against the Sample Document in order. It is not required that this succeed in order for proper runtime message processing. It is normal and common that the Run will fail because the Sample Document is invalid, while actual runtime messages are processed successfully.

Task List Groups

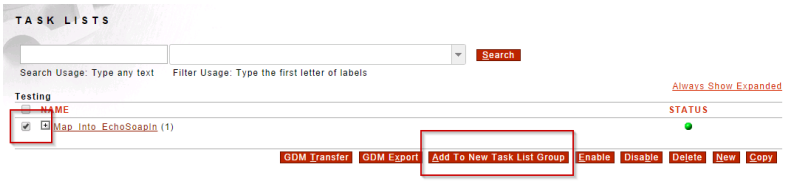
Task Lists are not associated directly to the runtime policies in Sentry. Instead, they are first associated to Task List Groups, which are then tied to the runtime policies (WSDL or Content Policies).

There are System Task List Groups for both requests and responses. These are global (system wide) Task List Groups and the tasks within these will run on all WSDL and Content Policies defined in Sentry.

Follow the steps below to add the "Map_into_EchoSoapIn" Task List to a Task List Group.

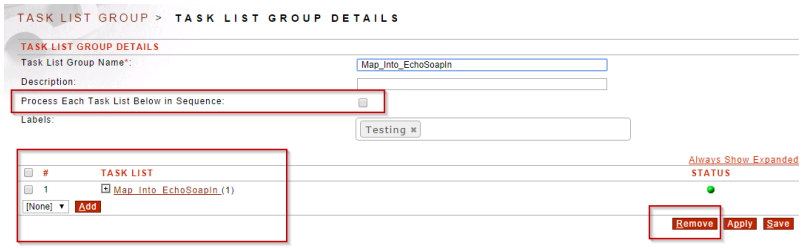
1. Navigate to the Gateway→Task Policies→Task Lists page.
2. Select (check the box next to) the "Map_Into_EchoSoapIn" Task List.

3. Click “Add to New Task List Group”



4. This will generate a new Task List Group that contains the Task List and brings you to the Task List Groups page. The Label will remain from the Task List.

5. Open the new Task List Group to view the settings.
    a. You can modify the name
    b. You can add a description
    c. Process Each Task List Below in Sequence – This is disable by default, so that the default behavior is conditional processing – start with the first Task List in the group and stop at the first that matches and runs. Enable this if you want all Task Lists in the group to run in order.
    d. Labels are set from the Task List but can be modified.
    e. You can add/remove existing Task Lists to/from this Task List Group.



Conditional Identification – Identify Document and Remote Routing Tasks

The extra step of adding Task Lists to Groups allows for conditional “if / then” processing. For instance, the requirement may be to do content based routing for an API. A common example of conditional processing is content based routing:

If the “server” query parameter value is “1”, then route to server 1.
OR
If the “server” query parameter value is “2”, then route to server 2.

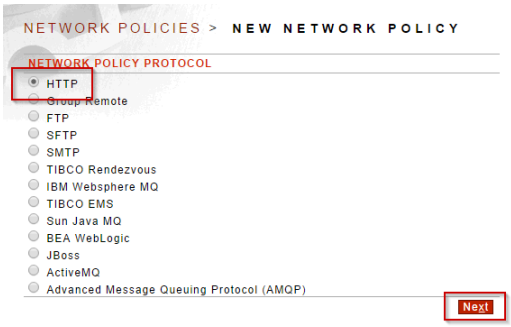
The Identify Document task is used to match a request or response based on a variety of criteria, as outlined earlier in this lab. Once the request or response is matched (the task succeeds), subsequent

tasks in the list can process accordingly. In other words, the Identify Document task is the IF part in the "if / then" processing.

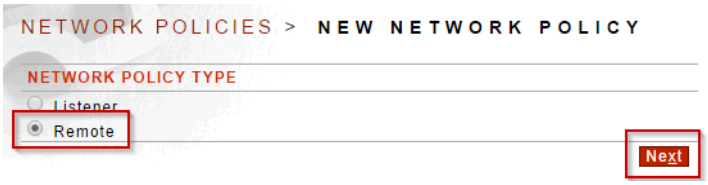
In this step, we will build a content based routing use case, which will utilize two Task Lists that identify and route to different servers. Both Task Lists will be in the same Task List Group.

Follow the instructions below to build a conditional Task List Group with content based routing.

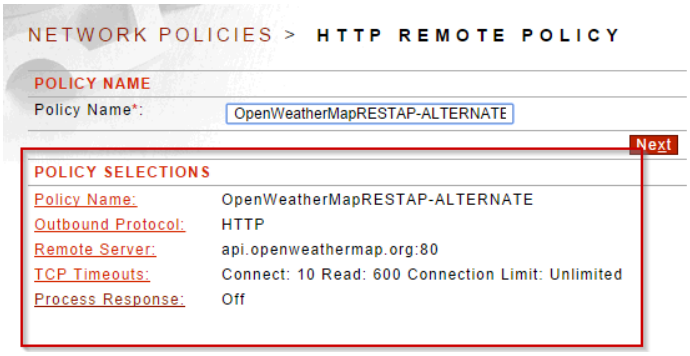
1. These instructions will build upon the OpenWeatherMap REST Policy built in Lab 6. If this REST Policy does not currently exist, build it or import an FSG for this policy.
2. Navigate to the Gateway→Network Policies→Network Policies page. Click New.
3. Create a new HTTP Remote Policy:
    a. Select HTTP and click Next



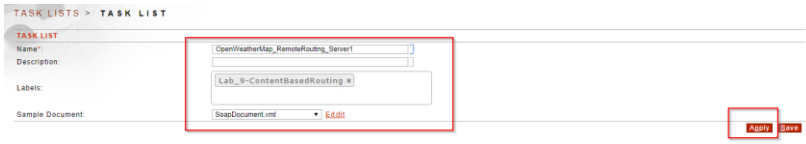
    b. Select Remote and click Next



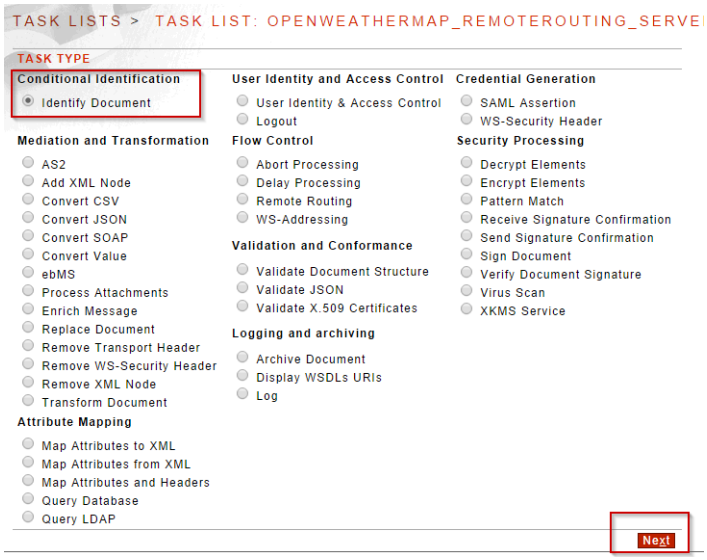
4. Build the HTTP Remote policy with the following criteria:
    a. Name: OpenWeatherMapRESTAPI-ALTERNATE
    b. Outbound Protocol: HTTP
    c. Remote Server: api.openweathermap.org
    d. Remote Port: 80
    e. TCP Timeouts: Default
    f. Process Response: Off
    g. If the option is not listed, leave the default value



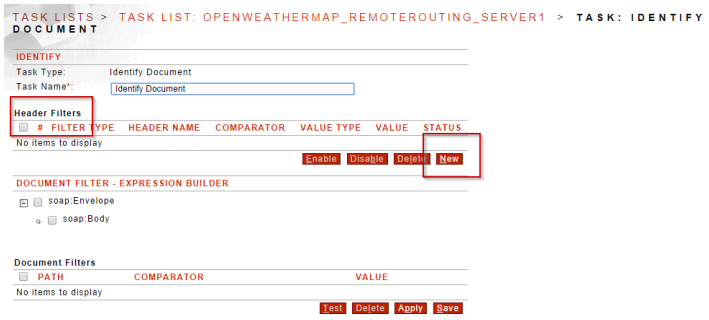
5. Navigate to the Gateway→Task Policies→Task Lists page and click New.
6. Create a new Task List with the following criteria:
    a. Name: OpenWeatherMap_RemoteRouting_Server1
    b. Description can be left empty
    c. Label: Lab_9-ContentBasedRouting
    d. Click Apply



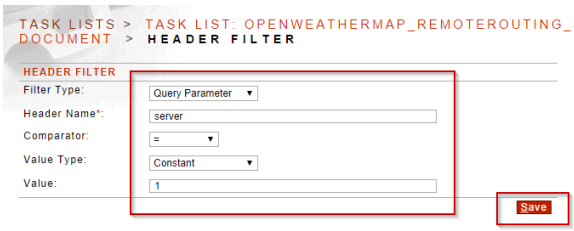
7. Click New to add a new task.
8. Select the Identify Document task under Conditional Identification and click Next.



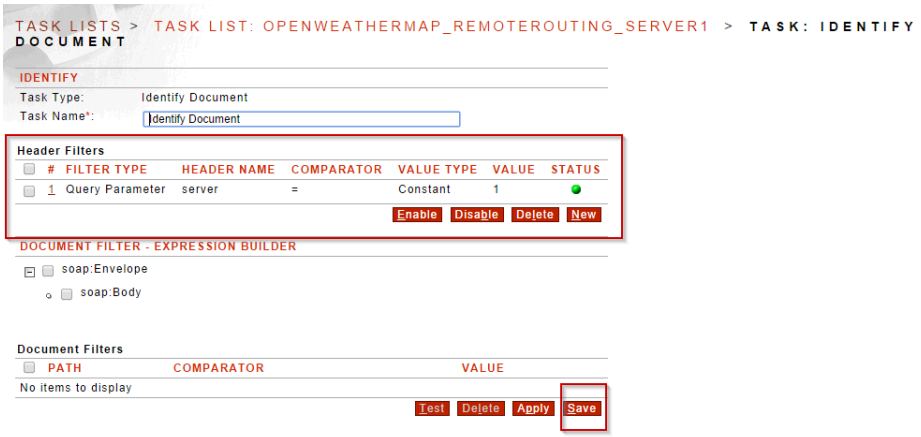
9. We will build a new Header Filter for this task to match a query parameter value. Click New.



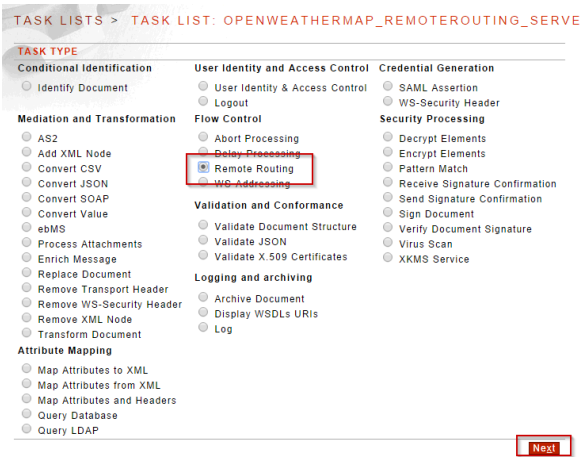
10. On the next screen, build the filter with the following criteria:
    a. Filter Type: Query Parameter
    b. Header Name: server
    c. Comparator: =
    d. Value Type: Constant
    e. Value: 1
    f. Click Save



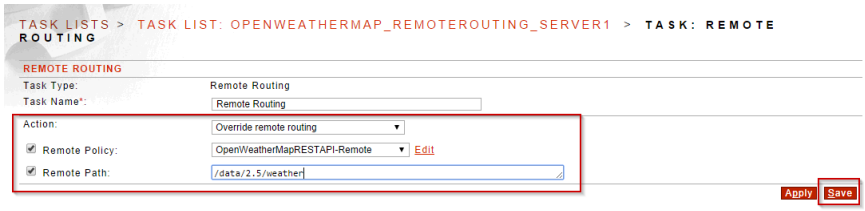
g. Click Save again



11. Click New to create another task in the same Task List.
12. Under the Flow Control category, select the Remote Routing task and click Next.

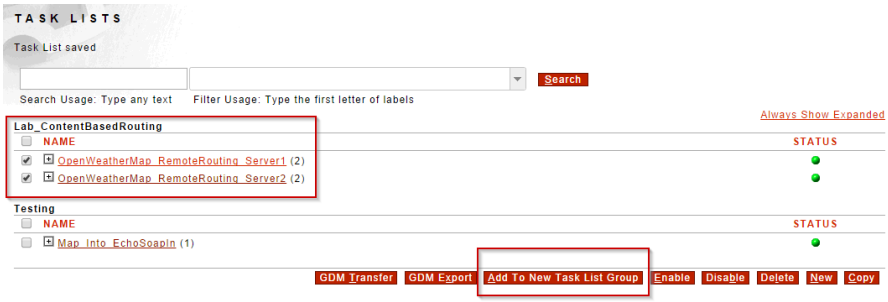


13. Configure the Remote Routing task as follows:
    a. Action: Override Remote Routing
    b. Remote Policy: OpenWeatherMapRESTAPI-Remote
    c. Remote Path: /data/2.5/weather
    d. Click Save

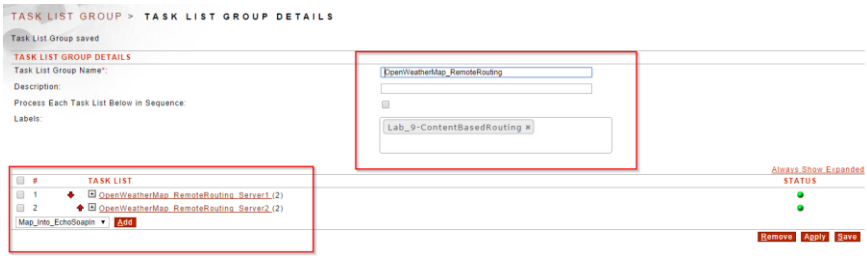


14. Save the Task List.
15. Make a copy of the Task List. Select the Task List just created and click the Copy button.

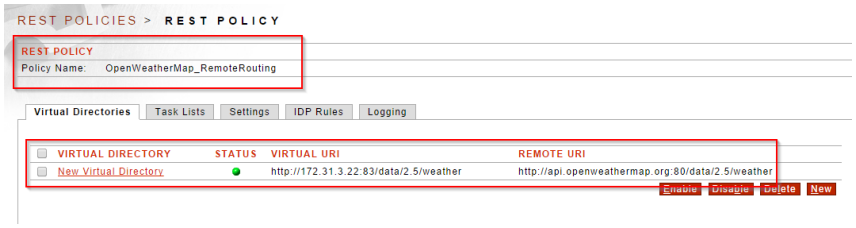
16. This will create a second Task List named “OpenWeatherMap_RemoteRouting_Server1_copy”.
17. Open the copied task list and make the following changes:
    a. Rename the Task List to: “OpenWeatherMap_RemoteRouting_Server2”
    b. Click Apply
    c. Modify the Identify Document task, changing the header filter to match the parameter value of 2
    d. Modify the Remote Routing task, change the Remote Policy to “OpenWeatherMapRESTAPI-ALTERNATE”
    e. Save the Task List
18. The next step to add both Task Lists to a new Task List Group. Select both Task Lists and click the “Add to new Task List Group” button.



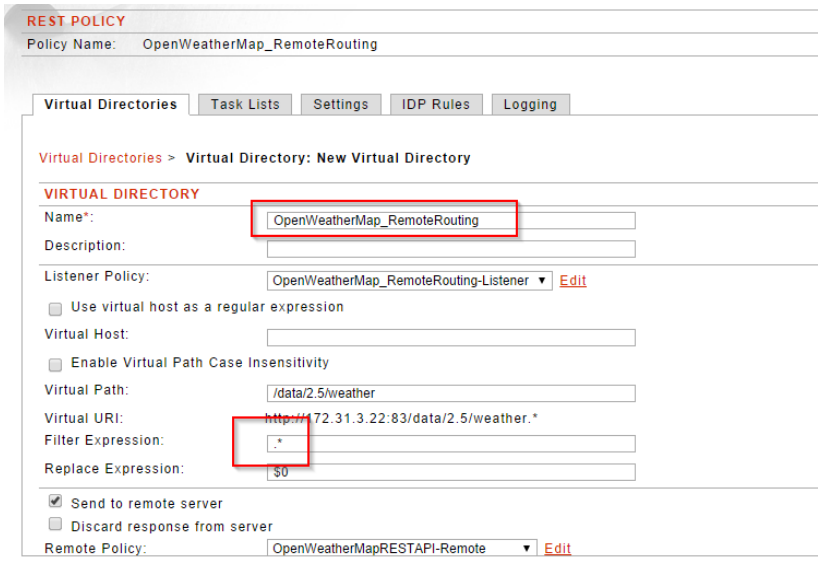
19. Review the Task List Group, which now has 2 Task Lists associated. This Task List Group is conditional because the “Process each.. in order” is not enabled. Change the name of the Task List Group to “OpenWeatherMap_RemoteRouting”.



20. Now we will associate the new Task List Group to a new REST policy for the OpenWeather Map API. Follow the instructions in Lab 6 of this series to build a new REST policy for the OpenWeather Map API. Use the following criteria
    a. Name: OpenWeatherMap_RemoteRouting
    b. Label: OpenWeatherMap
    c. Create a new listener policy, use port 83, enable the “Use Device IP” option
    d. Virtual Directory Path: /data/2.5/weather
    e. Select the existing HTTP Remote Policy named “OpenWeatherMapRESTAPI-Remote”
    f. The new REST Policy should look as follows:



    g. Go to the Virtual Directory of the new REST policy and make the following changes:
        i. Name: OpenWeatherMap_RemoteRouting
        ii. Filter Expression: .*
        iii. Save



21. Associate the remote routing Task List Group to the new REST Policy. Open the new REST Policy and click on the Task Lists tab. Select the “OpenWeatherMap_RemoteRouting” Task List Group as the Request Task List Group. Click Save.

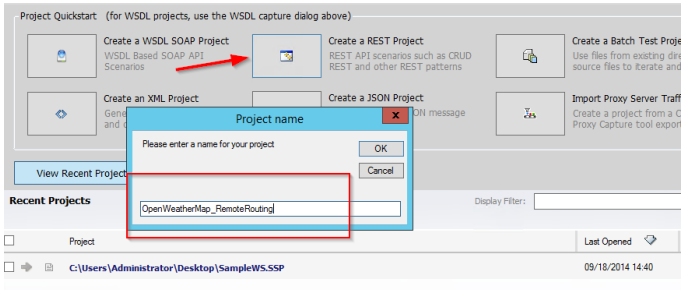
Testing the Content Based Routing Tasks with SOAPSonar

In this step we will test the new OpenWeatherMap_RemoteRouting REST Policy in Sentry. This testing is very similar to that performed in Lab 6.

Note that both remote policies in Sentry point to the same server, so to validate the Task Lists are working correctly, review the Sentry System log at DEBUG level.

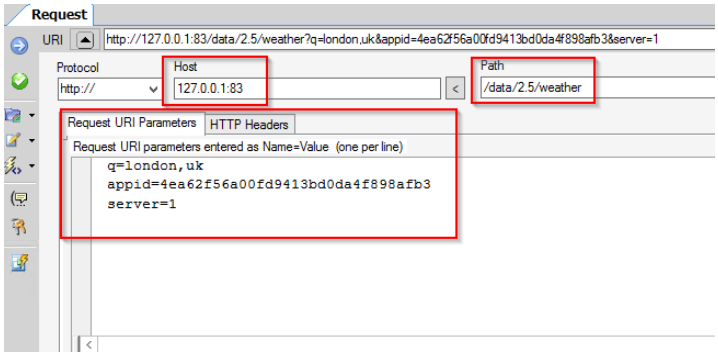
Follow the steps below to test OpenWeatherMap_RemoteRouting REST Policy using SOAPSonar.

1. Launch SOAPSonar and build a new REST Project.
    a. In the Project QuickStart menu choose Create a REST Project
    b. Name the project "OpenWeatherMap_RemoteRouting"



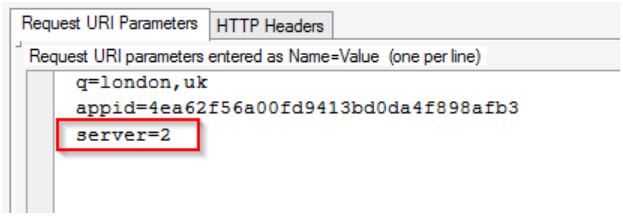
2. Build the REST Test case in SOAPSonar using the information listed below. You'll notice the Request URI field will update automatically.
    a. **Protocol:** HTTP
    b. **Host:** 127.0.0.1:83 (you can also specify the IP or hostname of Sentry listener)
    c. **Path:** /data/2.5/weather
    d. **Request URI Parameters / Name=Value pairs (one per line):**

        q=London,uk
        appid= 4eab62f56a00fd9413bd0da4f898afb3
        server=1

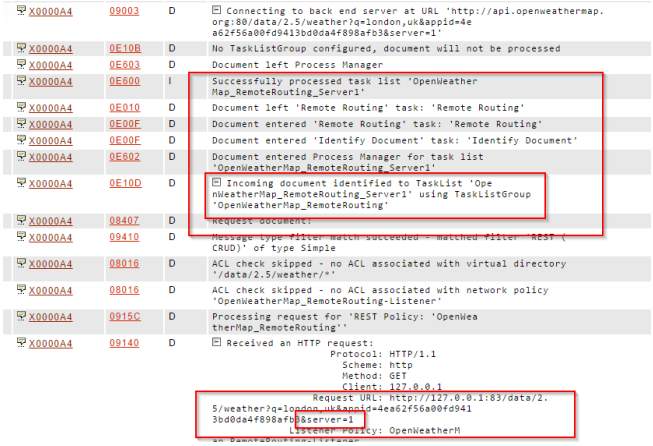


3. Click the  to commit the settings, and send the request using the  .
   You will receive a JSON format response with the weather data requested for the region entered.

4. Right click on the REST test case and Clone the test.
5. In the Clone test case, modify the Server URI parameter to value of 2.



6. Open the Sentry Access Log and review the last 2 transactions. Notice:
   When server=1 Sentry processes “OpenWeatherMap_RemoteRouting_Server1”
   When server=2 Sentry processes “OpenWeatherMap_RemoteRouting_Server2”



END

Additional Testing and More Reading

BACK IT UP!

It is recommended that you export your REST Policy and/or your full Sentry configuration after completing this lab. To export the REST policy, navigate to the REST Policies page, select the REST policy and use the GDM Export option to export the policy (and all dependencies) as a password encrypted FSG file. This can later be imported on the System→Configuration→Import/Export screen.

To export your full Sentry configuration, navigate to the System→Configuration→Import/Export screen and use the Export option in the center of the page to export the full Sentry configuration file as a password encrypted FSX file. This can later be imported on the same screen.

Backup your SOAPSonar project file by using the File→Save As option. All of your test cases will be saved in an .SSP file.

Additional Tests

1. What happens if you use server=3?
2. What happens if you leave off the server parameter?
3. Add a Map Attributes and Headers Task to add a custom HTTP Header to the request.

Additional Information

For more information on Task Lists see the following Helpdesk FAQs:
https://helpdesk.forumsys.com.com/entries/95448796-FAQ-How-to-Workaround-Design-Time-Task-List-Errors
https://helpdesk.forumsys.com/entries/70324767-How-To-Configure-Identify-Document-Task-to-Match-Multiple-Values
https://helpdesk.forumsys.com/entries/39364583-How-To-Configuring-Case-Sensitivity-with-the-Identify-Document-Task-in-Forum-Sentry
https://helpdesk.forumsys.com/entries/76439018-How-To-Modifying-or-Filtering-on-XML-Element-Attribute-Values

For more information, review the following Forum Sentry Admin Guide:

1. Network Policies Guide
2. Task Management Guide

About Forum Systems

Forum Systems is the global leader in API and Cloud Security technology with industry-certified, patented, and proven products deployed in the most rigorous and demanding customer environments worldwide. Forum Systems has been an industry leader for over 12 years and has built the core architecture of its technology on the foundation of FIPS 140-2 and NDPP. Forum Systems security-first mindset enables trusted, network edge deployments of its technology for protecting critical enterprise transactions.

Our product technology is purpose-built and designed for mission-critical, enterprise-class scalable solutions where business solutions require the modern day security and identity enforcement protection, while enabling a scalable architecture and low-latency, high-volume throughput.

Forum Systems supports global enterprise customers across industries in commercial, government, and military sectors. Forum Systems technology provides the leading-edge of modern-day cyber-security innovation with integrated identity and SSO features that enable out-of-the box business solutions with point-and-click technology.

Forum's patented; FIPS 140-2 and NDPP certified hardware and virtual products make modern-day business communications secure by actively protecting and accelerating data exchange and API service access across networks and business boundaries. For more information, please visit www.forumsys.com.