7 Basic Troubleshooting

7.1 Description

This tutorial provides a basic recommended approach to troubleshooting runtime policy behavior on a gateway.

Note: This tutorial assumes that you're gateway is currently configured with default auditing and logging settings. Thought this tutorial should be informative in almost any case, you may have unexpected results if your gateway is not currently configured with default auditing and logging settings.

7.2 Prerequisites

7.2.1 Environment

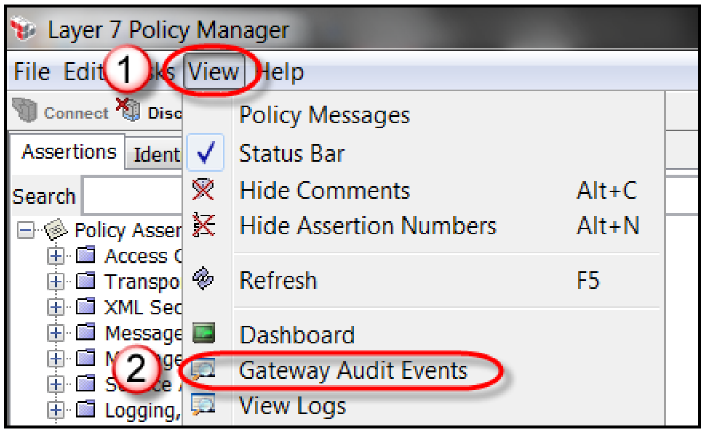
1. Layer 7 SecureSpan Gateway *(this tutorial was designed using a version 7.0 gateway; it may or may not work with earlier versions; it should work with later versions)*
2. Layer 7 Policy Manager *(this tutorial uses the Policy Manager software installation; the software installation version must match 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)*

7.2.2 Tutorials

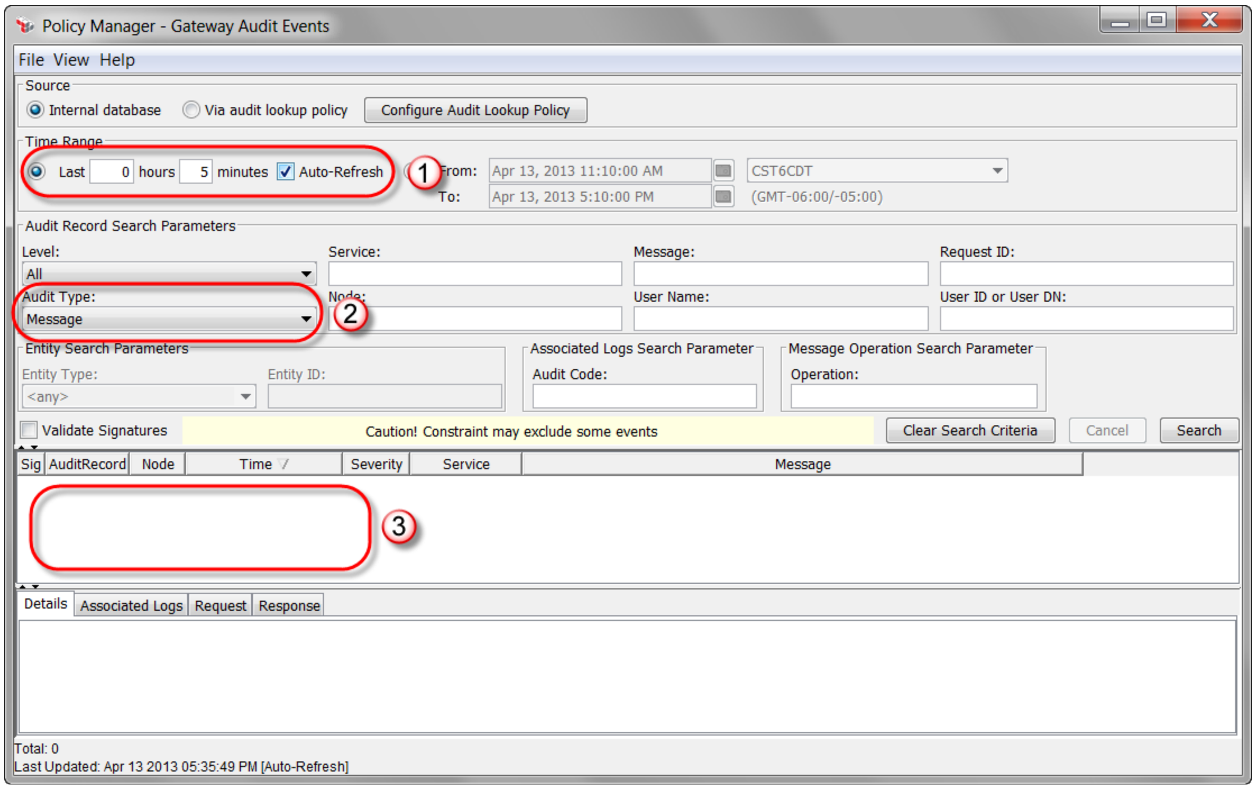
1. Layer 7 Tutorials - Getting Started
2. Tutorial 1 - Deploy Tutorial Services
3. Tutorial 3 - Test Tutorial REST Service
4. Tutorial 5 - Publish REST Service
5. Tutorial 6 - Basic Authentication

7.3 Tutorial Steps

1. Connect to your gateway using Policy Manager (see tutorial ***Layer 7 Tutorials - Getting Started***).
2. Per ***Layer 7 Tutorials - Getting Started /Basic Policy Concepts/Policy Authoring/Policy Revisions***, set the active policy version of the ***Warehouse REST Tutorials*** service to the version that has been commented with, **Tutorial 6 Complete**.
3. Select the ***View/Gateway Audit Events*** menu item.



4. In the Policy Manager - Gateway Audit Events dialog, in the Time Range frame, select the first option. Type ***0*** in the hours field and ***5*** in the minutes field. Check the ***Auto-Refresh*** option. In the Audit Record Search Parameters frame, for Audit Type, select ***Message***. You will probably not see any audit records at this time.


Policy Manager - Gateway Audit Events

File View Help

Source: (● Internal database) (Via audit lookup policy) Configure Audit Lookup Policy

Time Range: ● Last 0 hours 5 minutes ✓ Auto-Refresh
From: Apr 13, 2013 11:10:00 AM CST6CDT
To: Apr 13, 2013 5:10:00 PM (GMT-06:00/-05:00)

Audit Record Search Parameters
Level: All
Service:
Message:
Request ID:
Audit Type: Message
Node:
User Name:
User ID or User DN:

Entity Search Parameters
Entity Type: <any>
Entity ID:

Associated Logs Search Parameter
Audit Code:

Message Operation Search Parameter
Operation:

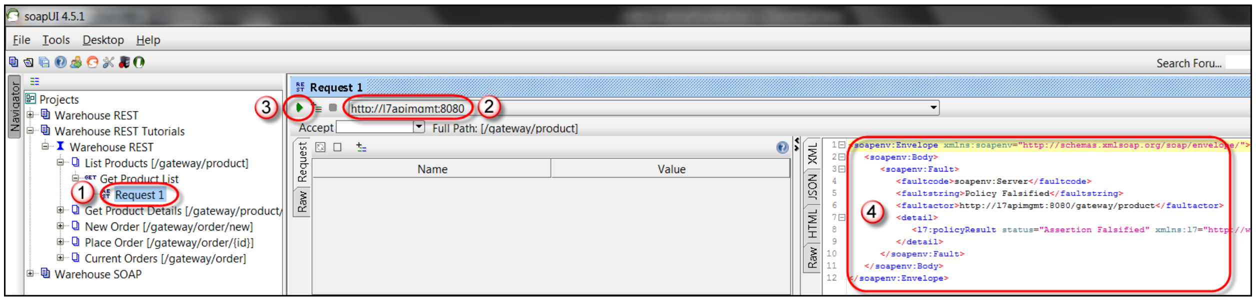
Validate Signatures
Caution! Constraint may exclude some events
Clear Search Criteria Cancel Search

Sig	AuditRecord	Node	Time	Severity	Service	Message

Details Associated Logs Request Response

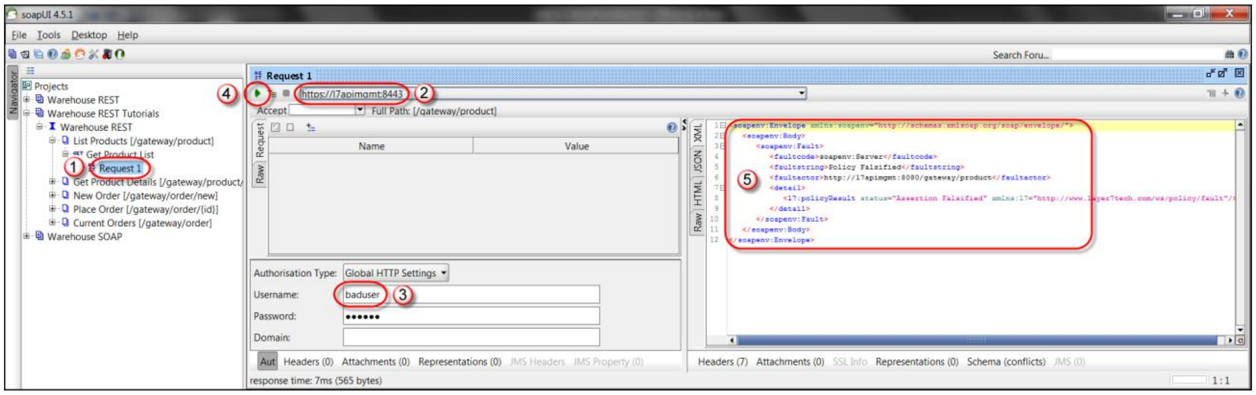
Total: 0
Last Updated: Apr 13 2013 05:35:49 PM [Auto-Refresh]

5. Open soapUI, and use the ***Warehouse REST Tutorials*** project created in the ***Tutorial 5 - Publish REST Service*** tutorial to test this tutorial's policy.
6. Send a Get Product List request to the HTTP (not HTTPS) endpoint and notice the SOAP fault response.


soapUI 4.5.1
File Tools Desktop Help
Projects
Warehouse REST
Warehouse REST Tutorials
Warehouse REST
List Products [/gateway/product]
Get Product List
Request 1
Get Product Details [/gateway/product/
New Order [/gateway/order/new]
Place Order [/gateway/order/{id}]
Current Orders [/gateway/order]
Warehouse SOAP
Request 1
http://l7apimgmt:8080
Accept
Full Path: [/gateway/product]
Name
Value
1
2
3
4

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>Policy Falsified</faultstring>
      <faultactor>http://l7apimgmt:8080/gateway/product</faultactor>
      <detail>
        <l7:policyResult status="Assertion Falsified" xmlns:l7="http://w
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

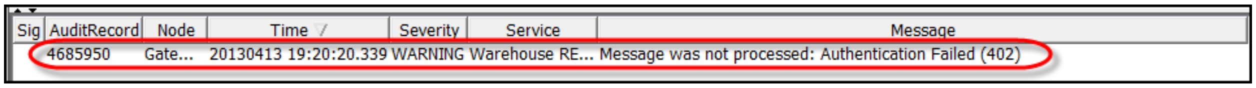
7. Now, go back and check the Policy Manager - Gateway Audit Events dialog. You should still not see any audit records. Some fault conditions do not rise to the level that a message audit record is generated. For example, someone trying to connect via HTTP to a service that requires HTTPS is an example of this.
8. Return to soapUI, and send a Get Product List request to the HTTPS endpoint with a bad user name and notice the different SOAP fault response.


soapUI 4.5.1
File Tools Desktop Help
Search Foru...
Projects
Warehouse REST
Warehouse REST Tutorials
Warehouse REST
List Products [/gateway/product]
Get Product List
Request 1
Get Product Details [/gateway/product/
New Order [/gateway/order/new]
Place Order [/gateway/order/{id}]
Current Orders [/gateway/order]
Warehouse SOAP
Request 1
https://l7apimgmt:8443
Accept
Full Path: [/gateway/product]
Name
Value
Aut Headers (0) Attachments (0) Representations (0) JMS Headers JMS Property (0)
Headers (7) Attachments (0) SSL Info Representations (0) Schema (conflicts) JMS (0)
response time: 7ms (565 bytes)
1:1
1
2
3
4
5

Authorisation Type: Global HTTP Settings
Username: baduser
Password: ••••••
Domain:

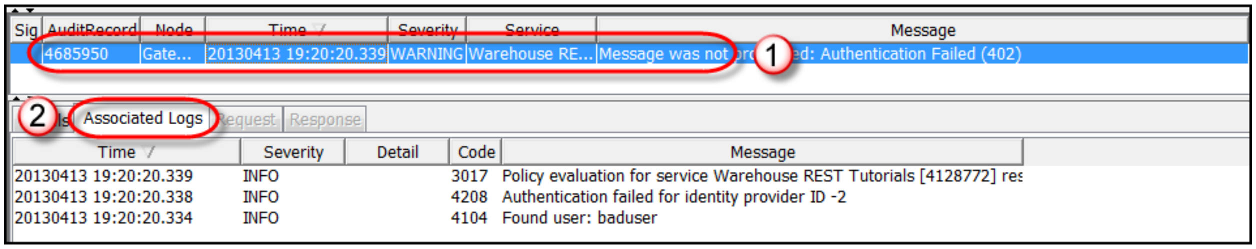
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>Policy Falsified</faultstring>
      <faultactor>http://l7apimgmt:8080/gateway/product</faultactor>
      <detail>
        <l7:policyResult status="Assertion Falsified" xmlns:l7="http://www.layer7tech.com/ws/policy/fault"/>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

9. Now, go back and check the Policy Manager - Gateway Audit Events dialog. This time you should see a message audit record. An authentication failure does rise to the level that a message audit record is generated.



Sig	AuditRecord	Node	Time	Severity	Service	Message
	4685950	Gate...	20130413 19:20:20.339	WARNING	Warehouse RE...	Message was not processed: Authentication Failed (402)

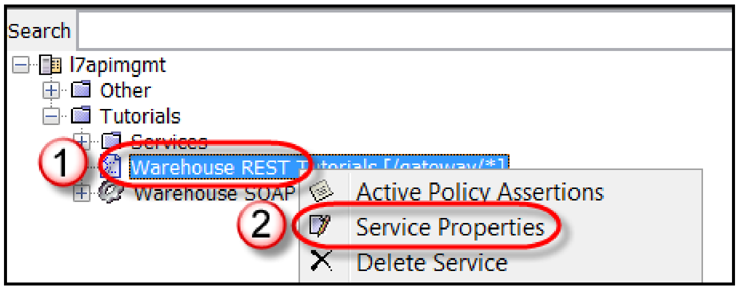
10. Select that audit message record in the list and click on the ***Associated Logs*** tab below. You see some useful but limited information. By default, an audit record will only include associated logs that are generated by the gateway as part of its normal message processing at a configurable level (the default is INFO) or above. Also notice that the ***Request*** and ***Response*** tabs are disabled. These tabs can show the outbound request (the request sent to the backend) and the outbound response (the response returned to the consumer). By default, the gateway does not save request or response messages with audit records.



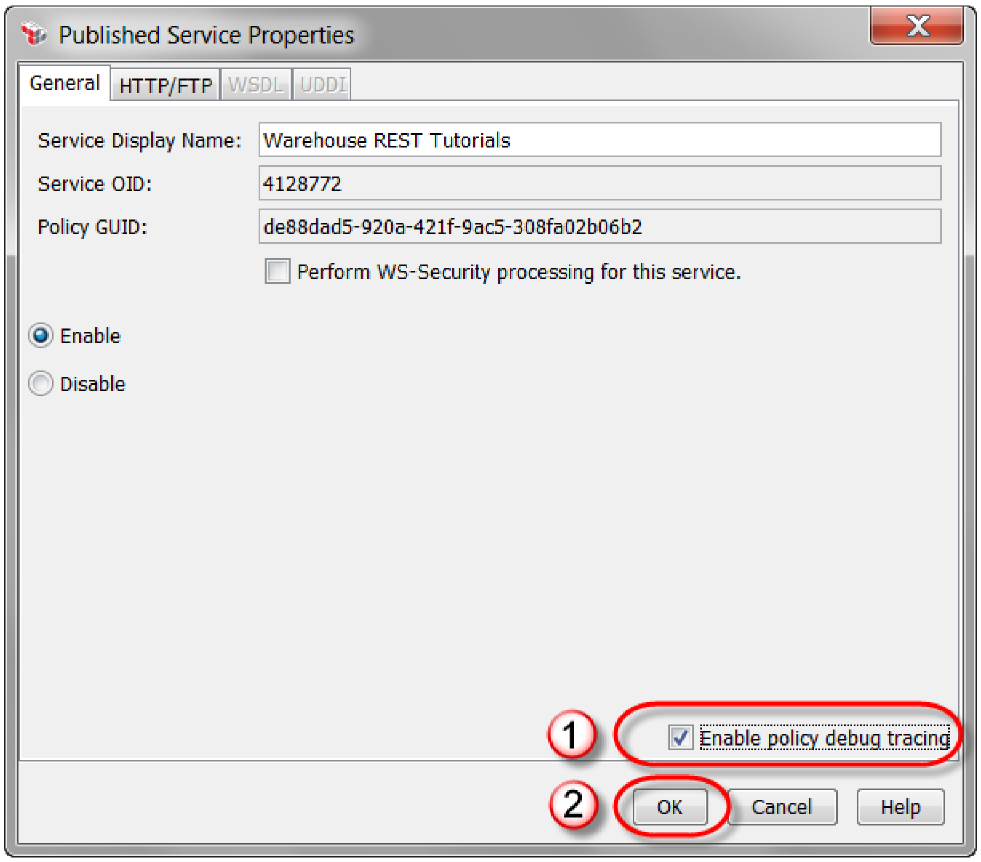
Sig	AuditRecord	Node	Time	Severity	Service	Message
	4685950	Gate...	20130413 19:20:20.339	WARNING	Warehouse RE...	Message was not pro... ed: Authentication Failed (402)

Time	Severity	Detail	Code	Message
20130413 19:20:20.339	INFO		3017	Policy evaluation for service Warehouse REST Tutorials [4128772] res
20130413 19:20:20.338	INFO		4208	Authentication failed for identity provider ID -2
20130413 19:20:20.334	INFO		4104	Found user: baduser

11. We will now enable debug trace to create more verbose audit records for all messages processed by a given service regardless of their success or failure. In Policy Manager, in the services and policies tree, right click on the ***Warehouse REST Tutorials*** service, and in the context menu, select the ***Service Properties*** menu item.



12. In the Published Service Properties dialog, select the ***Enable policy debug tracing option***, and click the ***OK*** button.


Published Service Properties

General | HTTP/FTP | WSDL | UDDI

Service Display Name: Warehouse REST Tutorials
Service OID: 4128772
Policy GUID: de88dad5-920a-421f-9ac5-308fa02b06b2

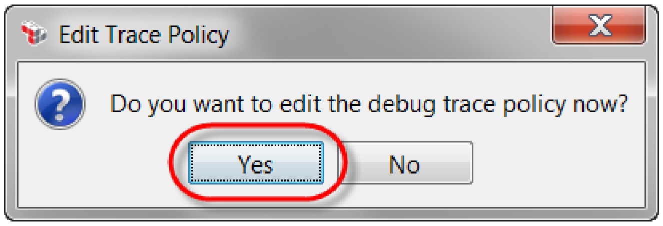
☐ Perform WS-Security processing for this service.

● Enable
○ Disable

☑ Enable policy debug tracing

OK | Cancel | Help

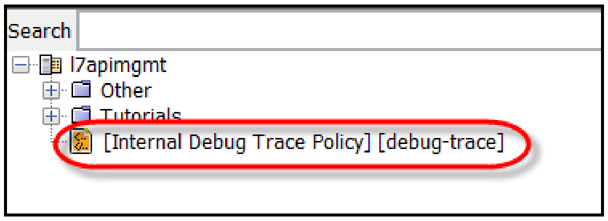
13. In the Edit Trace Policy dialog, click the ***Yes*** button.


Edit Trace Policy

Do you want to edit the debug trace policy now?

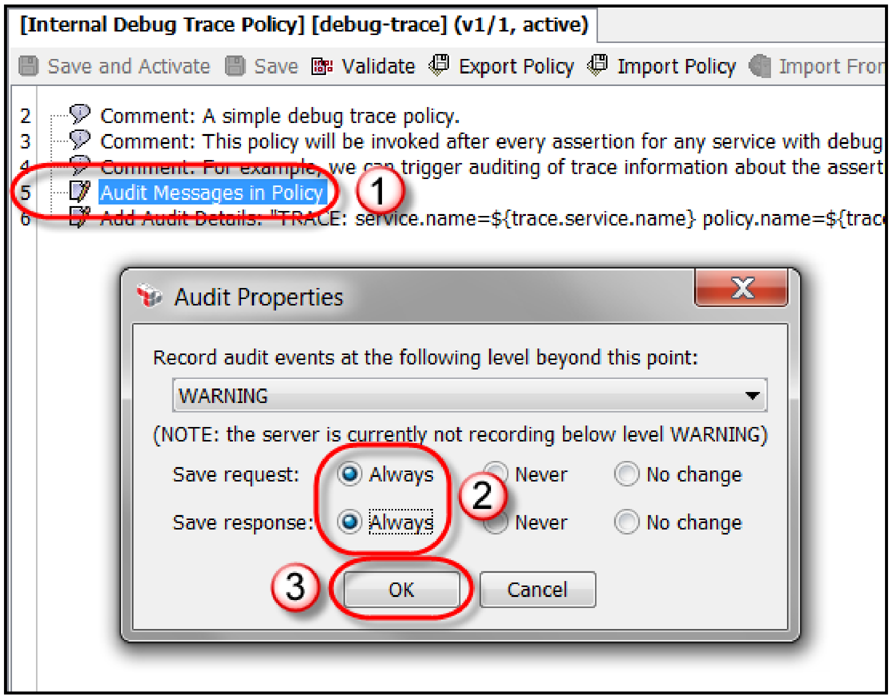
Yes | No

14. In the services and policies tree, notice the ***[Internal Debug Trace Policy]*** that has been added to the gateway. This is the one debug trace policy shared across all services that have debug trace enabled. This policy is executed after the execution of every policy assertion in the active policy of a service with debug trace enabled.


Search
l7apimgmt
Other
Tutorials
[Internal Debug Trace Policy] [debug-trace]

15. We will now make some changes to the default debug trace policy to have it automatically collect more information and to present that information in an easier read way in the associated

logs of audit records. First, in the policy editor, double-click on ***assertion #5, the Audit Messages in Policy*** assertion. Then, in the Audit Properties dialog, select the options to ***Always*** save request and response messages, and click the ***OK*** button.


[Internal Debug Trace Policy] [debug-trace] (v1/1, active)

2 Comment: A simple debug trace policy.
3 Comment: This policy will be invoked after every assertion for any service with debug
4 Comment: For example, we can trigger auditing of trace information about the assert
5 Audit Messages in Policy
6 Add Audit Details: "TRACE: service.name=${trace.service.name} policy.name=${trac

Audit Properties

Record audit events at the following level beyond this point: WARNING

(NOTE: the server is currently not recording below level WARNING)

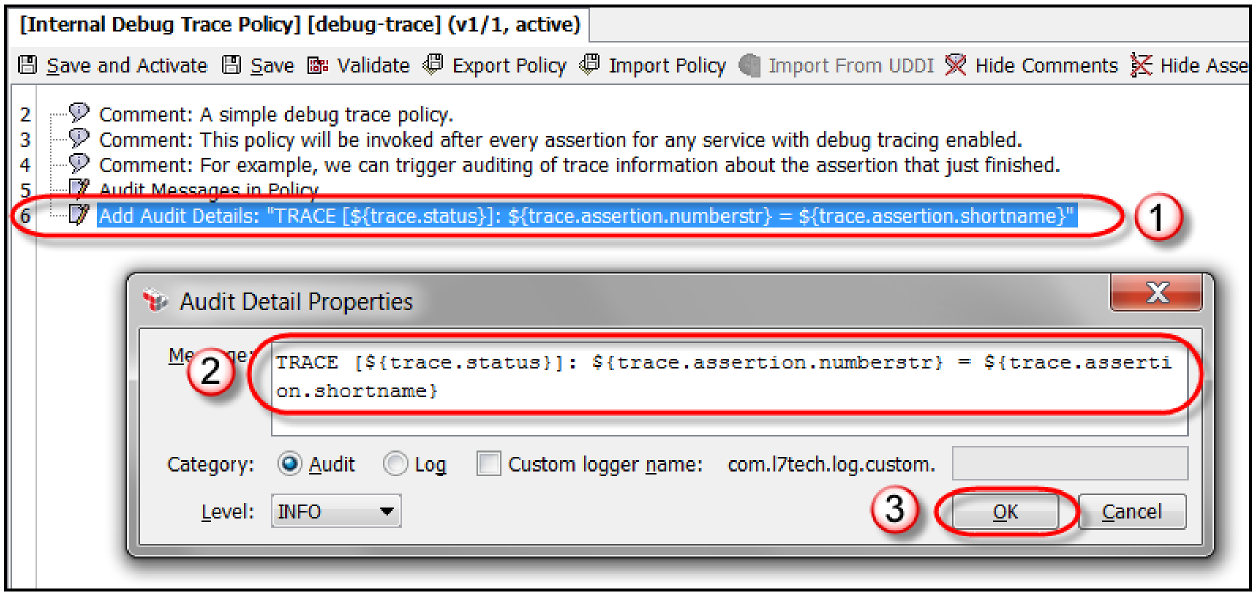
Save request: [✓ Always] Never No change

Save response: [✓ Always] Never No change

*Note: The **Audit Messages in Policy** assertion is what pushes the current audit level of a given message being processed above the default message audit record threshold. Certain exceptional events do the same thing, like failing authentication as demonstrated earlier in this tutorial. Using the **Audit Message in Policy** assertion ensures that an audit record will be created even if an exceptional event has not occurred. In other words, the inclusion of this assertion in the debug trace policy is what ensures that every message processed by a service with debug trace enabled will be audited. The **Audit Message in Policy** assertion can also be used directly in a service's policy when you want to ensure that an audit record will be created for every message processed by that service, even if debug tracing is disabled.*

16. Next, in the policy editor, double-click on ***assertion #6, the Add Audit Details*** assertion. Then, in the Audit Detail Properties dialog, type (or paste) the following text, and click the ***OK*** button.

    TRACE [${trace.status}]: ${trace.assertion.numberstr} = ${trace.assertion.shortname}



```
Comment: A simple debug trace policy.
Comment: This policy will be invoked after every assertion for any service with debug tracing enabled.
Comment: For example, we can trigger auditing of trace information about the assertion that just finished.
Audit Messages in Policy
Add Audit Details: "TRACE [${trace.status}]: ${trace.assertion.numberstr} = ${trace.assertion.shortname}"
```

Message: `TRACE [${trace.status}]: ${trace.assertion.numberstr} = ${trace.assertion.shortname}`

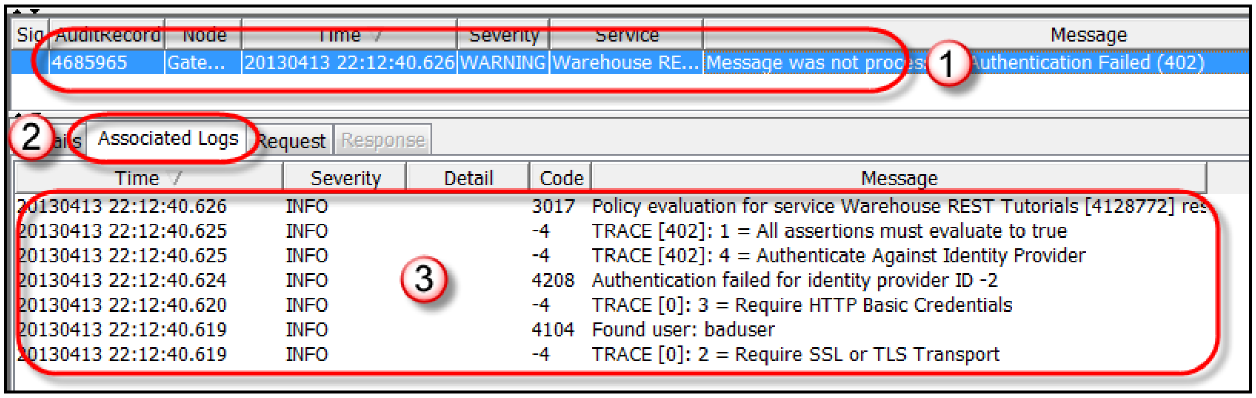
Category: [✓ Audit] Log Custom logger name: com.l7tech.log.custom.

Level: INFO

*Note: The **Add Audit Details** assertion is used to inject additional associated logs into an audit record. The default **Add Audit Details** assertion message in the default debug trace policy contains too much detailed information that is not useful to the casual user and that makes reading the verbose associated logs more difficult. The change we made in this step keeps the most useful information and makes it easier to read.*

*Note: The **Add Audit Details** assertion is often used in the policy of services to emit more useful runtime information to audit records or logs. We will take a look at this in a later step in this tutorial.*

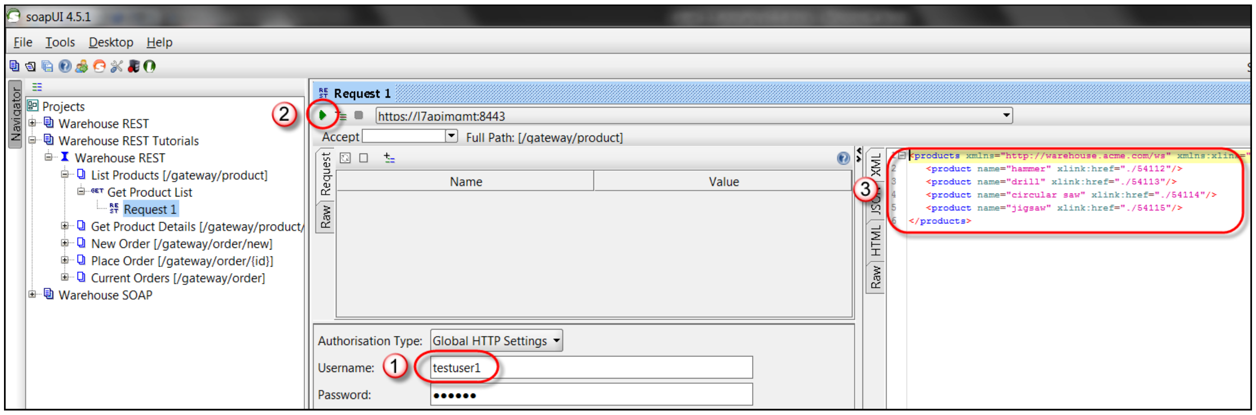
17. On the policy editor toolbar, click the ***Save and Activate*** button.
18. Go to soapUI, and resend the request sent in step 8 of this tutorial.
19. Go to the Policy Manager - Gateway Audit Events dialog, select the most recent audit record, select the ***Associated Logs*** tab, and notice the additional associated logs that have been added to the audit record by the debug trace policy. Also notice that the ***Request*** tab is now enabled, though because our test request was a HTTP GET, its contents are empty. The ***Response*** tab is still disabled, because in this case the response was a SOAP fault, and SOAP fault response messages are not stored with audit records.



Sig	AuditRecord	Node	Time	Severity	Service	Message
	4685965	Gate...	20130413 22:12:40.626	WARNING	Warehouse RE...	Message was not proces... Authentication Failed (402)

Time	Severity	Detail	Code	Message
20130413 22:12:40.626	INFO		3017	Policy evaluation for service Warehouse REST Tutorials [4128772] res
20130413 22:12:40.625	INFO		-4	TRACE [402]: 1 = All assertions must evaluate to true
20130413 22:12:40.625	INFO		-4	TRACE [402]: 4 = Authenticate Against Identity Provider
20130413 22:12:40.624	INFO		4208	Authentication failed for identity provider ID -2
20130413 22:12:40.620	INFO		-4	TRACE [0]: 3 = Require HTTP Basic Credentials
20130413 22:12:40.619	INFO		4104	Found user: baduser
20130413 22:12:40.619	INFO		-4	TRACE [0]: 2 = Require SSL or TLS Transport

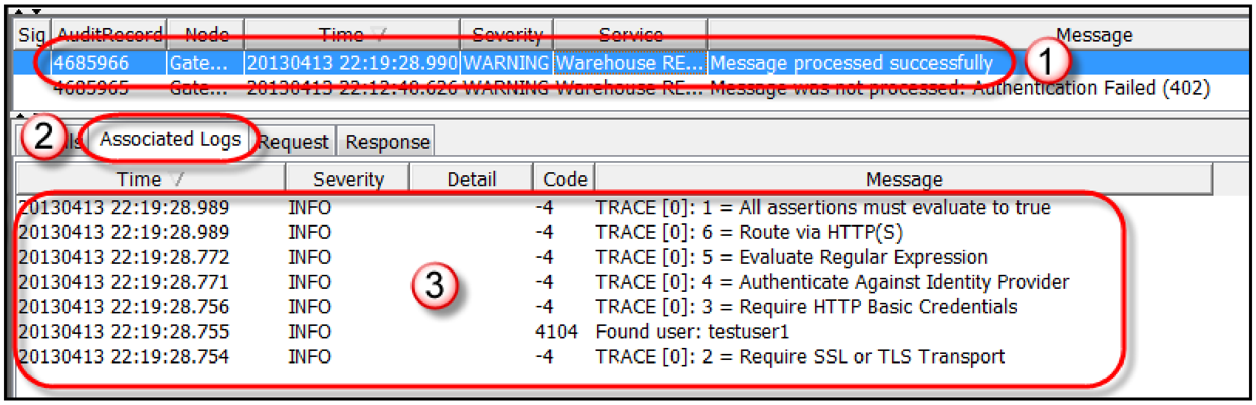
20. Go to soapUI, change the username of the request to a valid user, ***testuser1***, and then resend the request. Notice the successful response.



Authorisation Type: Global HTTP Settings
Username: testuser1
Password: ••••••

```
<products xmlns="http://warehouse.acme.com/ws" xmlns:xlink="
   <product name="hammer" xlink:href="./54112"/>
   <product name="drill" xlink:href="./54113"/>
   <product name="circular saw" xlink:href="./54114"/>
   <product name="jigsaw" xlink:href="./54115"/>
</products>
```

21. Go to the Policy Manager - Gateway Audit Events dialog, select the most recent audit record, select the ***Associated Logs*** tab, and notice the additional associated logs that have been added to the audit record by the debug trace policy.

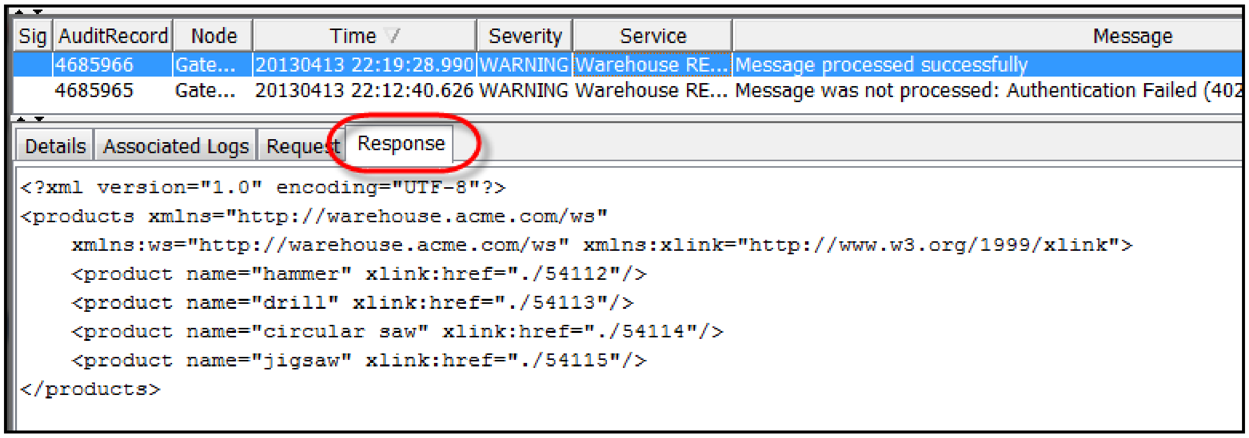


Sig	AuditRecord	Node	Time	Severity	Service	Message
	4685966	Gate...	20130413 22:19:28.990	WARNING	Warehouse RE...	Message processed successfully
	4685965	Gate...	20130413 22:12:48.626	WARNING	Warehouse RE...	Message was not processed: Authentication Failed (402)

Time	Severity	Detail	Code	Message
20130413 22:19:28.989	INFO		-4	TRACE [0]: 1 = All assertions must evaluate to true
20130413 22:19:28.989	INFO		-4	TRACE [0]: 6 = Route via HTTP(S)
20130413 22:19:28.772	INFO		-4	TRACE [0]: 5 = Evaluate Regular Expression
20130413 22:19:28.771	INFO		-4	TRACE [0]: 4 = Authenticate Against Identity Provider
20130413 22:19:28.756	INFO		-4	TRACE [0]: 3 = Require HTTP Basic Credentials
20130413 22:19:28.755	INFO		4104	Found user: testuser1
20130413 22:19:28.754	INFO		-4	TRACE [0]: 2 = Require SSL or TLS Transport

Note: We are only seeing an audit record for this successful message because debug trace is enabled.

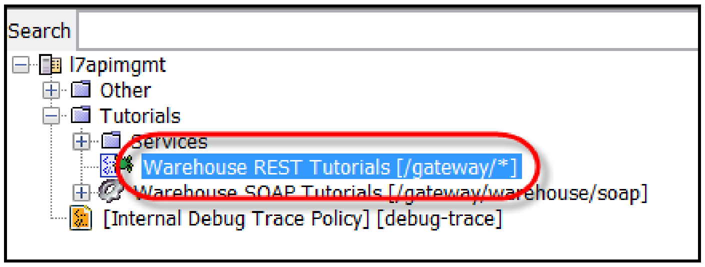
22. Click on the Response tab, and notice the outbound response message that has been stored with the audit record.



Sig	AuditRecord	Node	Time	Severity	Service	Message
	4685966	Gate...	20130413 22:19:28.990	WARNING	Warehouse RE...	Message processed successfully
	4685965	Gate...	20130413 22:12:40.626	WARNING	Warehouse RE...	Message was not processed: Authentication Failed (402

```
<?xml version="1.0" encoding="UTF-8"?>
<products xmlns="http://warehouse.acme.com/ws"
    xmlns:ws="http://warehouse.acme.com/ws" xmlns:xlink="http://www.w3.org/1999/xlink">
    <product name="hammer" xlink:href="./54112"/>
    <product name="drill" xlink:href="./54113"/>
    <product name="circular saw" xlink:href="./54114"/>
    <product name="jigsaw" xlink:href="./54115"/>
</products>
```

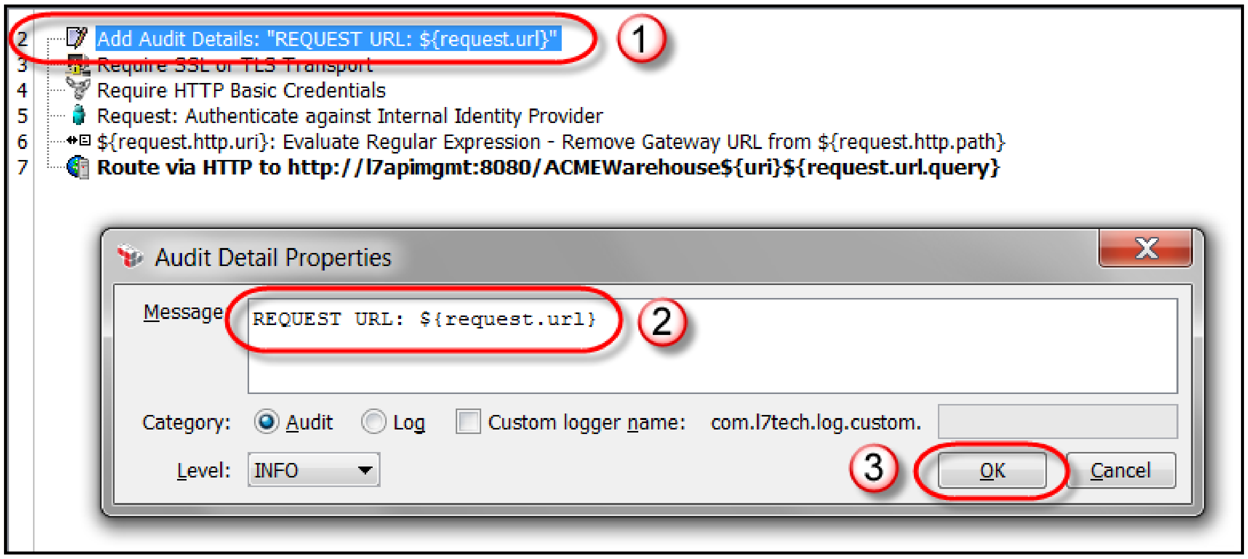
23. Now, go back to Policy Manager, and in the services and policies tree, double-click on the ***Warehouse REST Tutorials*** service to open its active policy in the policy editor.



Search
l7apimgmt
Other
Tutorials
Services
Warehouse REST Tutorials [/gateway/*]
Warehouse SOAP Tutorials [/gateway/warehouse/soap]
[Internal Debug Trace Policy] [debug-trace]

24. From the policy assertion tree, drag and drop the ***Policy Assertions/Logging, Auditing and Alerts/Add Audit Detail*** assertion so that it's assertion #2 in the ***Warehouse REST Tutorials*** service policy in the policy editor. And in the Audit Details Properties dialog, enter the following text, and click the ***OK*** button.

REQUEST URL: ${request.url}



```
2  Add Audit Details: "REQUEST URL: ${request.url}"
3  Require SSL or TLS Transport
4  Require HTTP Basic Credentials
5  Request: Authenticate against Internal Identity Provider
6  ${request.http.uri}: Evaluate Regular Expression - Remove Gateway URL from ${request.http.path}
7  Route via HTTP to http://l7apimgmt:8080/ACMEWarehouse${uri}${request.url.query}
```

Audit Detail Properties

Message: REQUEST URL: ${request.url}

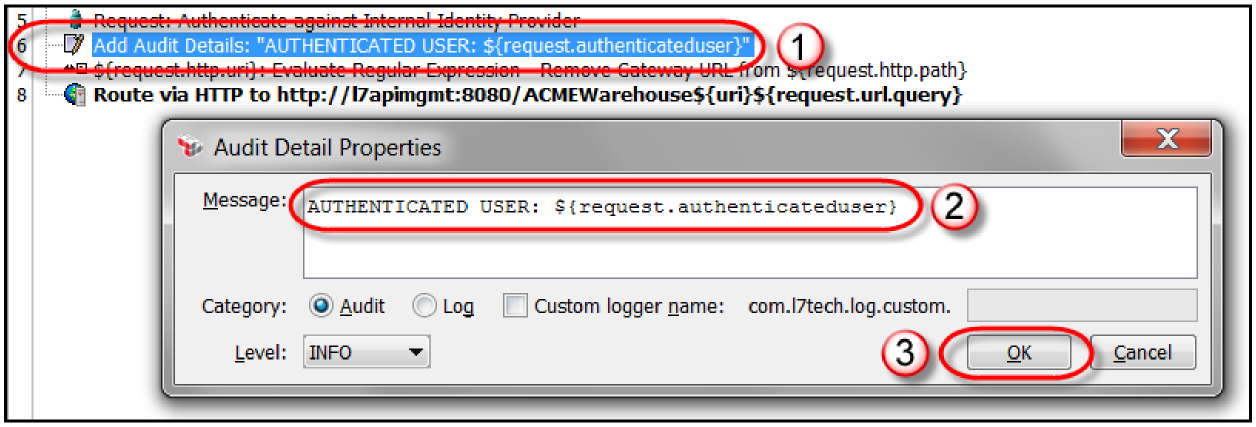
Category: [✓ Audit] Log ☐ Custom logger name: com.l7tech.log.custom.

Level: INFO

Note: ${request.url} is a system defined context variable. You can find more information about it and other system defined context variables in Appendix C of the online Policy Manager help.

25. From the policy assertion tree, drag and drop the ***Policy Assertions/Logging, Auditing and Alerts/Add Audit Detail*** assertion so that it's assertion #6 in the ***Warehouse REST Tutorials*** service policy in the policy editor. And in the Audit Details Properties dialog, enter the following text, and click the ***OK*** button.

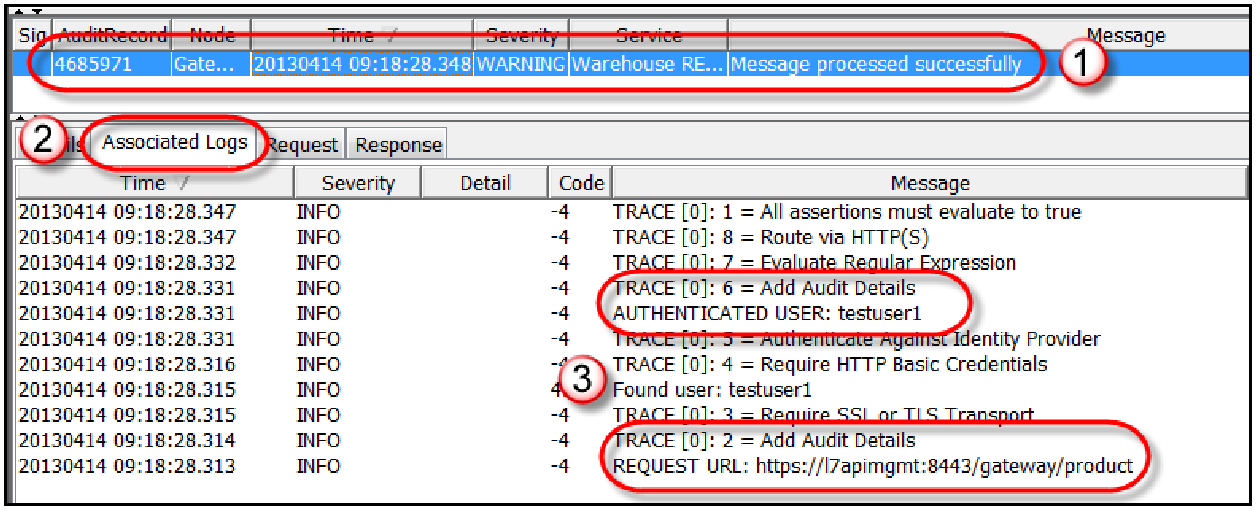
AUTHENTICATED USER: ${request.authenticateduser}



```
5  Request: Authenticate against Internal Identity Provider
6  Add Audit Details: "AUTHENTICATED USER: ${request.authenticateduser}"
7  ${request.http.uri}: Evaluate Regular Expression - Remove Gateway URL from ${request.http.path}
8  Route via HTTP to http://l7apimgmt:8080/ACMEWarehouse${uri}${request.url.query}
```

1
Audit Detail Properties
Message: AUTHENTICATED USER: ${request.authenticateduser}
2
Category: [✓ Audit] Log
Custom logger name: com.l7tech.log.custom.
Level: INFO
3
OK
Cancel

26. On the policy editor toolbar, click the ***Save and Activate*** button.
27. Go to soapUI, and resend the successful request sent on step 20 of this tutorial.
28. Go to the Policy Manager - Gateway Audit Events dialog, select the most recent audit record, select the ***Associated Logs*** tab, and notice the additional associated logs that have been added to the audit record by the debug trace policy.

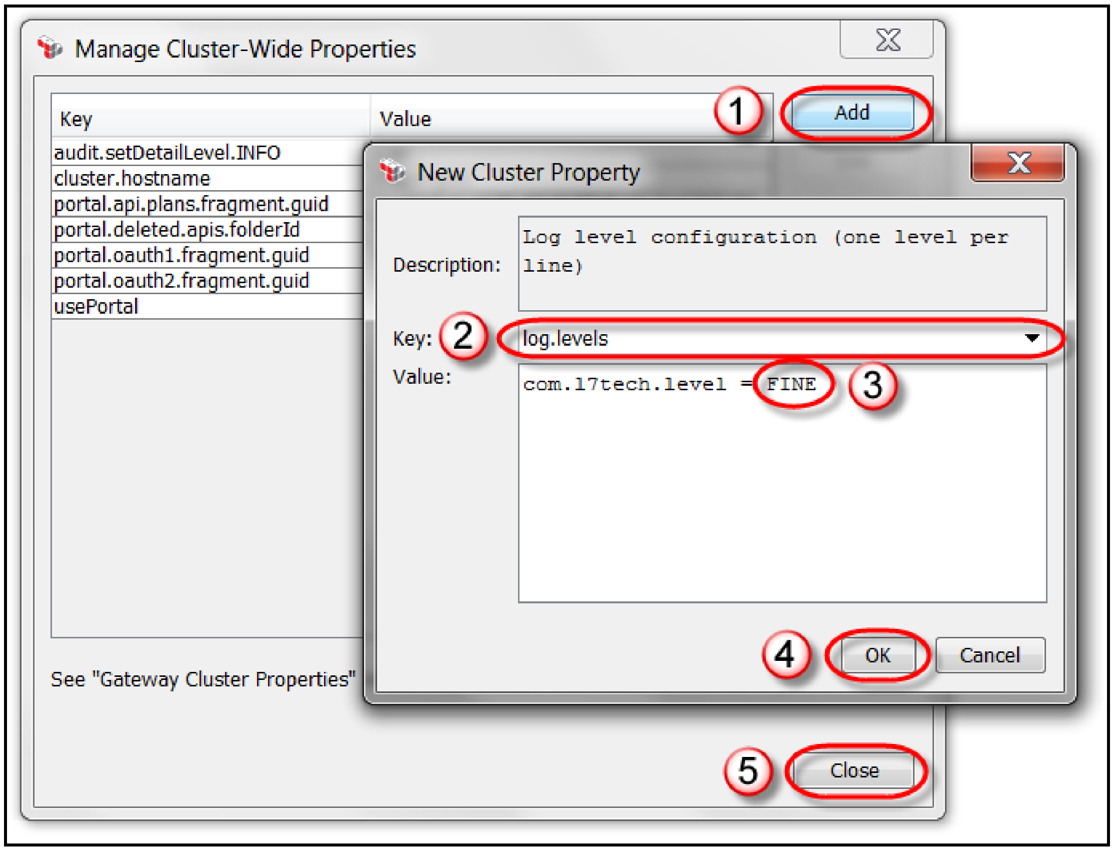


Sig	AuditRecord	Node	Time	Severity	Service	Message
	4685971	Gate...	20130414 09:18:28.348	WARNING	Warehouse RE...	Message processed successfully

Time	Severity	Detail	Code	Message
20130414 09:18:28.347	INFO		-4	TRACE [0]: 1 = All assertions must evaluate to true
20130414 09:18:28.347	INFO		-4	TRACE [0]: 8 = Route via HTTP(S)
20130414 09:18:28.332	INFO		-4	TRACE [0]: 7 = Evaluate Regular Expression
20130414 09:18:28.331	INFO		-4	TRACE [0]: 6 = Add Audit Details
20130414 09:18:28.331	INFO		-4	AUTHENTICATED USER: testuser1
20130414 09:18:28.331	INFO		-4	TRACE [0]: 5 = Authenticate Against Identity Provider
20130414 09:18:28.316	INFO		-4	TRACE [0]: 4 = Require HTTP Basic Credentials
20130414 09:18:28.315	INFO		-4	Found user: testuser1
20130414 09:18:28.315	INFO		-4	TRACE [0]: 3 = Require SSL or TLS Transport
20130414 09:18:28.314	INFO		-4	TRACE [0]: 2 = Add Audit Details
20130414 09:18:28.313	INFO		-4	REQUEST URL: https://l7apimgmt:8443/gateway/product

Note: Most problems can be diagnosed with the information found in a verbose debug trace audit record. However, when more detailed information is needed, you can lower the level at which the gateway emits log events, and you can create a log to catch events for a specific service. We will do this next.

29. In Policy Manager, select the ***Tasks/Manage Cluster-Wide Properties*** menu item.
30. In the Manage Cluster-Wide Properties dialog, click the ***Add*** button. Then, in the New Cluster Property dialog, select ***log.levels*** from the Key dropdown list, replace the word CONFIG with ***FINE*** in the Value field, and click the ***OK*** button. Finally, in the Manage Cluster-Wide Properties dialog, click the ***Close*** button.



Key	Value
audit.setDetailLevel.INFO	
cluster.hostname	
portal.api.plans.fragment.guid	
portal.deleted.apis.folderId	
portal.oauth1.fragment.guid	
portal.oauth2.fragment.guid	
usePortal	

See "Gateway Cluster Properties"

Description: Log level configuration (one level per line)

Key: log.levels

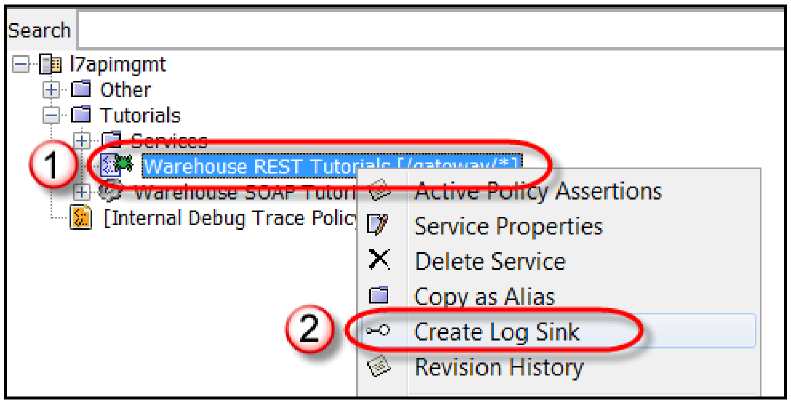
Value:

```
com.l7tech.level = FINE
```

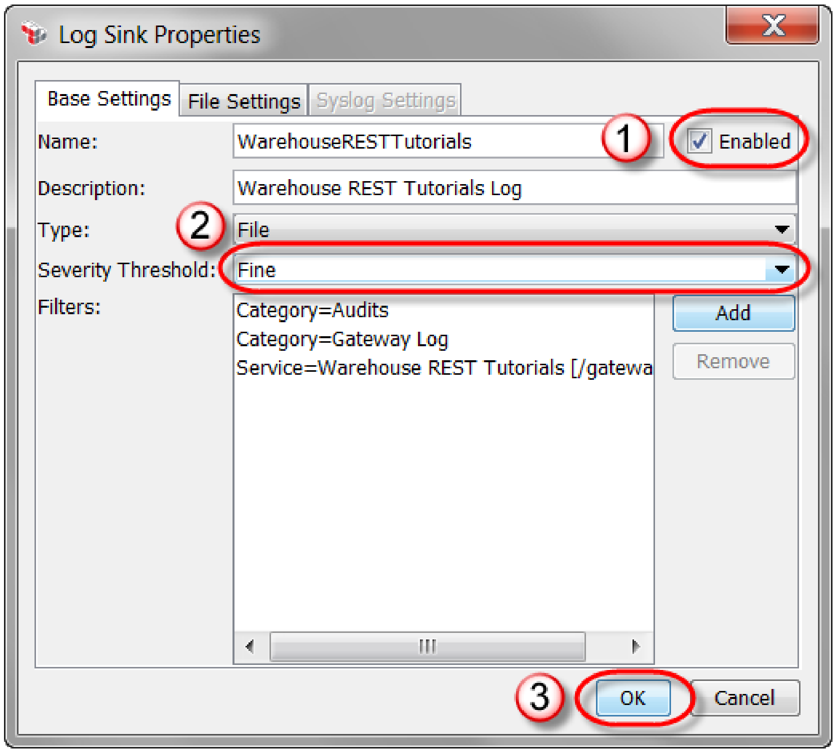
Note: The log.levels cluster property sets the level at and above which the gateway will emit log events. In other words, the gateway is instrumented throughout its code to emit log events at varying levels of detail (including FINEST, FINER, FINE, CONFIG, INFO, WARNING, SEVERE). If log.levels is set to CONFIG, and the gateway encounters a FINE event, it will not be emitted, but if the gateway encounters CONFIG or INFO (or above) events, then they will be emitted.

Emitted events effectively go nowhere and cause little to no performance overhead on the gateway if there are no log sinks configured to catch those events. We will configure a log sink to catch FINE or higher events for the Warehouse REST Tutorials service next.

31. In the services and policy tree, right click the ***Warehouse REST Tutorials*** service, and in the context menu, select the ***Create Log Sink*** menu item.



32. In the Log Sink Properties dialog, check the ***Enabled*** option, select ***Fine*** from the Severity Threshold dropdown list, and click the ***OK*** button.



Name: WarehouseRESTTutorials

Enabled [✓]

Description: Warehouse REST Tutorials Log

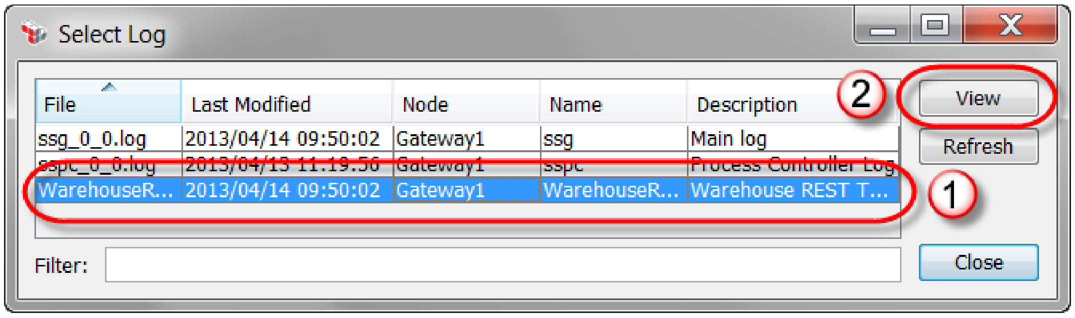
Type: File

Severity Threshold: Fine

Filters:
Category=Audits
Category=Gateway Log
Service=Warehouse REST Tutorials [/gatewa

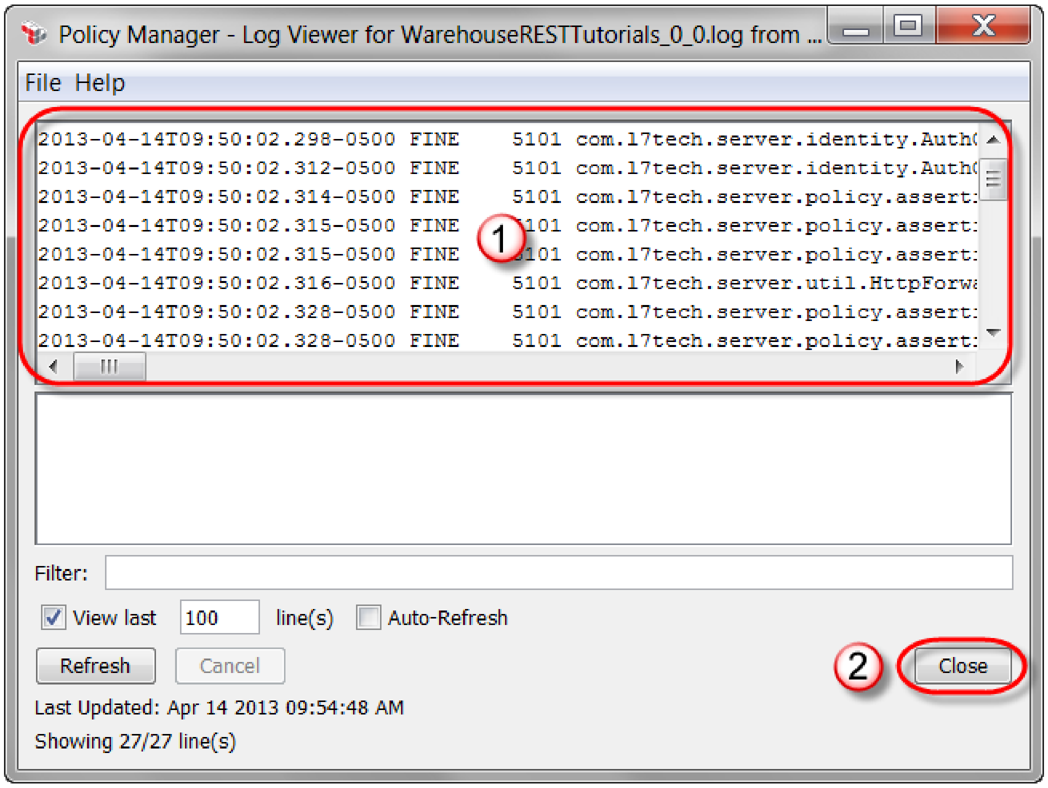
33. Go to soapUI, and resend the successful request sent on step 20 of this tutorial.
34. In Policy Manager, select the ***View/View Logs*** menu item.

35. In the Select Log dialog, select the ***WarehouseRESTTutorials_0_0.log*** file, and click the ***View*** button.



File	Last Modified	Node	Name	Description
ssg_0_0.log	2013/04/14 09:50:02	Gateway1	ssg	Main log
sspc_0_0.log	2013/04/13 11:19:56	Gateway1	sspc	Process Controller Log
WarehouseR...	2013/04/14 09:50:02	Gateway1	WarehouseR...	Warehouse REST T...

36. In the Policy Manager - Log Viewer, you should see all of the log events associated with the last request message sent. Explore the log events, and when you're ready, click the ***Close*** button.

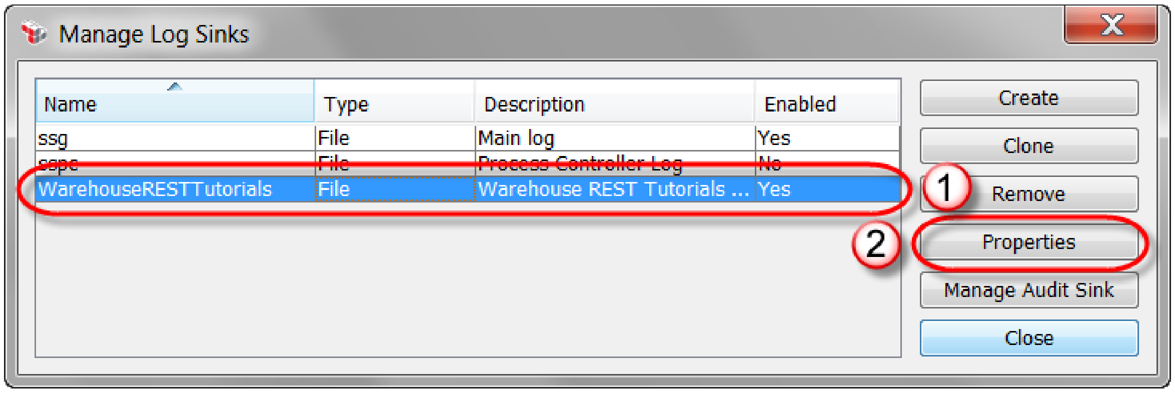

Policy Manager - Log Viewer for WarehouseRESTTutorials_0_0.log from ...
File Help
Filter:
View last
100
line(s)
Auto-Refresh
Refresh
Cancel
Close
1
2

```
2013-04-14T09:50:02.298-0500 FINE    5101 com.l7tech.server.identity.Auth(
2013-04-14T09:50:02.312-0500 FINE    5101 com.l7tech.server.identity.Auth(
2013-04-14T09:50:02.314-0500 FINE    5101 com.l7tech.server.policy.assert:
2013-04-14T09:50:02.315-0500 FINE    5101 com.l7tech.server.policy.assert:
2013-04-14T09:50:02.315-0500 FINE    5101 com.l7tech.server.policy.assert:
2013-04-14T09:50:02.316-0500 FINE    5101 com.l7tech.server.util.HttpForwa
2013-04-14T09:50:02.328-0500 FINE    5101 com.l7tech.server.policy.assert:
2013-04-14T09:50:02.328-0500 FINE    5101 com.l7tech.server.policy.assert:
```

Last Updated: Apr 14 2013 09:54:48 AM
Showing 27/27 line(s)

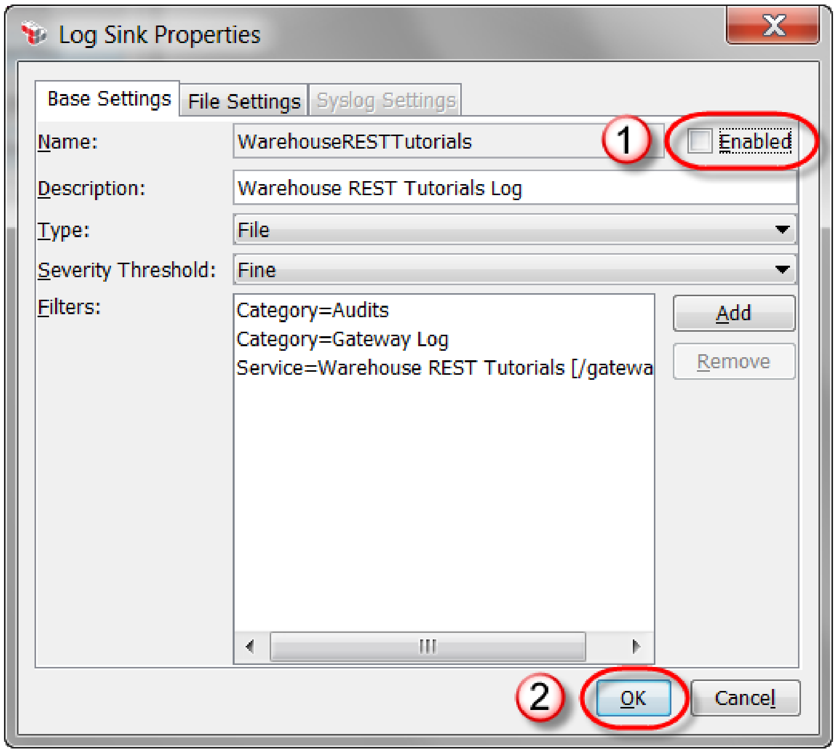
37. In the Select Log dialog, click the ***Close*** button.
38. In Policy Manager, select the ***Tasks/Manage Log/Audit Sinks*** menu item.

39. In the Manage Log Sinks dialog, select the ***WarehouseRESTTutorials*** log sink, and click the ***Properties*** button.



Name	Type	Description	Enabled
ssg	File	Main log	Yes
~~sspc~~	~~File~~	~~Process Controller Log~~	~~No~~
WarehouseRESTTutorials	File	Warehouse REST Tutorials ...	Yes

40. In the Log Sink Properties dialog, uncheck the ***Enabled*** option, and click the ***OK*** button.



Name: WarehouseRESTTutorials ☐ Enabled

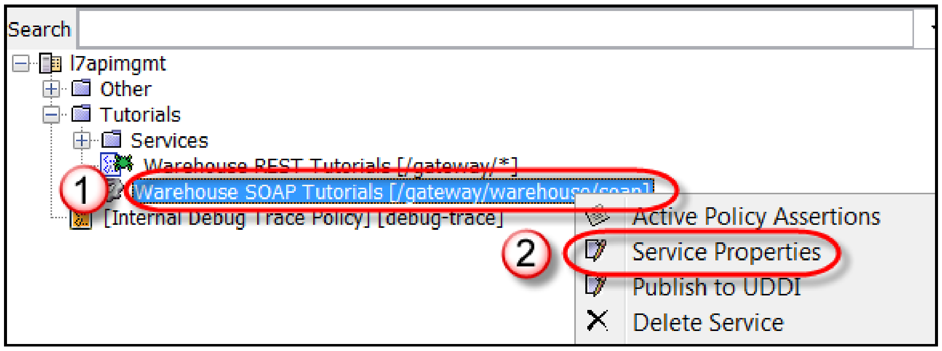
Description: Warehouse REST Tutorials Log

Type: File

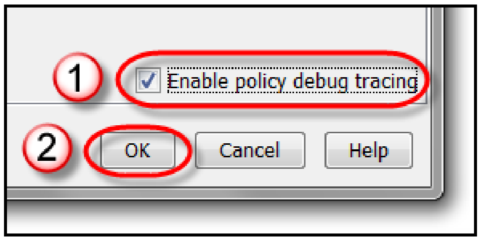
Severity Threshold: Fine

Filters:
Category=Audits
Category=Gateway Log
Service=Warehouse REST Tutorials [/gatewa

41. Because it may be useful for other tutorials, let's enable debug trace for the Warehouse SOAP Tutorials as well. In the services and policies tree, right click the ***Warehouse SOAP Tutorials*** service, and in the context menu, select the ***Service Properties*** menu item.


Search
l7apimgmt
Other
Tutorials
Services
Warehouse REST Tutorials [/gateway/*]
1
Warehouse SOAP Tutorials [/gateway/warehouse/soap]
[Internal Debug Trace Policy] [debug-trace]
2
Active Policy Assertions
Service Properties
Publish to UDDI
Delete Service

42. In the Published Service Properties dialog, check the ***Enable policy debug tracing option***, and click the ***OK*** button.



43. 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 7 Complete**.
44. You are done with this tutorial.

7.4 Additional Context

There are many tools, tips and techniques for troubleshooting runtime policy behavior on a gateway. This tutorial does not try to cover all of them.

This tutorial demonstrates a commonly used approach the does not initially require changes to your policy or a lot of effort to enable and disable. Namely, this tutorial shows you how to enable debug tracing of a service to create verbose audit records for messages processed by that service.

Most problems can be understood by looking at verbose audit records. In some cases, more information is needed. Sometimes this information can be provided by adding additional policy assertions to the policy of your service to emit more useful information to the audit trail. Other times this information can be provided by enabling more verbose logging and creating a log sink to capture detailed log messages for the service that you're troubleshooting. This tutorial also explores both of these methods.

On rare occasions, problems occur before or after the gateway has an opportunity to audit or log what's happening. In those cases, it can sometimes help to take a network packet capture of service traffic on the gateway, and use a free network packet capture analysis solution like Wireshark to inspect the packets. This tutorial does not cover how that's done.

Note: Debug trace should not be left enabled in a production or pre-production performance test environment. It almost always captures more information than is need for anything but troubleshooting, and it does so for a significant performance and storage cost.

Note: Verbose log sinks should not be left enabled in a production or pre-production performance test environment. They almost always captures more information than is need for anything but troubleshooting, and they do so for a significant performance and storage cost.