Hands-On Lab:
Using a Routing Policy

Objectives

In this hands-on lab you will define a routing policy using content-based routing.

Steps

1) Open the **Windows Services** panel and double-check that the following services, needed for API Gateway and the native services, are up and running. If a service is not running, start the service.

   a) **Software AG Integration Server 10.11 (default)**

   b) **Software AG Internal Integration Server 10.11**

   c) **Software AG Threat Protection Integration Server 10.11**

2) Open a tab in Mozilla Firefox and login to the (internal) API Gateway UI as user **Sumala** | **manage**.

3) On tab **APIs**, click **+ Create API** to create a SOAP-based API.
   Select option **Import API from URL** and provide the following properties:

   a) URL: **http://<*hostname*>:7777/ws/calculator:calculator?WSDL**

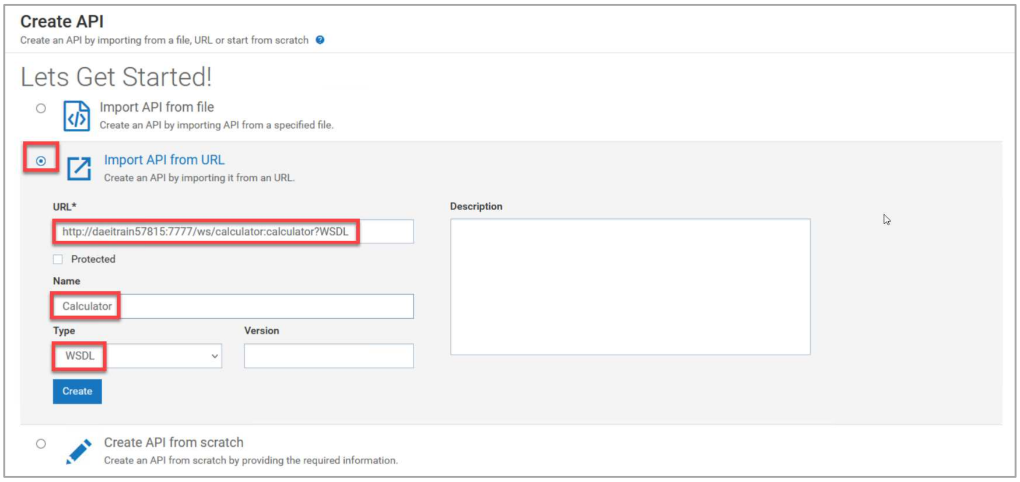
   b) Description: < ***leave empty*** >

   c) Protection: < ***leave unchecked*** >

   d) Name: **Calculator**

   e) Type: **WSDL**

   f) Version**:** < ***leave empty*** >



Create API
Create an API by importing from a file, URL or start from scratch

Lets Get Started!

○ Import API from file
Create an API by importing API from a specified file.

● Import API from URL
Create an API by importing it from an URL.

URL*: http://daeitrain57815:7777/ws/calculator:calculator?WSDL

□ Protected

Name: Calculator

Type: WSDL

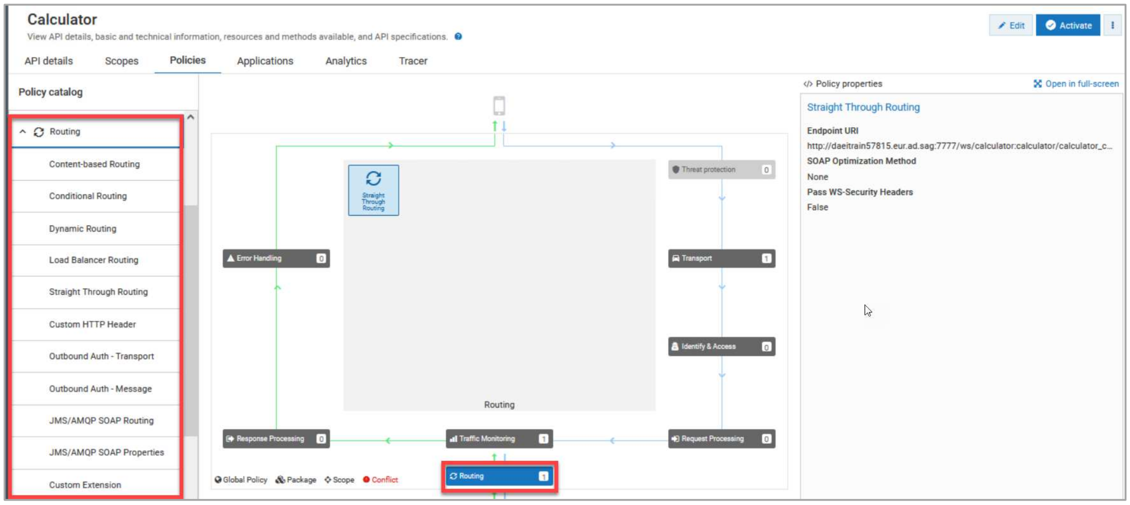
Version:

Description:

○ Create API from scratch
Create an API from scratch by providing the required information.

   Click **Create**.

4) On the API's **Policies** tab, expand **Routing** from the left-hand Policy catalog.



Straight Through Routing

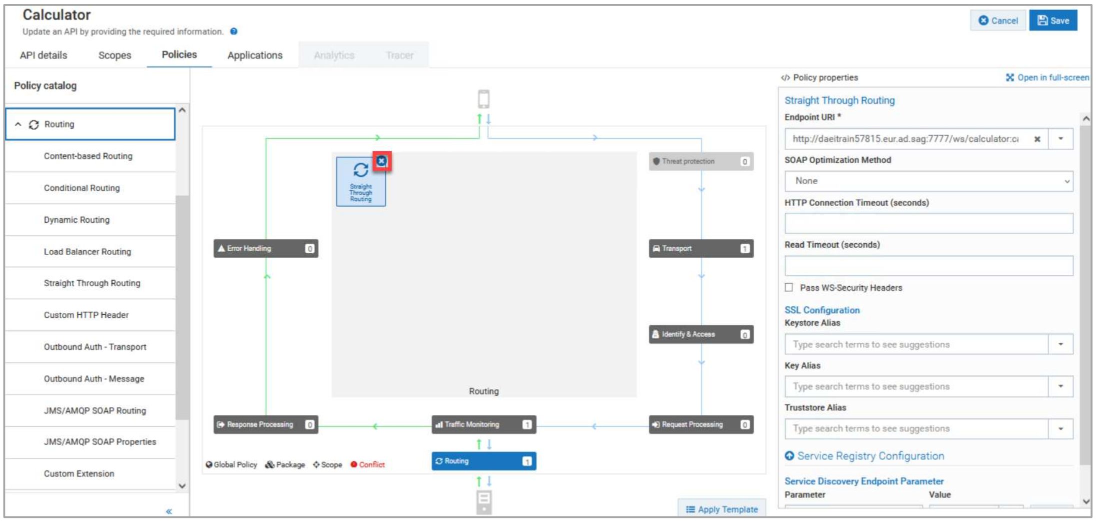
Endpoint URI
http://daeitrain57815.eur.ad.sag:7777/ws/calculator:calculator/calculator_c...

SOAP Optimization Method
None

Pass WS-Security Headers
False

a. Switch to **Edit** mode.

b. Remove the configured default Policy **Straight Through Routing**.



Straight Through Routing

Endpoint URI *
http://daeitrain57815.eur.ad.sag:7777/ws/calculator:c

SOAP Optimization Method
None

HTTP Connection Timeout (seconds)

Read Timeout (seconds)

□ Pass WS-Security Headers

SSL Configuration

Keystore Alias
Type search terms to see suggestions

Key Alias
Type search terms to see suggestions

Truststore Alias
Type search terms to see suggestions

Service Registry Configuration

Service Discovery Endpoint Parameter

Parameter	Value

c. Add a **Content-based Routing** policy from the left-hand Policy catalog.

d. To configure the routing definitions, open the text file **CalculatorRouting.txt** located in folder **C:\Training\E456B-7BE\Lab11**. This file contains input values for copy & paste.

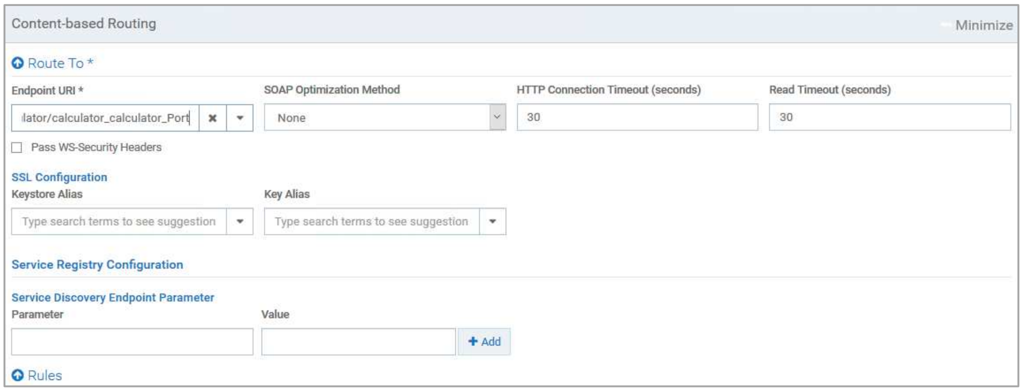
e. Open the **Content-based Routing** policy in **full-screen mode**. Provide the following properties:

i. Route To.

1. Endpoint URI: < ***copy & paste from text file*** >
2. SOAP Optimization Method: **None**
3. HTTP Connection Timeout(seconds) **30**
4. Read Timeout (Seconds) **30**
5. Pass WS-Security Headers: < ***leave unchecked*** >

ii. SSL Configuration:

1. Keystore Alias: < ***leave empty*** >
2. Key Alias: < ***leave empty*** >



Content-based Routing — Minimize

Route To *

Endpoint URI *	SOAP Optimization Method	HTTP Connection Timeout (seconds)	Read Timeout (seconds)
ılator/calculator_calculator_Port	None	30	30

□ Pass WS-Security Headers

SSL Configuration

Keystore Alias	Key Alias
Type search terms to see suggestion	Type search terms to see suggestion

Service Registry Configuration

Service Discovery Endpoint Parameter

Parameter	Value	
		+ Add

Rules

iii. Rules:

Click on **+Add rule** to create the following Rule to follow when calling the Add operation:

1. Name: **MyContentRoutingForAdd**

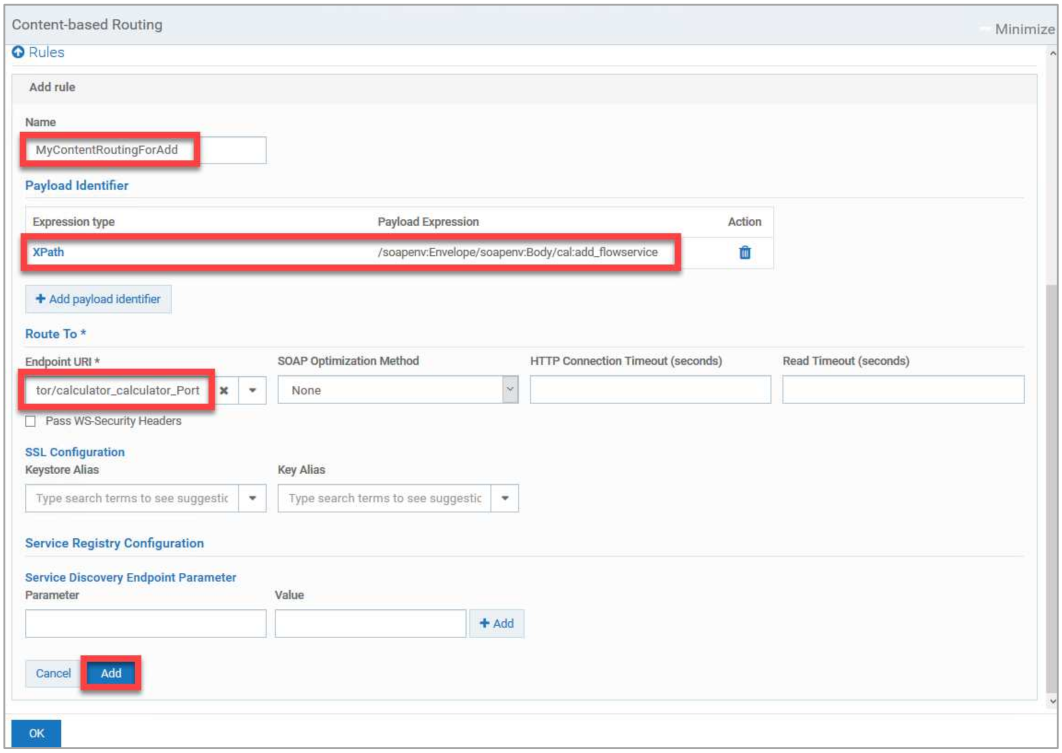
Click on **+ Add payload identifier**.

2. Expression type: **XPath**
3. Payload Expression: < ***copy and paste XPath Expression for Add*** >

Click on **Add**.

4. Route To > Endpoint URI: < ***copy & paste Changed Endpoint URI for Add*** >

Leave all other default values unchanged.



Content-based Routing

Rules

Add rule

Name: MyContentRoutingForAdd

Payload Identifier

Expression type	Payload Expression	Action
XPath	/soapenv:Envelope/soapenv:Body/cal:add_flowservice	🗑

Route To *

Endpoint URI *: tor/calculator_calculator_Port

SOAP Optimization Method: None

HTTP Connection Timeout (seconds):

Read Timeout (seconds):

□ Pass WS-Security Headers

SSL Configuration

Keystore Alias: Type search terms to see suggestic

Key Alias: Type search terms to see suggestic

Service Registry Configuration

Service Discovery Endpoint Parameter

Parameter:

Value:

Click on **Add** at the very bottom of the Rules section.

iv. Rules:

Click on **+ Add rule** again to create a second Rule to follow when calling the Multiply operation:

1. Name: **MyContentRoutingForMultiply**

Click on **Add payload identifier**.

2. Expression type: **XPath**

3. Payload Expression: < ***copy and paste XPath Expression for Multiply*** >

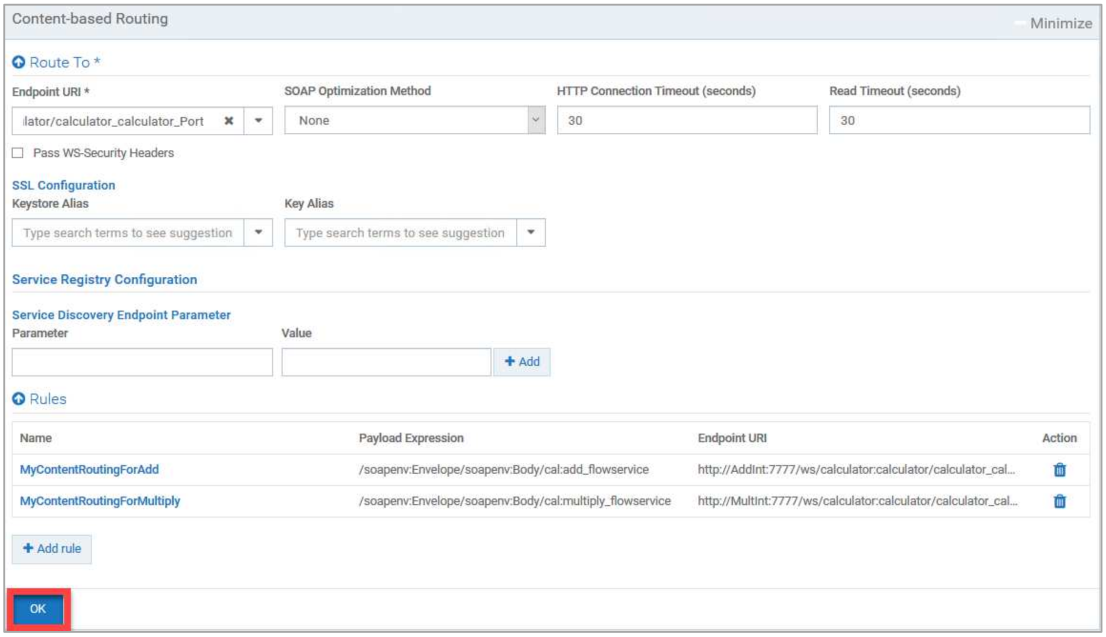
Click on **Add**.

4. Route To > Endpoint URI: < ***copy & paste Changed Endpoint URI for Multiply*** >

Leave all other default values unchanged.

Click on **Add** at the very bottom of the Rules section.

f. Click **OK**.



Content-based Routing

Route To *

Endpoint URI *: lator/calculator_calculator_Port

SOAP Optimization Method: None

HTTP Connection Timeout (seconds): 30

Read Timeout (seconds): 30

□ Pass WS-Security Headers

SSL Configuration

Keystore Alias: Type search terms to see suggestion

Key Alias: Type search terms to see suggestion

Service Registry Configuration

Service Discovery Endpoint Parameter

Parameter

Value

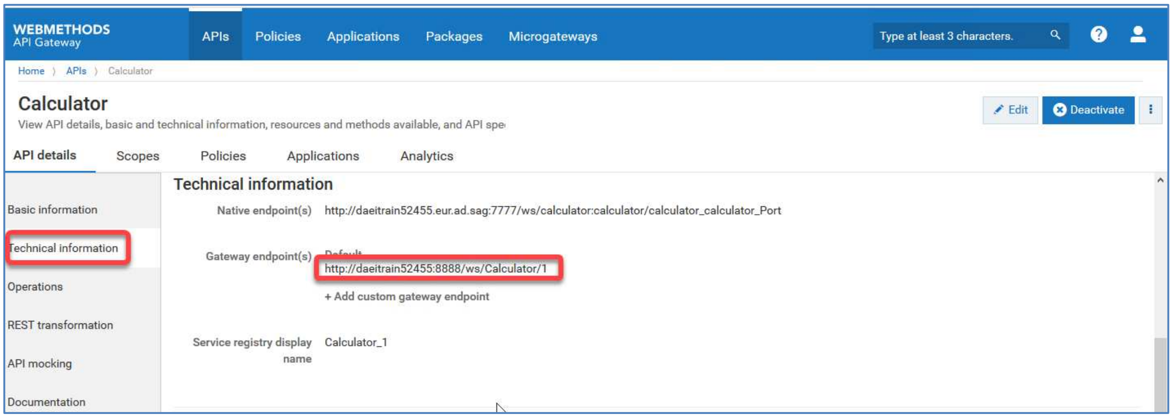
Rules

Name	Payload Expression	Endpoint URI	Action
MyContentRoutingForAdd	/soapenv:Envelope/soapenv:Body/cal:add_flowservice	http://AddInt:7777/ws/calculator:calculator/calculator_cal...	🗑
MyContentRoutingForMultiply	/soapenv:Envelope/soapenv:Body/cal:multiply_flowservice	http://MultInt:7777/ws/calculator:calculator/calculator_cal...	🗑

5. **Save** your API.

6. **Activate** the **Calculator** API. Confirm with **Yes**.

7. Navigate to **API details > Technical information**. Review and copy the **Gateway endpoint** URI of the Calculate API into the clipboard.



Home 〉 APIs 〉 Calculator

Calculator

View API details, basic and technical information, resources and methods available, and API spe

Technical information

Native endpoint(s): http://daeitrain52455.eur.ad.sag:7777/ws/calculator:calculator/calculator_calculator_Port

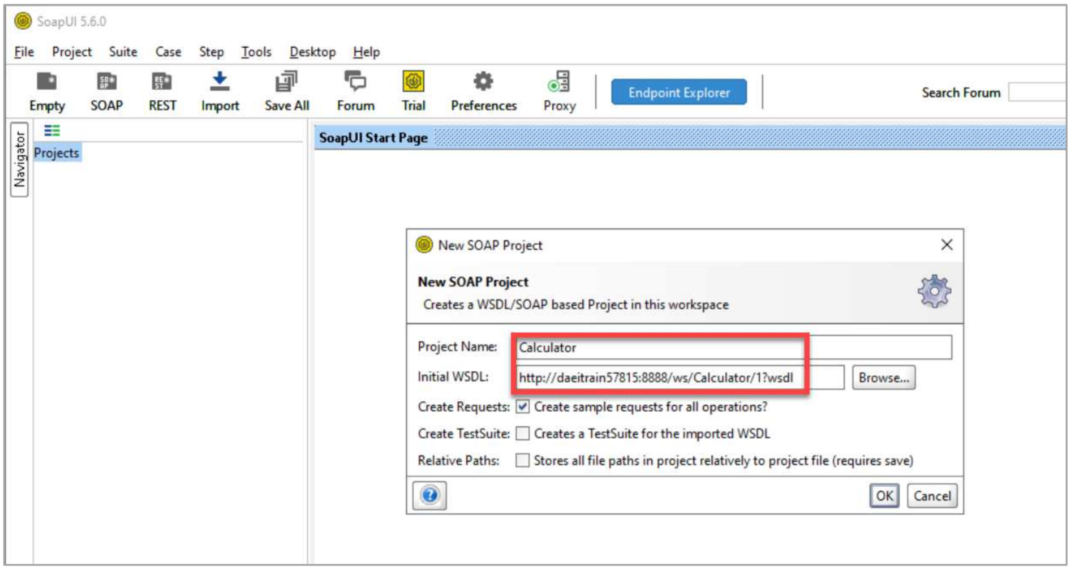
Gateway endpoint(s): Default
http://daeitrain52455:8888/ws/Calculator/1

Service registry display name: Calculator_1

Note: Because we defined a Load balancer in hands-on lab “Managing API Threat Protection”, the Gateway endpoint shows port **8888** of our Thread Protection API Gateway instead of port 5555 of the (internal) API Gateway.

8. Launch **SOAP UI** to test the Calculator API:

   a. Use **File** > **New SOAP Project** to create at test project with name **Calculator.** Paste the API Gateway endpoint URL from the clipboard into field **Initial WSDL** and append the suffix **?wsdl.**
   This value should like this:
   http://<*hostname*>:8888/ws/Calculator/1?wsdl


SoapUI 5.6.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Navigator
Projects
SoapUI Start Page
New SOAP Project
OK
Cancel
Browse...

New SOAP Project
Creates a WSDL/SOAP based Project in this workspace

Project Name: Calculator
Initial WSDL: http://daeitrain57815:8888/ws/Calculator/1?wsdl
Create Requests: ☑ Create sample requests for all operations?
Create TestSuite: □ Creates a TestSuite for the imported WSDL
Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

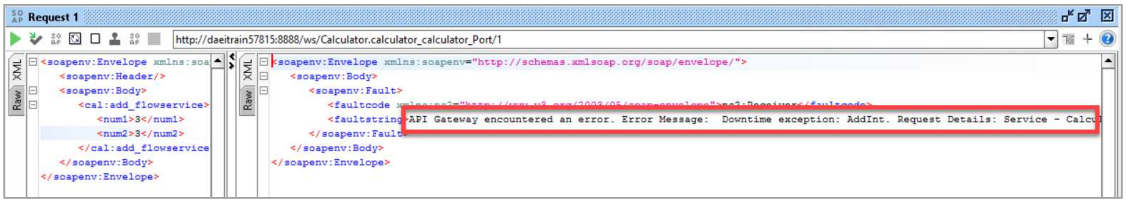
   Click **OK** to create a test client.

   b. In the Projects view, navigate to **Calculator** > **calculator_calculator_Binder** > **add_flowservice** > **Request 1**. Double-click **Request 1** to open a corresponding request test panel. Provide the following values:

      i. num1: **3**

      ii. num2: **3**

   c. Hit the run ▶ icon.



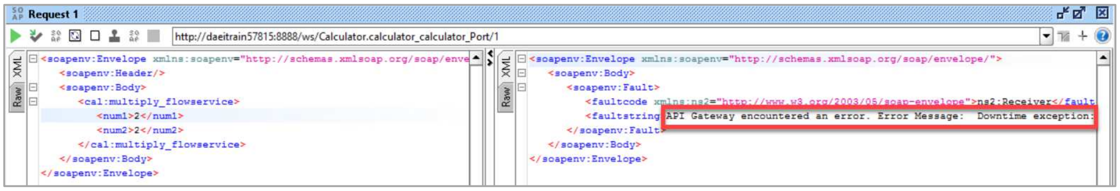
http://daeitrain57815:8888/ws/Calculator.calculator_calculator_Port/1

```
<soapenv:Envelope xmlns:soa
   <soapenv:Header/>
   <soapenv:Body>
      <cal:add_flowservice>
         <num1>3</num1>
         <num2>3</num2>
      </cal:add_flowservice
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <soapenv:Fault>
         <faultcode xmlns:ns2="http://www.w3.org/2003/05/soap-envelope">ns2:Receiver</faultcode>
         <faultstring>API Gateway encountered an error. Error Message:  Downtime exception: AddInt. Request Details: Service - Calcu
      </soapenv:Fault>
   </soapenv:Body>
</soapenv:Envelope>
```

Note: SOAP UI returns a downtime exception error, because host **AddInt** as defined as changed endpoint in our content-based routing policy does not exist.

d. Configure and run a second request for operation **multiply_flowservice**.



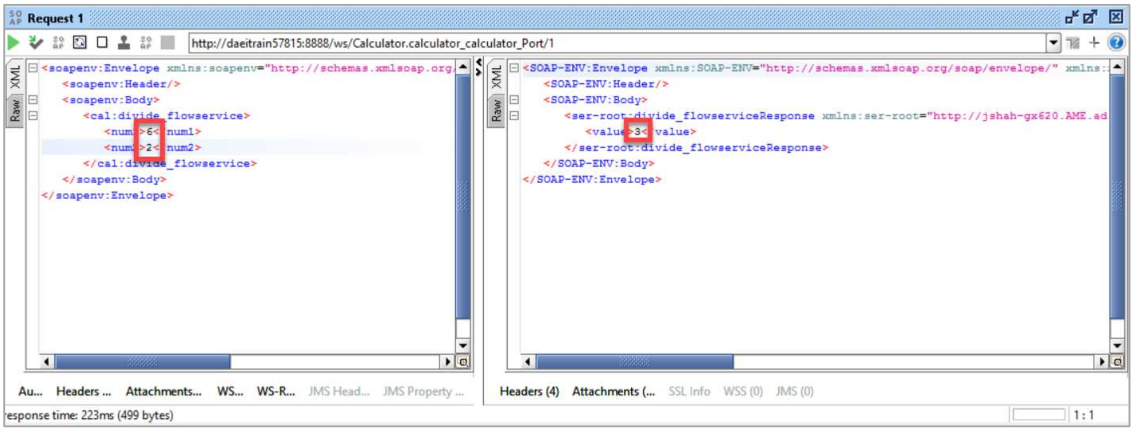
http://daeitrain57815:8888/ws/Calculator.calculator_calculator_Port/1

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/enve
   <soapenv:Header/>
   <soapenv:Body>
      <cal:multiply_flowservice>
         <num1>2</num1>
         <num2>2</num2>
      </cal:multiply_flowservice>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <soapenv:Fault>
         <faultcode xmlns:ns2="http://www.w3.org/2003/05/soap-envelope">ns2:Receiver</fault
         <faultstring>API Gateway encountered an error. Error Message:  Downtime exception:
      </soapenv:Fault>
   </soapenv:Body>
</soapenv:Envelope>
```

Note: Again, SOAP UI returns a dwontime exception error, because host **MultInt** as defined as changed endpoint in our content-based routing policy does not exist.

e. Configure and run a third request for operation **divide_flowservice**. This will succeed:



http://daeitrain57815:8888/ws/Calculator.calculator_calculator_Port/1

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
   <soapenv:Header/>
   <soapenv:Body>
      <cal:divide_flowservice>
         <num1>6</num1>
         <num2>2</num2>
      </cal:divide_flowservice>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns::
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ser-root:divide_flowserviceResponse xmlns:ser-root="http://jshah-gx620.AME.ad
         <value>3</value>
      </ser-root:divide_flowserviceResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

esponse time: 223ms (499 bytes)

Note: In case you get an IS error like **Access to WSDescriptor calculator:calculator denied...** you have to change the execution permissions of the invoked Web Service Descriptor in your internal Integration Server (port :7777).

This can be done in Designer's Service Development perspective. In the Package Navigator view, change the **Execute ACL** permissions of the Web Servive Descriptor (WSD) **calculator:calculator** within IS package **MediatorECRWebservices** to **Anonymous**.

9. *For extra credit:*
   Enable a proper execution of the **add_flowservice** and **multiply_flowservice** by adding appropriate entries for “artificial” hosts named **AddInt** and **MultInt** to your **hosts** file at **C:\Windows\system32\drivers\etc**:

```
127.0.0.1 Addint Multint
```

Rerun the SOAP UI test clients.