

Hands-on Lab: Data Masking and Transformation

Objectives

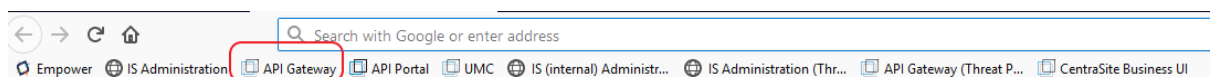
In this hands-on lab, you will import a REST API then use the Data Masking and Transformation to alter the payload and header.

Steps

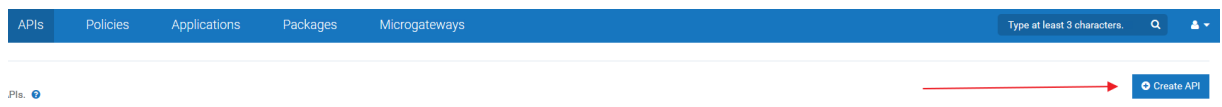
1. Open Windows Services UI and check that the following services are started. If they are not running, start the service.
 - a) **Software AG Integration Server 10.5 (default)**
 - b) **Software AG Internal Integration Server 10.5 (default)**
 - c) **Software AG API Gateway Data Store 10.5**

Note: Since we are not going to interface with any other Software AG components, only these are needed to be started.

2. You can monitor the progress of the Integration Server startup sequence in **Baretail** by opening the following logfiles in **Baretail**:
 - a) **C:\SoftwareAG\IntegrationServer\instances\default\logs\server.log**
 - b) **C:\SoftwareAGInternal\IntegrationServer\instances\default\logs\server.log**
3. Login to API Gateway by clicking the API Gateway link (just below the URL field). Login as API Gateway Administrator Sumala (**Sumala | manage**).



4. You are asked to create a **Numbers** API by importing a swagger file. To create the Numbers API, click on the **APIs** tab then click the **+ Create API** button located in the upper right.

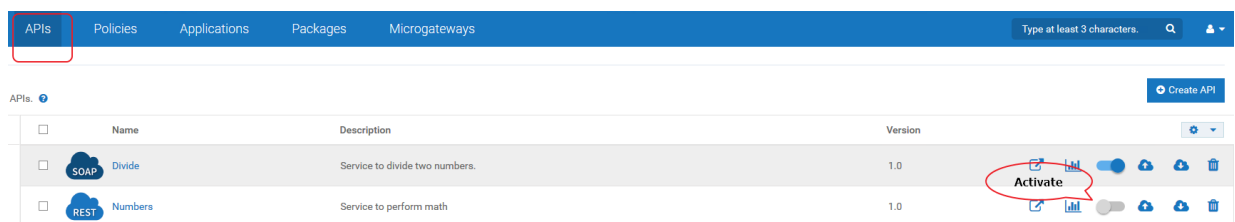


- Click the **Browse** button and select the **Numbers.swagger** located in: **C:\Training\E456C03-75E\Resources\Swagger**
- Name: **Numbers**
- Type: **Swagger** (Select the drop-down list to choose)
- Version: **1.0**
- Description: **Service to perform math.**

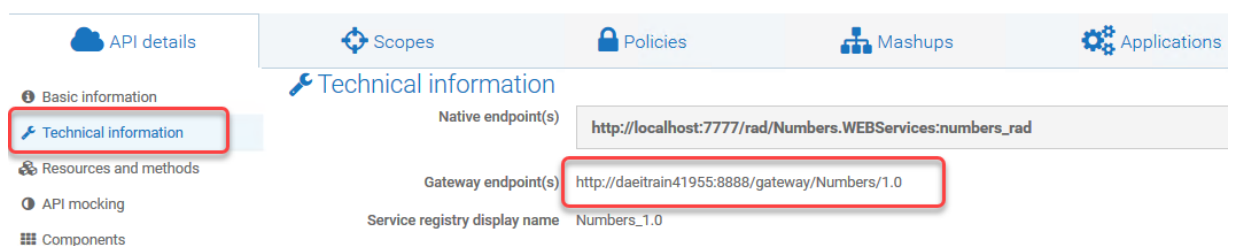
A screenshot of the 'Import API from file' form. The form has a title 'Import API from file' and a subtitle 'Create an API by importing API from a specified file.' Below the title, there is a 'Select file*' section with a text input field containing 'Numbers.swagger' and a 'Browse' button. To the right of the 'Select file*' section is a 'Description' section with a text input field containing 'Service to perform math'. Below the 'Select file*' section, there is a 'Name' section with a text input field containing 'Numbers'. Below the 'Name' section, there is a 'Type' section with a dropdown menu set to 'Swagger' and a 'Version' section with a text input field containing '1.0'. At the bottom of the form is a 'Create' button.

Select **Create**.

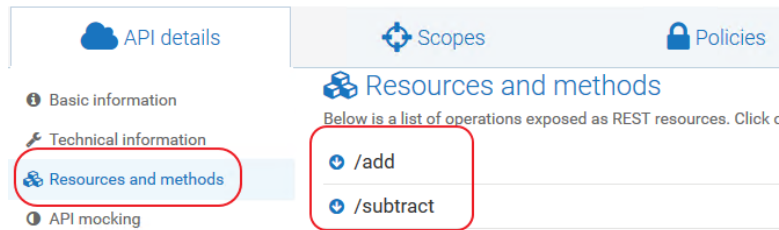
5. Activate the API. To do so, select the **APIs** tab, then select the slider bar to activate the **Numbers** API. Answer **Yes** to “Are you sure...”.



6. Select the **Numbers** API and navigate to **Technical information**. Copy the **Gateway endpoint** to the clipboard.

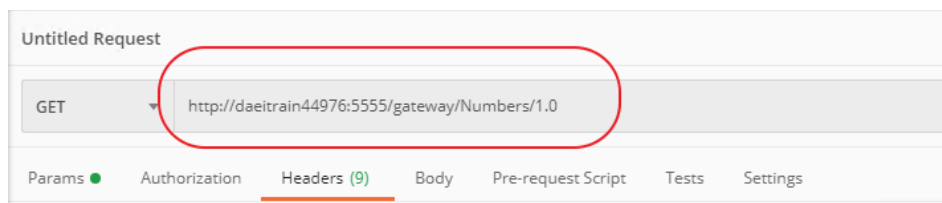


7. Click on **Resources and methods** and note the two resources **add** and **subtract**.



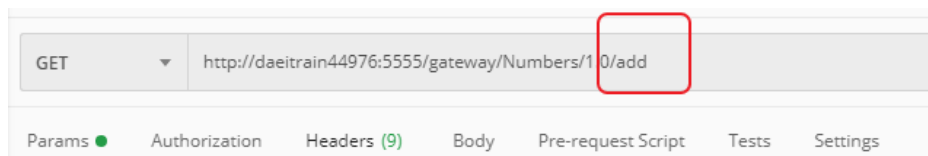
8. Test the **Numbers** API by using Postman:

- a) Open **Postman** and paste the Gateway URL you copied from the Numbers API as a URL of a GET request.



Note: Your hostname will be different than the example shown above.

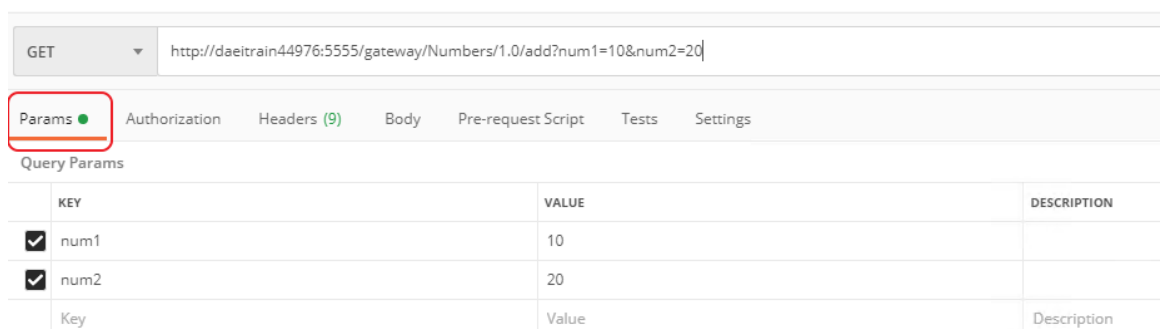
- b) Append **/add** to URL to invoke the add method in the Numbers API.



- c) Need to tell the API what numbers to add. In this example, we will add $10 + 20$. To do this, select the **Params** tab and add two new key/value pairs

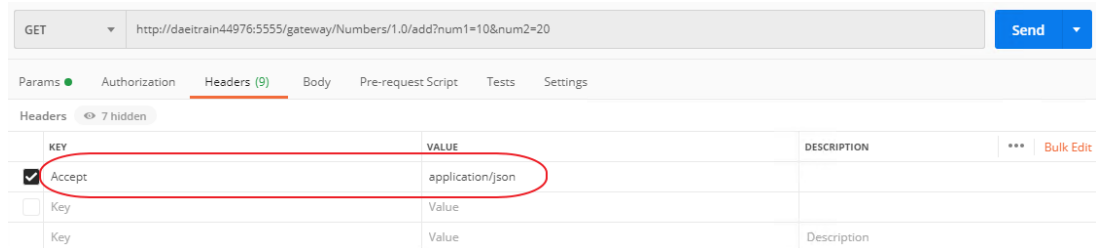
- i. KEY: **num1** VALUE: **10**
- ii. KEY: **num2** VALUE: **20**

Note: As you add the KEY/VALUE pair, note the URL – the values are added to the URL for you.

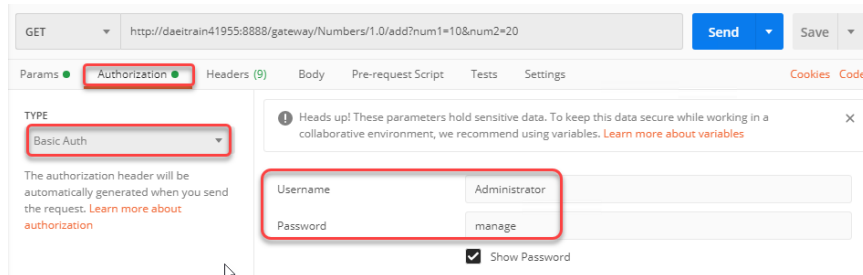


d) We want the response data to be JSON. On the Headers tab, add a new key/value pair:

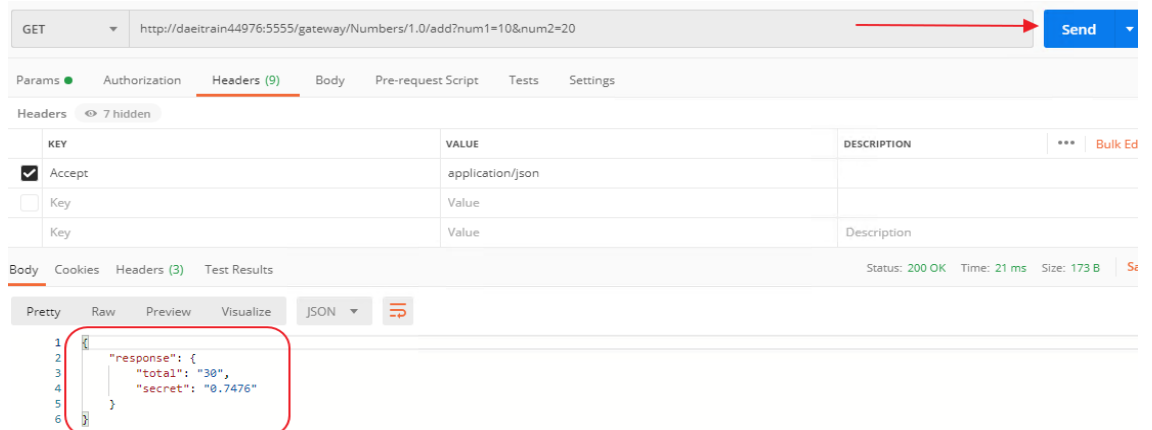
- i. KEY: **Accept**
- ii. VALUE: **application/json**



e) Because the native Numbers REST service at the internal IS requires for basic authentication, provide **Administrator | manage** as Username | Password on the **Authorization** tab with TYPE **Basic Auth** selected.



f) Submit the request by clicking the **Send** button.



g) We want the response data to be XML. On the Headers tab, adjust the following key/value pair:

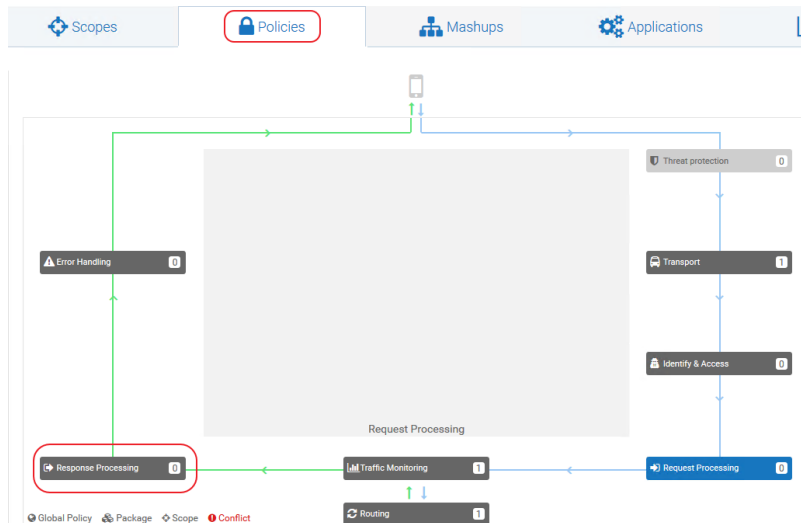
- i. KEY: **Accept**
- ii. VALUE: **application/xml**

h) Submit the request again by clicking the **Send** button.

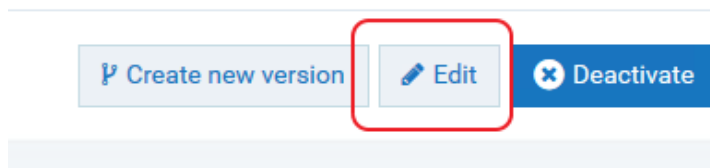
9. Note the response data. Both responses contain a value we were not expecting.
secret shouldn't be passed in the response!

Use Data Masking to filter out Secret and its value:

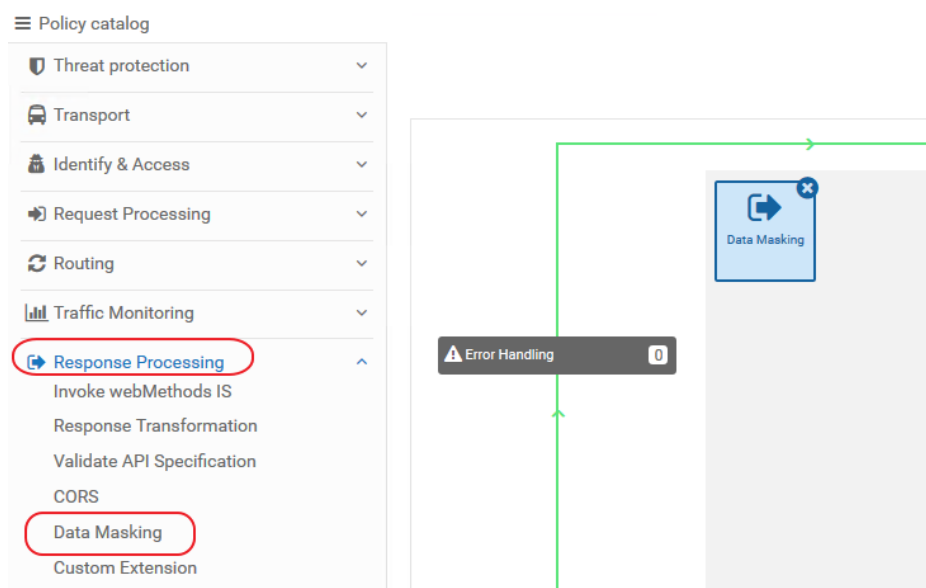
- a) Log in to the API Gateway as Sumala (**Sumala | manage**). Open the **Numbers** API and click on the **Policies** tab. Click on **Response Processing**.



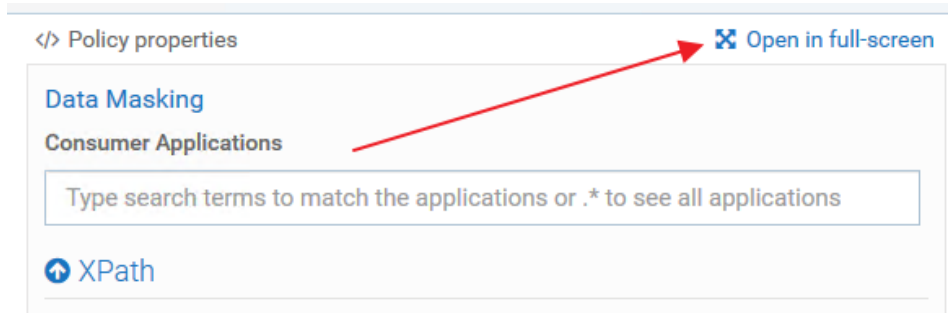
- b) To add a policy to the **API**, select **Edit** and select **Yes** to “Are you sure...”.



- c) In the **Response Processing** policy group, select **Data Masking**.

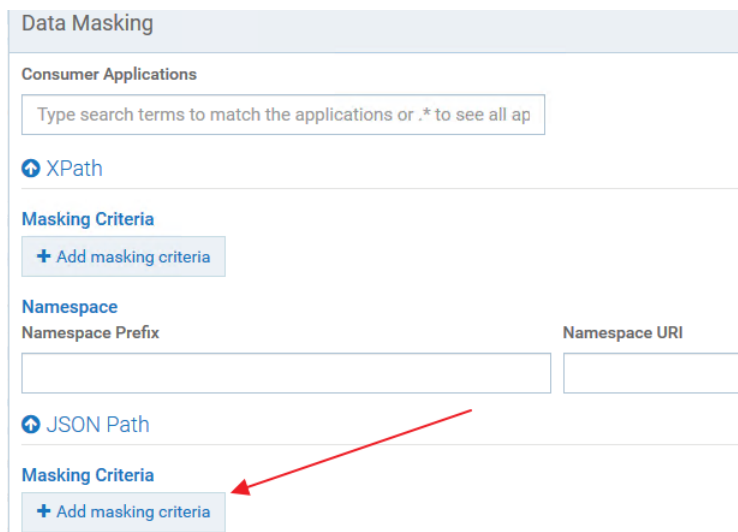


- d) On the right side of the screen, click on **Open in full-screen**.



Note: We will want to add two masks, one for XPath (XML), and one for JSON Path (json).

- e) Under **JSON Path**, select **Add masking criteria**.



- f) Enter the JSON path to the value we want to mask.

Note: For help on building the json path you may use <http://codebeautify.org/jsonpath-tester>.

Remember, the JSON response was:

```
{
  "response": {
    "total": "17",
    "secret": "0.7091"
  }
}
```

So the path to secret is response.secret.

- i. Query Expression: **\$.response.secret**
- ii. Masking Type: **Filter**

Select **Add**.

Masking Criteria

Add masking criteria

Query Expression*

Masking Type*

☐ Mask ☒ Filter

Cancel

Add

g) Once you select **Add**, the masking criteria is added:

JSON Path

Masking Criteria

Query Expression	Masking Type
\$.response.secret	Filter

h) Now add the masking criteria for XML. Under **XPath**, select **+Add masking criteria**.

XPath

Masking Criteria

+ Add masking criteria

Namespace

Namespace Prefix

i) Enter the XPath path to the value we want to mask.

Note: For help on building the XPath expressions you may want to use <https://www.freeformatter.com/xpath-tester.html>

Remember, the XML response was:

```
<?xml version="1.0"?>
<add_GET_response>
  <response>
    <total>17</total>
    <secret>0.8609</secret>
  </response>
</add_GET_response>
```

So, the XPath expression to <secret> is **//secret**.

- i. Query Expression: **//secret**
- ii. Masking Type: **Filter**

Select **Add**.

The screenshot shows a dialog box titled 'Add masking criteria'. It has two main sections: 'Query Expression*' and 'Masking Type*'. The 'Query Expression*' field contains the text '//secret'. The 'Masking Type*' section has two radio buttons: 'Mask' and 'Filter', with 'Filter' being selected. Below these fields are 'Cancel' and 'Add' buttons. A red arrow points to the 'Add' button.

- j) Select **Minimize**, then **Save** the changes.

The screenshot shows the API Gateway console. At the top, there are 'Cancel' and 'Save' buttons. Below them is a section for 'Analytics'. The main part of the screen shows 'Policy properties' with a 'Data Masking' policy applied to 'Consumer Applications'. There is an 'Open in full-screen' link.

10. Use Postman to accept JSON in the response and to verify the Data Masking:

- a) In Postman, change the **Accept** KEY/VALUE pair from **application/xml** back to **application/json**.
- b) **Send** the request again to API Gateway. Verify **secret** and a corresponding value is no longer present.

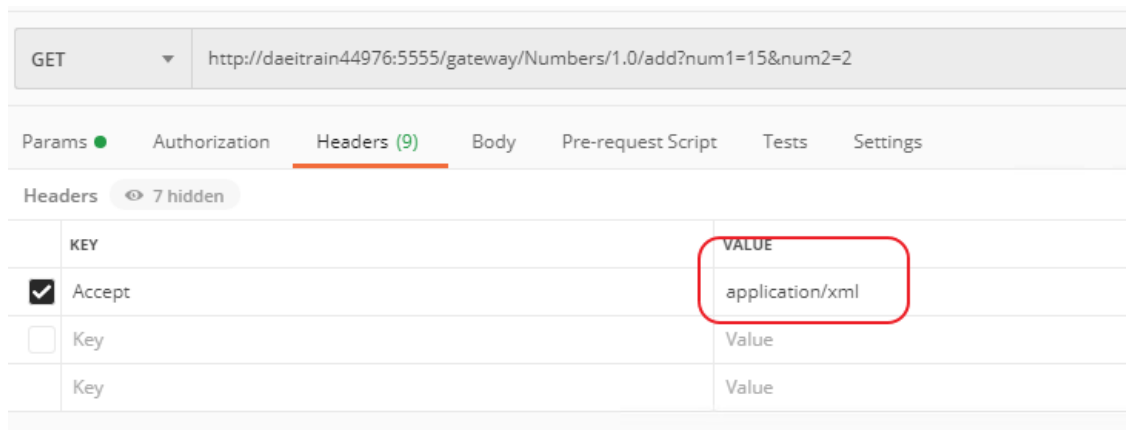
The screenshot shows a Postman interface. The top bar shows a GET request to 'http://daetrain44976:5555/gateway/Numbers/1.0/add?num1=10&num2=20'. The 'Headers' tab is selected, showing a table with 'KEY', 'VALUE', and 'DESCRIPTION' columns. The 'Accept' header is set to 'application/json'. The 'Body' tab is also selected, showing a JSON response: {'response': {'total': '30'}}. The status bar at the bottom shows 'Status: 200 OK', 'Time: 15 ms', and 'Size: 157 B'.

Note: If **secret** is still present verify the following:

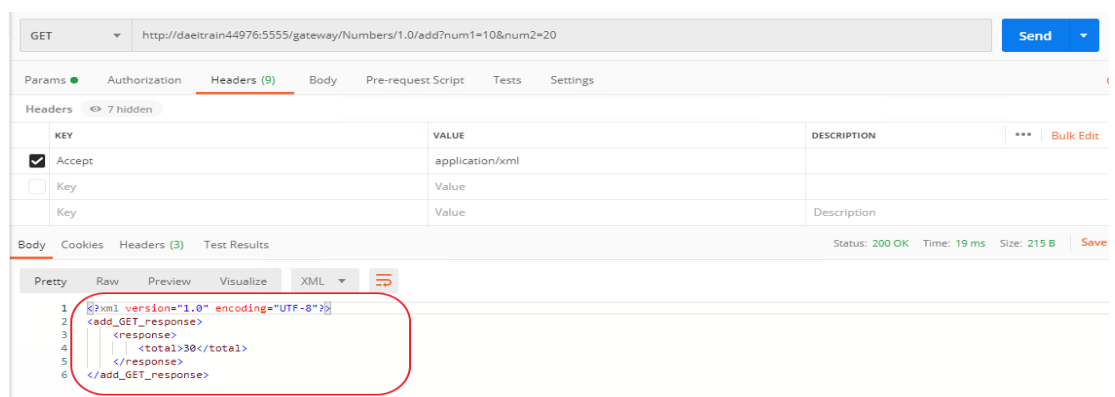
- The Numbers API was saved.
- The Data Masking policy is in the Response Processing - should NOT be in the Request Processing.

11. Use Postman to accept XML in the response and to verify the Data Masking:

- a) In Postman, change the **Accept** KEY/VALUE pair from **application/json** to **application/xml**.



- b) Select **Send**.

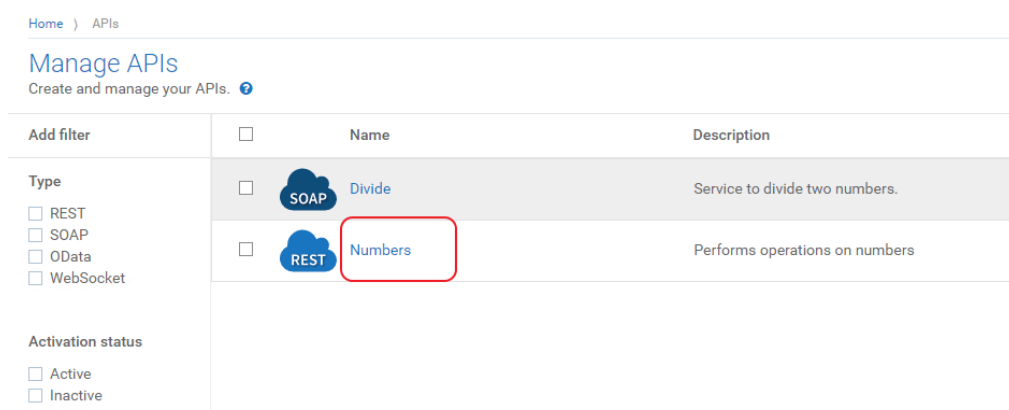


- c) Verify **secret** is not contained in the response.

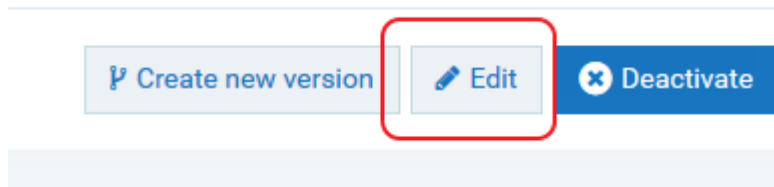
12. The native API owners of the Numbers API asked you to insert a custom value to the request header when sending data to the native API. They ask you to insert to the request header:

APIGateway=true

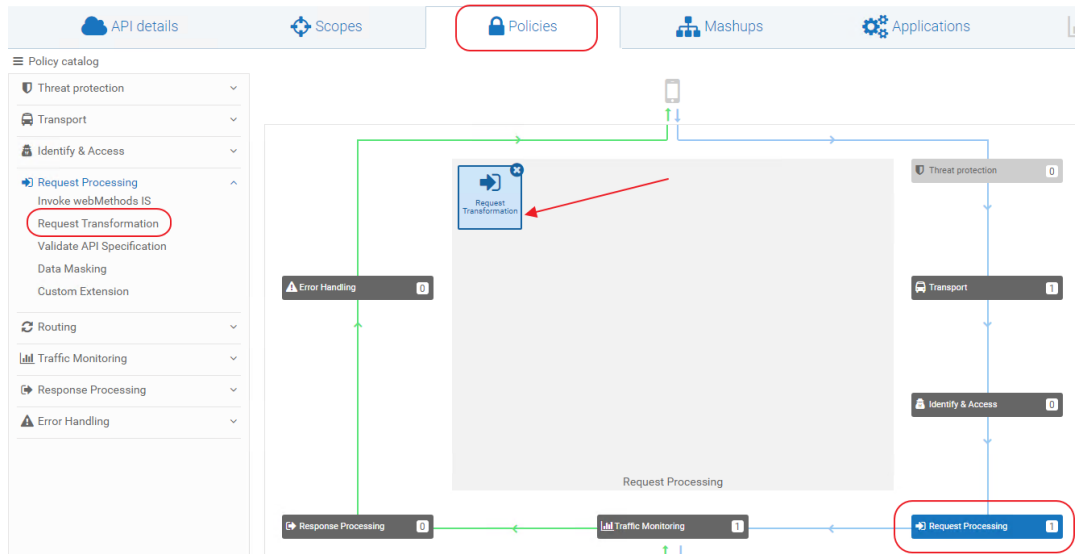
- a) Log into API Gateway as user Sumala (**Sumala | manage**). Click on the **Numbers** API.



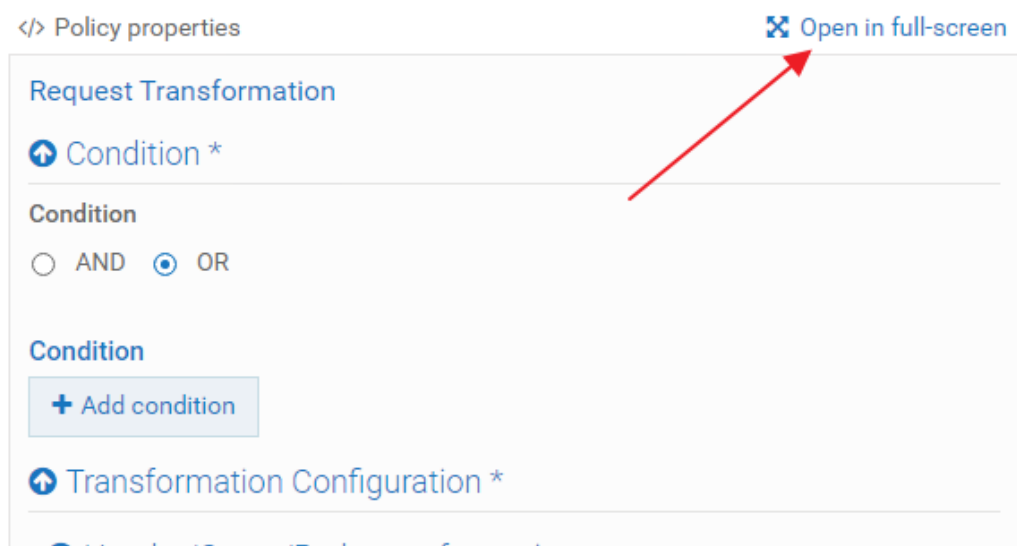
- b) Select **Edit**. Then select **Yes** to “Are you sure...”.



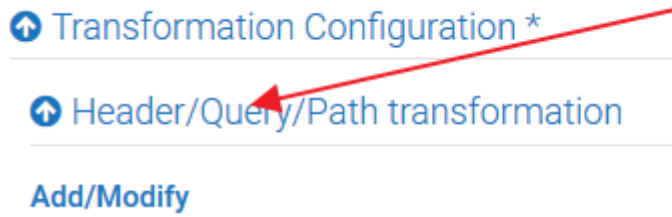
- c) Select Policies group **Request Processing**, then **Request Transformation**. A Request Transformation policy is added to the API.



- d) On the upper right side of the screen, select **Open in full-screen**.



e) Select the **Header/Query/Path transformation**.



Add the following:

- i. Variable: **`${request.headers.APIGateway}`**
- ii. Value: **true**

Transformation Configuration *

Header/Query/Path transformation

Add/Modify

Variable *

`${request.headers.APIGateway}`

Value *

true

+ Add

Remove

`${request.headers.Content-Type}`

Select **+ Add**.

f) Review the entries:

Transformation Configuration *

Header/Query/Path transformation

Add/Modify



Variable *

`${request.headers.Content-Type}`

Value *

`${request.payload.xpath[EXPRESSION]}`

+ Add

Variable *	Value *	Action
<code>\${request.headers.APIGateway}</code>	true	 

Remove

`${request.headers.Content-Type}`

Method Transformation

Payload Transformation

Advanced Transformation

Transformation Metadata

Namespace

Namespace Prefix

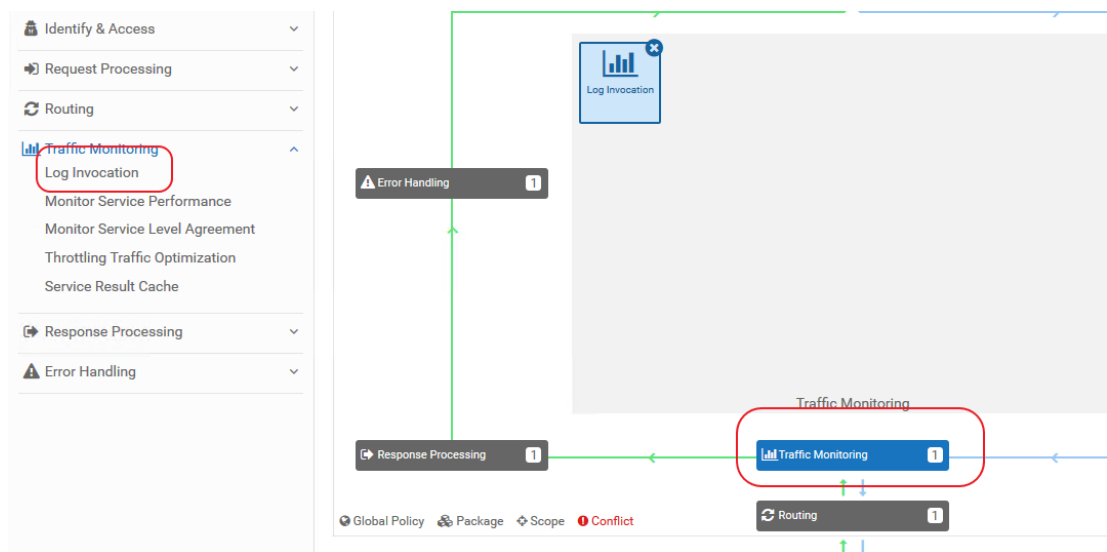
Namespace URI

+ Add

OK

Select **OK**.

- g) Select **Traffic Monitoring** to verify policy Log Invocation was added. If there is no Log Invocation, select **Log Invocation** under Traffic Monitoring:



- h) Ensure the following settings are made:

- i. Store Request Headers: *<selected>*
- ii. Store Request Payload: *<selected>*
- iii. Store Response Headers: *<selected>*
- iv. Store Response Payload: *<selected>*
- v. Compress Payload Data: *<not selected>*
- vi. Destination: **API Gateway**

Log Invocation

☒ Store Request Headers
 ☒ Store Request Payload
 ☒ Store Response Headers
 ☒ Store Response Payload
 ☐ Compress Payload Data

Log Generation Frequency*

Always

Destination *

☒ API Gateway

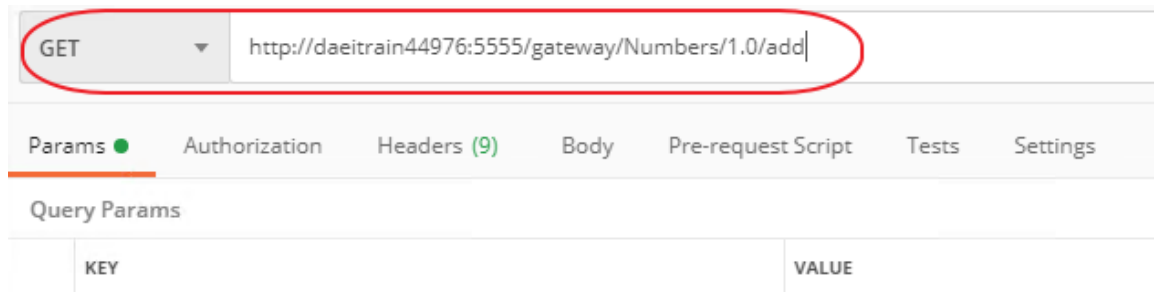
- i) Save all the changes made to the API by selecting **Save**.

13. Test the enhances made to the Numbers API by using Postman and Analytics:

- a) Open **Postman** by selecting the Postman icon in the toolbar, or by selecting the Windows Start button, type **Postman** and selecting Postman.
- b) Once Postman is started, enter the Numbers Gateway endpoint URL.

*Note: you can find the Numbers URL located in API Gateway under **APIs > Numbers > Technical Details**. Your hostname will vary from the screenshot.*

- c) Append **/add** to the URL. Ensure method **GET** is selected:

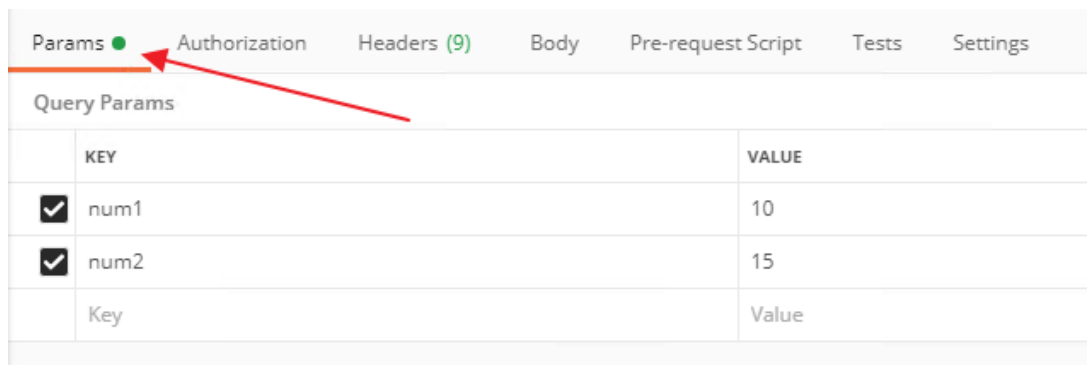


The screenshot shows the Postman interface. The method dropdown is set to **GET** and the URL is `http://daeitrain44976:5555/gateway/Numbers/1.0/add`. The **Params** tab is selected, showing a table for Query Params.

KEY	VALUE
-----	-------

- d) On the **Params** tab, add two KEY/VALUE pairs:

- i. **NUM1: 10**
- ii. **NUM2: 15**

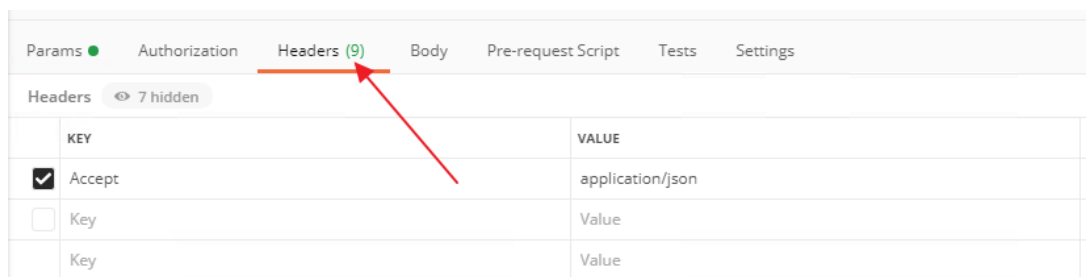


The screenshot shows the Postman interface with the **Params** tab selected. Two query parameters have been added:

KEY	VALUE
num1	10
num2	15
Key	Value

- e) On the **Headers** tab, add (if necessary) a KEY/VALUE pair:

Accept application/json



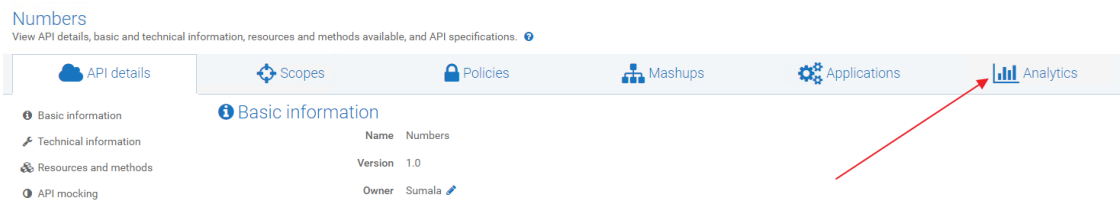
The screenshot shows the Postman interface with the **Headers** tab selected. One header has been added:

KEY	VALUE
Accept	application/json
Key	Value
Key	Value

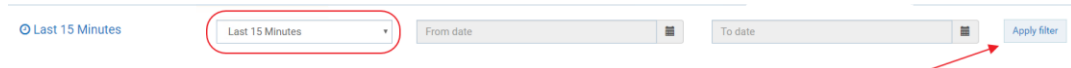
- f) On tab **Authorization** ensure **Administrator | manage** is set as Username | Password and TYPE **Basic Auth** is selected.

- g) Select **Send**.

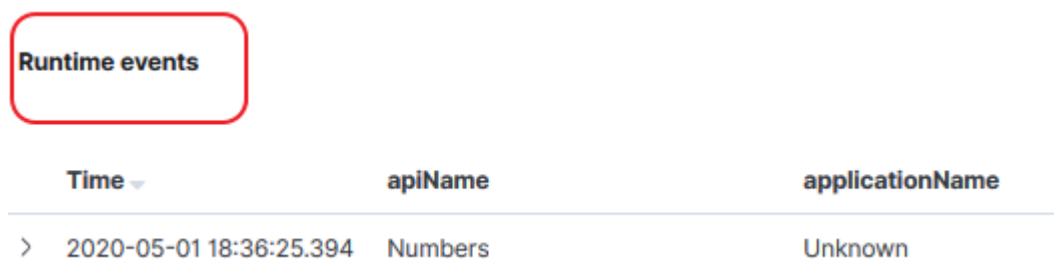
- h) Log into API Gateway as user **Sumala**, open the **Numbers** API and select the **Analytics** tab.



Change the time criteria to **Last 15 Minutes**, and select the **Apply** filter.



- i) Scroll to the bottom of the page till you locate **Runtime events**.



- j) Find the first **Transactional** event type. Open the details of the Transactional event by select the > symbol next to the event.

Runtime events

Time	apiName	applicationName	eventType	alertType	status
> 2020-04-29 20:14:52.474	Numbers	Unknown	Transactional	-	SUCCESS
> 2020-04-29 20:14:05.768	Numbers	Unknown	Transactional	-	SUCCESS
> 2020-04-29 20:11:16.697	Numbers	Unknown	Transactional	-	SUCCESS
> 2020-04-29 20:05:03.596	Numbers	Unknown	Transactional	-	SUCCESS
> 2020-04-29 20:03:50.514	Numbers	Unknown	Transactional	-	SUCCESS
> 2020-04-29 20:03:48.364	Numbers	Unknown	Transactional	-	SUCCESS
> 2020-04-29 20:03:39.121	Numbers	Unknown	Transactional	-	SUCCESS

k) Verify **nativeRequestHeaders.APIGateway** is set to **true**.

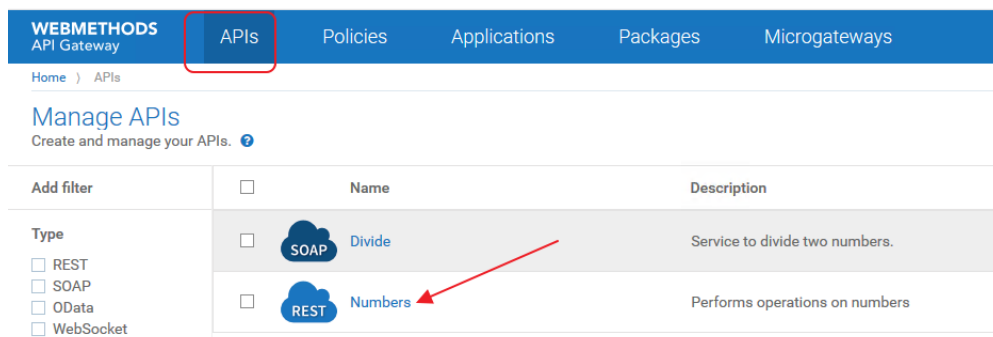
```
Runtime events
t callbackRequest false
t correlationID APIGW:7a68bc6b-6492-4a02-t
@ creationDate 2020-04-29 20:14:52.474
t eventType Transactional
? externalCalls {
  "externalCallType": "
  "externalURL": "http:
  "callStartTime": 1588
  "callEndTime": 158818
  "callDuration": 6,
  "responseCode": "200"
}
t httpMethod get
t nativeHttpMethod GET
t nativeReqPayload
t nativeRequestHeaders.APIGateway true
t nativeRequestHeaders.Accept application/json
```

Note: If you cannot locate the **nativeRequestHeaders.APIGateway**, verify the following:

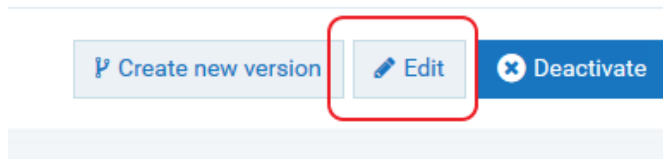
- You're looking at the latest Transactional event AND are NOT looking at a Performance event.
- The Numbers API was saved.
- The transformation was done on the request and not on the response.

14. In the Numbers API, you are asked to have the total response added to a custom header value called **ResponseTotal**. This will be done by adding a header value back to the consumer who called the API.

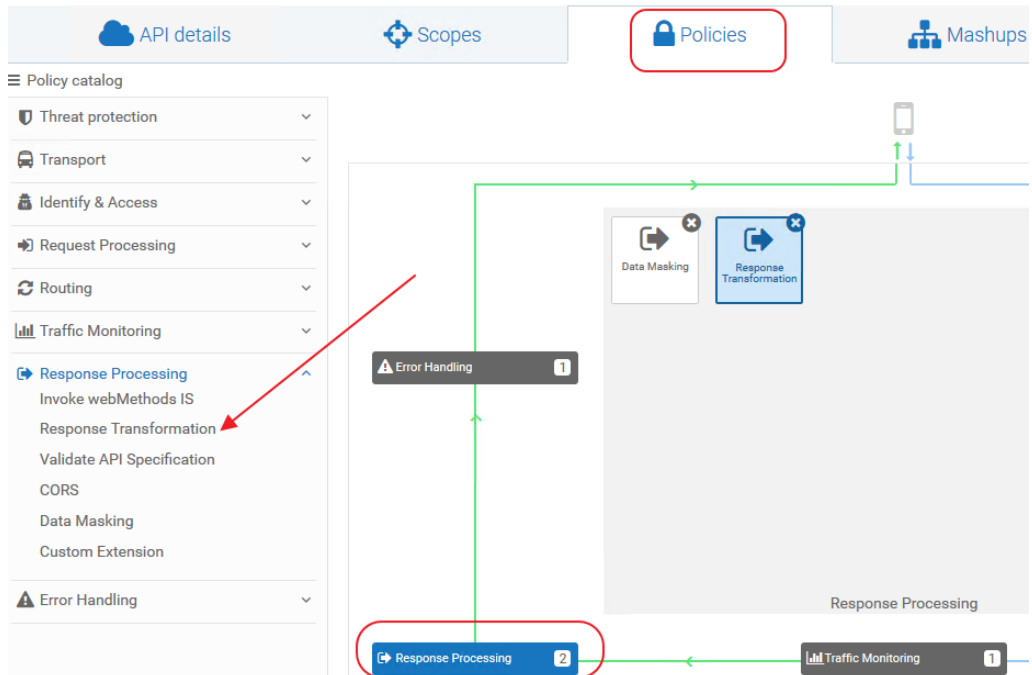
- a) Login to API Gateway using API Gateway Administrator Sumala (**Sumala | manage**).
- b) Select the APIs tab and click on the **Numbers** API.



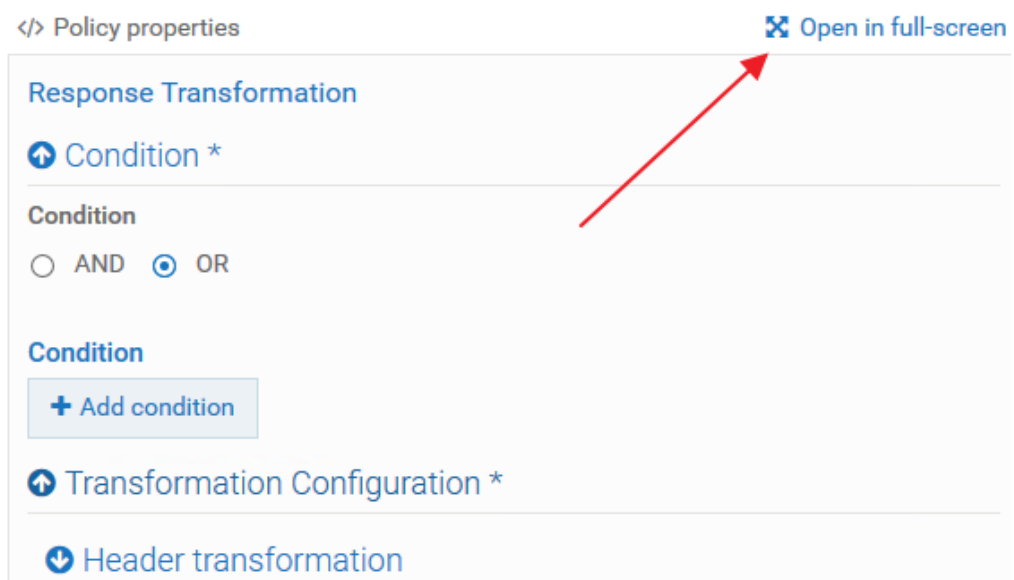
- c) Select **Edit** and select **Yes** to “Are you sure...”.



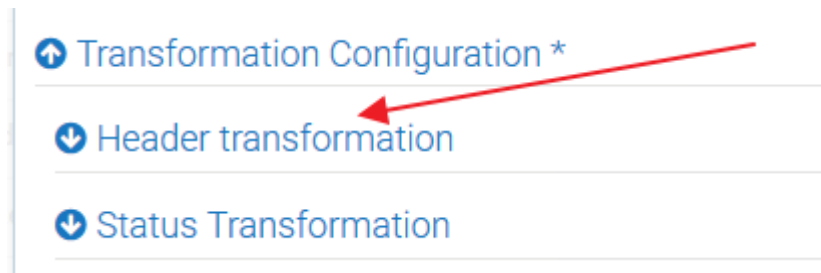
- d) Select the **Policies** tab, then policy group **Response Processing**. Select the **Response Transformation** policy.



- e) Select **Open in full-screen**.



f) Select **Header transformation**.

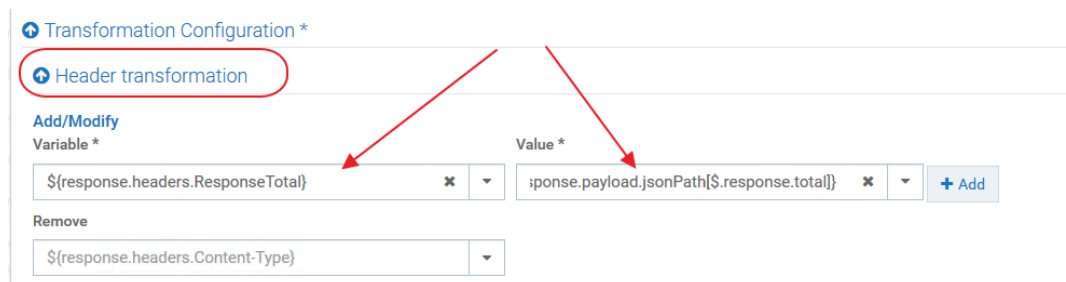


g) Type in the following:

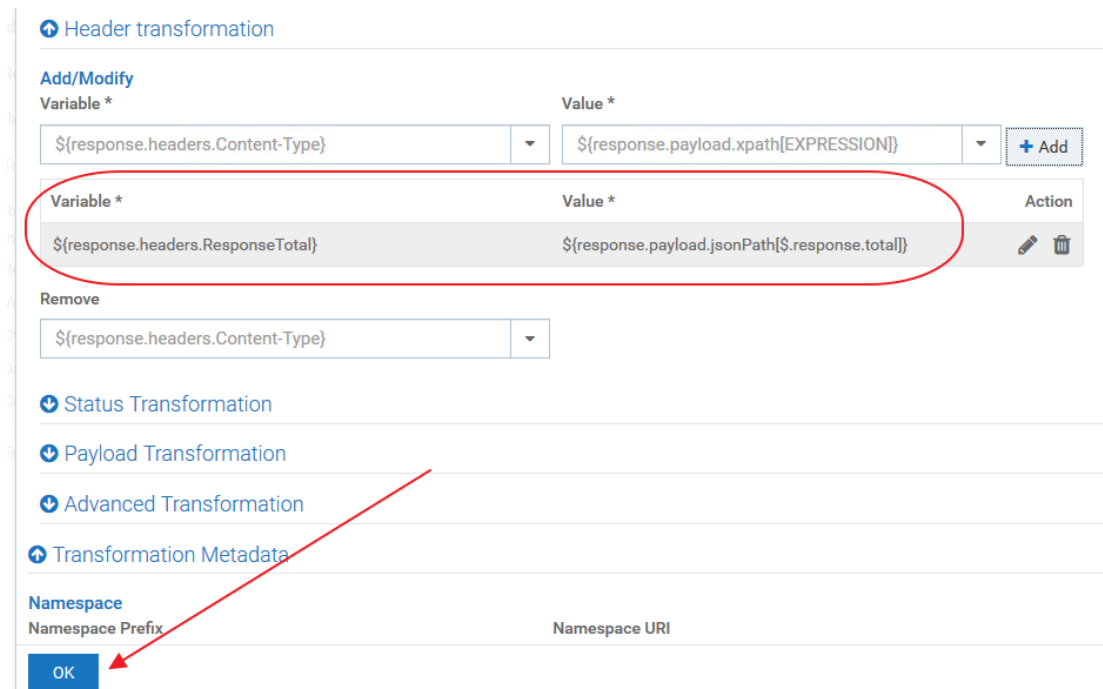
i. Variable: **`${response.headers.ResponseTotal}`**

ii. Value: **`${response.payload.jsonPath[.response.total]}`**

Select the **+Add** button.



h) Verify the data, then select **OK**.



i) Select **Save** to save the changes

15. Once the API is saved, you will test the changes. Switch back to **Postman**.

- a) Ensure the URL contains the Numbers API (you can always copy the URL from API Gateway > **Numbers** API > **Technical Details**). If necessary, append **/add** to the URL.
- b) On the **Params** tab, have two KEY/VALUE pairs, **num1 = 15**, and **num2 = 2**.

The screenshot shows the Postman interface with the 'Params' tab selected. The URL is `http://daeitrain44976:5555/gateway/Numbers/1.0/add?num1=15&num2=2`. The 'Query Params' section contains a table with the following data:

	KEY	VALUE
<input checked="" type="checkbox"/>	num1	15
<input checked="" type="checkbox"/>	num2	2
<input type="checkbox"/>	num3	4
	Key	Value

Below the table, there are tabs for 'Body', 'Cookies', 'Headers (4)', and 'Test Results'.

- c) Select **Send** to invoke the API.
- d) Verify the response total value is correct. Should be 17.

The screenshot shows the Postman interface with the 'Body' tab selected. The 'Query Params' section is the same as in the previous screenshot. The 'Body' section shows the JSON response in 'Pretty' format, with a red box highlighting the 'total' value:

```
1 {
2   "response": {
3     "total": "17"
4   }
5 }
```

Below the response, there are tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize', along with a 'JSON' dropdown and a refresh icon.

