

Fault Handler Updates

Introduction

The API Manager Fault Handler is designed to handle errors that may occur during transactions of user APIs published via API Manager. The capability allows users to customize the implementation of their own error handling; such as introduction of custom Security Policies and external integrations. For more details, please see https://docs.axway.com/bundle/axway-open-docs/page/docs/apim_administration/apimgr_admin/api_mgmt_custom_policies/index.html

The below document outlines, the current design and implementation, followed by the issues and in some cases bugs that our clients have observed in the implementation. The document will then do a walk through of the changes implemented in the February 26 Release and the benefits these changes will bring. Finally, the document will provide a list of before and after worked examples to clearly outline the changes along with some Breaking Changes that users may see in their error handling flow.

Design and Implementation

The API Gateway via Policy Studio allows for the creation of REST APIs that are exposed on endpoints that when hit can trigger a policy flow, this is outlined in detail in [Develop REST APIs in Policy Studio](#) and [Introduction to policies](#), with the API Gateway REST API Fault Handling implementation employing [the out-of-box API Gateway policy fault handling workflow](#).

The API Manager policies workflow [https://docs.axway.com/bundle/axway-open-docs/page/docs/api_mgmt_overview/key_concepts/policies/index.html#api-manager-policies] is an extension of this allowing for deeper and richer features; these include the well-known API flow options of Global Request , Global Response, API Request , API Response, Fault Handler Policy. Details on configuration of these policies can be found here - [Configure API Manager custom policies](#).

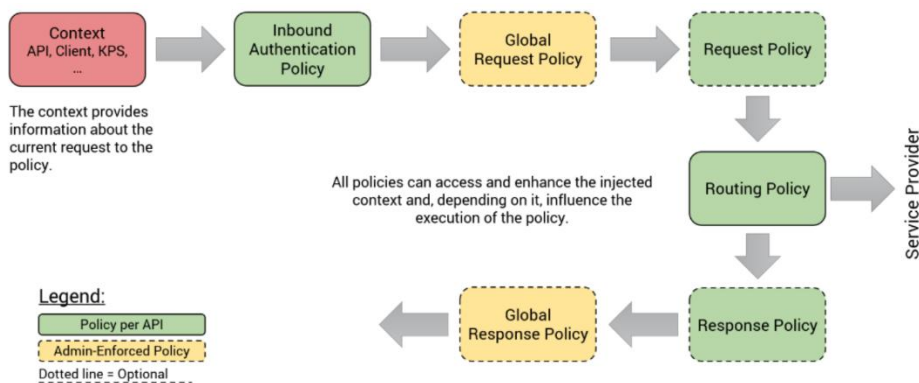


Fig 1. Process of an API request processed at runtime by the API Manager

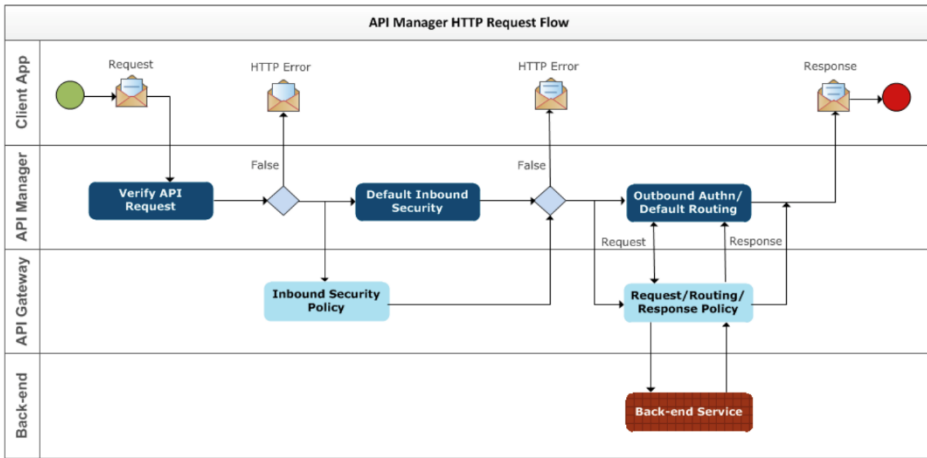


Fig 2. Detailed API Manager process flow

The API Manager Global Fault Handler; default or custom, is configured to process errors for a API request when an error or exception occurs during the API Manager runtime API invocation. An error is generally triggered in two scenarios; when a HTTP Error is generated or when a Back-end Service responds with non-2xx response code.



Fig 3. API Manager Fault handler scope

Known Issues

- With the enhanced complexity and feature set of API Manager, the policy execution flow between API Manager and API Gateway diverged. This divergence caused differences in the behaviour of API Manager's API Broker and API Gateway's REST APIs' API Broker. This led to successful messages displaying different Traffic Monitor filter execution flows and orders.
- A particular scenario of above; and the prime driver of this change, is in the case of Fault Handling invocation during the creation of API Service Contexts showing inconsistencies in Transaction and Traffic Monitor logging. This leads to incorrect and at times misleading representations of the transactions in the API Gateway Manager Traffic Monitor and API Gateway Analytics Metrics database. The error handling workflow in API Manager has multiple branches that handle possible `ApiShunt` errors differently depending on API Manager fault handler configuration. This complicates various areas such as; the test coverage, maintaining consistent error handling; and addressing reported issues.

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Set service context	✓		0
API Proxy	✓		16
Default API Proxy Routing			
Default Routing	✓		16
Default API Proxy Routing			
not-required-here	✓		15
Default Profile-based Routing			
Connect to URL	✓		15
API Broker			
Set service context	✓		0
Backend Error	⚠	Backend Error	0
Fault Handler Policy	✓		17
Custom API Fault Handler			
Forward the error	✓		17

Fig 3. API Manager Fault Handler example where an error is triggered on a backend Response. An initial check on the Traffic monitor would indicate a successful API Broker transaction and have to follow up on a secondary API Broker flow to check on the error.

- The customisation of the Fault Handling mechanism also shows scenarios with unexpected behaviour, for example, invoking fault handlers twice.
- OPTIONS requests processing in API Manager has incorrectly diverged from API Gateway in some cases. This contributed to the problems in Error Handling that is expected to be identical in API Gateway and API Manager. For example, incoming client CORS requests to API Manager, did not align with API Gateway.

Important Changes

The API Manager broker invocation has the following important changes:

- The API Broker policy execution path is now consistent and accurate

- Inbound Security custom policy fault handling will only invoke a fault handler defined in API Manager, matching the diagram **Fig 3**
- API Manager CORS request processing aligned with API Gateway CORS, responses are now more correctly inline with the RFC requirements

Benefits of the Change

The benefits to the customer for this change are;

- Consistency in the handling of errors between API manager and API Gateway policies workflow allowing API Analytics and Traffic Monitor correct presentation of transactions, API Broker policy execution path
 - Consistent execution filter paths, matching the flow shown on diagram **Fig 1**.
 - consistent level of the fault handler invocation
 - trivial parsing of the Traffic Monitor data
 - simple analysis of successes and failures in Traffic Monitor
 - always starts with single node "API Broker"

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	Backend Error	51
API method			
Context Validation	✓		1
API Proxy	✓		30
Default API Proxy Routing			
Default Routing	✓		30
Default API Proxy Routing			
not-required-here	✓		29
Default Profile-based Routing			
Connect to URL	✓		29
Validate backend API response	⚠		1
Fault Handler Policy	✓		23
Custom API Fault Handler			
Forward the error	✓		23

Fig 3. API Manager Fault Handler example an error is triggered on a backend API response. Traffic Monitor now correctly indicates a single and unsuccessful API Broker transaction with Transaction Audit Message of Backend Error.

- Fault handlers always have access to the "api.error.source" and "api.error.reason" message attributes
 - see [Fault handler generated errors](#)
- Custom fault handlers can override the default final transaction status On-Demand
 - Please see for further details - [Set response status filter](#)
 - The status reflected in the API Gateway Manager Traffic Monitoring, Analytics Traffic Monitoring and API Manager Monitoring accordingly
- API Manager uses less memory footprint for API Broker circuits
- Accurate API Broker filters timing reflected in Traffic Monitor

Runtime Examples

API Broker Success and Failure Scenarios

Below are several examples showing success and failure scenarios along the difference in behaviour in the Traffic Monitor Execution Path of API Manager and API Gateway Transactions before and after the February 2026 Release.

In the setup of these examples, the following was configured

- A default **API Key Inbound Security Device** where the application Id is 3bde5ea8-ff29-4f38-8fb5-29442716e260
- **Custom Fault Handler** that preserves the transaction status and the HTTP response code/status set by API Manager

API Manager Broker Invocation

API Shunt	Response	TM Filter Execution Path of API Manager Transaction Pre the February 26 Release	TM Filter Execution Path of API Manager Transaction Post the February 26 Release	Changes																																																																																																												
n/a	200 OK	<div data-bbox="386 347 1039 669"> <p>FILTER EXECUTION PATH</p> <table border="1"> <thead> <tr> <th>Filter</th> <th>Status</th> <th>Transaction Audit Message</th> <th>Execution Time (ms)</th> </tr> </thead> <tbody> <tr> <td>API Broker</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Set service context</td> <td>✓</td> <td></td> <td>0</td> </tr> <tr> <td> API Proxy</td> <td>✓</td> <td></td> <td>9</td> </tr> <tr> <td> Default API Proxy Routing</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Default Routing</td> <td>✓</td> <td></td> <td>8</td> </tr> <tr> <td> Default API Proxy Routing</td> <td></td> <td></td> <td></td> </tr> <tr> <td> not-required-here</td> <td>✓</td> <td></td> <td>7</td> </tr> <tr> <td> Default Profile-based Routing</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Connect to URL</td> <td>✓</td> <td></td> <td>7</td> </tr> </tbody> </table> </div> <div data-bbox="386 711 1039 1328"> <table border="1"> <tr> <td>Error Trace</td> <td>n/a</td> </tr> <tr> <td>circuit.failure.reason</td> <td>n/a</td> </tr> <tr> <td>Transaction Audit Message</td> <td>none</td> </tr> <tr> <td>Service Context</td> <td>{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 14 }</td> </tr> </table> </div>	Filter	Status	Transaction Audit Message	Execution Time (ms)	API Broker				Set service context	✓		0	API Proxy	✓		9	Default API Proxy Routing				Default Routing	✓		8	Default API Proxy Routing				not-required-here	✓		7	Default Profile-based Routing				Connect to URL	✓		7	Error Trace	n/a	circuit.failure.reason	n/a	Transaction Audit Message	none	Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 14 }	<div data-bbox="1064 347 1717 669"> <p>FILTER EXECUTION PATH</p> <table border="1"> <thead> <tr> <th>Filter</th> <th>Status</th> <th>Transaction Audit Message</th> <th>Execution Time (ms)</th> </tr> </thead> <tbody> <tr> <td>API Broker</td> <td></td> <td></td> <td></td> </tr> <tr> <td> API Transaction</td> <td>✓</td> <td>Success</td> <td>18</td> </tr> <tr> <td> API method</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Context Validation</td> <td>✓</td> <td></td> <td>1</td> </tr> <tr> <td> API Proxy</td> <td>✓</td> <td></td> <td>7</td> </tr> <tr> <td> Default API Proxy Routing</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Default Routing</td> <td>✓</td> <td></td> <td>6</td> </tr> <tr> <td> Default API Proxy Routing</td> <td></td> <td></td> <td></td> </tr> <tr> <td> not-required-here</td> <td>✓</td> <td></td> <td>5</td> </tr> <tr> <td> Default Profile-based Routing</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Connect to URL</td> <td>✓</td> <td></td> <td>5</td> </tr> <tr> <td> Validate backend API response</td> <td>✓</td> <td></td> <td>0</td> </tr> </tbody> </table> </div> <div data-bbox="1064 711 1717 1328"> <table border="1"> <tr> <td>Error Trace</td> <td>n/a</td> </tr> <tr> <td>circuit.failure.reason</td> <td>n/a</td> </tr> <tr> <td>Transaction Audit Message</td> <td>Success</td> </tr> <tr> <td>Service Context</td> <td>{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 20 }</td> </tr> </table> </div>	Filter	Status	Transaction Audit Message	Execution Time (ms)	API Broker				API Transaction	✓	Success	18	API method				Context Validation	✓		1	API Proxy	✓		7	Default API Proxy Routing				Default Routing	✓		6	Default API Proxy Routing				not-required-here	✓		5	Default Profile-based Routing				Connect to URL	✓		5	Validate backend API response	✓		0	Error Trace	n/a	circuit.failure.reason	n/a	Transaction Audit Message	Success	Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 20 }	<ul style="list-style-type: none"> The API Broker now reports Transaction Audit Message as "Success"
Filter	Status	Transaction Audit Message	Execution Time (ms)																																																																																																													
API Broker																																																																																																																
Set service context	✓		0																																																																																																													
API Proxy	✓		9																																																																																																													
Default API Proxy Routing																																																																																																																
Default Routing	✓		8																																																																																																													
Default API Proxy Routing																																																																																																																
not-required-here	✓		7																																																																																																													
Default Profile-based Routing																																																																																																																
Connect to URL	✓		7																																																																																																													
Error Trace	n/a																																																																																																															
circuit.failure.reason	n/a																																																																																																															
Transaction Audit Message	none																																																																																																															
Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 14 }																																																																																																															
Filter	Status	Transaction Audit Message	Execution Time (ms)																																																																																																													
API Broker																																																																																																																
API Transaction	✓	Success	18																																																																																																													
API method																																																																																																																
Context Validation	✓		1																																																																																																													
API Proxy	✓		7																																																																																																													
Default API Proxy Routing																																																																																																																
Default Routing	✓		6																																																																																																													
Default API Proxy Routing																																																																																																																
not-required-here	✓		5																																																																																																													
Default Profile-based Routing																																																																																																																
Connect to URL	✓		5																																																																																																													
Validate backend API response	✓		0																																																																																																													
Error Trace	n/a																																																																																																															
circuit.failure.reason	n/a																																																																																																															
Transaction Audit Message	Success																																																																																																															
Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 20 }																																																																																																															

noMethodMatched

403 No Match For Request

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Set service context	✓		0
No Match For Request	⚠	No Match For Request	0
Fault Handler Policy	✓		0
Custom Fault Handler			
Forward the error	✓		0

Trace	<pre>Filter that caused failure: No Match For Request Policy 'API Broker' { Filter 'Set service context' Status: PASSED Filter 'No Match For Request' Status: FAILED } No Match For Request filter failed</pre>
circuit.failure.reason	No Match For Request filter failed
Transaction Audit Message	No Match For Request
Service Context	<pre>{"service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": null, "status": "failure", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 9}</pre>

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	No match found for request	3
Fault Handler Policy	✓		1
Custom API Fault Handler			
Forward the error	✓		1

Trace	<pre>Filter that caused failure: API Transaction Policy 'API Broker' { Filter 'API Transaction' Status: FAILED } API Transaction filter failed</pre>
circuit.failure.reason	API Transaction filter failed
Transaction Audit Message	No match found for request
Service Context	<pre>{"service": "My API test", "client": null, "method": null, "status": "failure", "monitor": true, "org": null, "app": null, "duration": 5}</pre>

- The API Broker now accurately displays the execution time.
- API Broker now presents a consistent view of the filter names and their flows.

tooManyRequests

429 Too Many Requests

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Set service context	✓		0
Too Many Requests	⚠	Too Many Requests	0
Fault Handler Policy	✓		0
Custom Fault Handler			
Forward the error	✓		0

Trace	Filter that caused failure: Too Many Requests Policy 'API Broker' { Filter 'Set service context' Status: PASSED Filter 'Too Many Requests' Status: FAILED }
circuit.failure.reason	Too Many Requests filter failed
Transaction Audit Message	Too Many Requests
Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "failure", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 11 }

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	Too Many Requests	3
Fault Handler Policy	✓		0
Custom API Fault Handler			
Forward the error	✓		0

Trace	Filter that caused failure: API Transaction Policy 'API Broker' { Filter 'API Transaction' Status: FAILED }
circuit.failure.reason	API Transaction filter failed
Transaction Audit Message	Too Many Requests
Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "healthcheck", "status": "failure", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 6 }

- The API Broker now accurately displays the execution time.
- API Broker now presents a consistent view of the filter names and their flows.

allAuthNDevicesFailed

401 Security failure

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Set service context	✓		0
Authentication Failure	⚠	Authentication Failure	0
Fault Handler Policy	✓		0
Custom Fault Handler			
Forward the error	✓		0

Trace	Filter that caused failure: Authentication Failure Policy 'API Broker' { Filter 'Set service context' Status: PASSED Filter 'Authentication Failure' Status: FAILED }
circuit.failure.reason	Authentication Failure filter failed
Transaction Audit Message	Authentication Failure
Service Context	{ "service": "My API test", "client": null, "method": "healthcheck", "status": "failure", "monitor": true, "org": null, "app": null, "duration": 8 }

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	Authentication Failure	3
Fault Handler Policy	✓		0
Custom API Fault Handler			
Forward the error	✓		0

Trace	Filter that caused failure: API Transaction Policy 'API Broker' { Filter 'API Transaction' Status: FAILED }
circuit.failure.reason	API Transaction filter failed
Transaction Audit Message	Authentication Failure
Service Context	{ "service": "My API test", "client": null, "method": "healthcheck", "status": "failure", "monitor": true, "org": null, "app": null, "duration": 10 }

- The API Broker now accurately displays the execution time.
- API Broker now presents a consistent view of the filter names and their flows.

noApiM
atched

404 Not
Found

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Not Found	⚠	Not Found	0
Fault Handler Policy	✓		5
Custom API Fault Handler			
Forward the error	✓		5

Trace	Filter that caused failure: Not Found Policy 'API Broker' { Filter 'Not Found' Status: FAILED } Not Found filter failed
circuit.failure.reason	Not Found filter failed
Transaction Audit Message	Not Found
Service Context	n/a

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	Not Found	0
Fault Handler Policy	✓		1
Custom API Fault Handler			
Forward the error	✓		0

Trace	Filter that caused failure: API Transaction Policy 'API Broker' { Filter 'API Transaction' Status: FAILED } API Transaction filter failed
circuit.failure.reason	API Transaction filter failed
Transaction Audit Message	Not Found
Service Context	n/a

- The API Broker now accurately displays the execution time.
- API Broker now presents a consistent view of the filter names and their flows.
- Decreased memory allocations observed in failed transactions.

backendError

400 Bad Request

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Set service context	✓		0
API Proxy	✓		16
Default API Proxy Routing			
Default Routing	✓		16
Default API Proxy Routing			
not-required-here	✓		15
Default Profile-based Routing			
Connect to URL	✓		15
API Broker			
Set service context	✓		0
Backend Error	⚠	Backend Error	0
Fault Handler Policy	✓		17
Custom API Fault Handler			
Forward the error	✓		17

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	Backend Error	51
API method			
Context Validation	✓		1
API Proxy	✓		30
Default API Proxy Routing			
Default Routing	✓		30
Default API Proxy Routing			
not-required-here	✓		29
Default Profile-based Routing			
Connect to URL	✓		29
Validate backend API response	⚠		1
Fault Handler Policy	✓		23
Custom API Fault Handler			
Forward the error	✓		23

Trace

```
Filter that caused failure: Backend Error
Policy 'API Broker' {
  Filter 'Set service context' Status: PASSED
  Filter 'API Proxy' Status: PASSED
  Policy 'Default API Proxy Routing' {
    Filter 'Default Routing' Status: PASSED
    Filter 'not-required-here' Status: PASSED
    Policy 'Default Profile-based Routing' {
      Filter 'Connect to URL' Status: PASSED
    }
  }
  Policy 'API Broker' {
    Filter 'Set service context' Status: PASSED
    Filter 'Backend Error' Status: FAILED
  }
}
```

Trace

```
Filter that caused failure: Validate backend API response
Policy 'API Broker' {
  Filter 'API Transaction' Status: FAILED
  Policy 'API method' {
    Filter 'Context Validation' Status: PASSED
    Filter 'API Proxy' Status: PASSED
    Policy 'Default API Proxy Routing' {
      Filter 'Default Routing' Status: PASSED
      Filter 'not-required-here' Status: PASSED
      Policy 'Default Profile-based Routing' {
        Filter 'Connect to URL' Status: PASSED
      }
    }
    Filter 'Validate backend API response' Status: FAILED
  }
}
```

- The Service Context now reports the status accurately as a failure.
- Custom fault handlers can now override the transaction status to set it to "success" on demand by invoking the "Set Response Status" filter configured as "Pass", which makes sense when the HTTP response status is also set to 200 OK (by invoking the "HTTP Status Code" filter).
- The API Broker now accurately displays the execution time.
- API Broker now presents a consistent view of the filter

		Backend Error filter failed		} }	names and their flows.
	circuit.failure.reason	Backend Error filter failed		Validate backend API response filter failed	
	Transaction Audit Message	Backend Error	circuit.failure.reason	Validate backend API response filter failed	
	Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "400", "status": "success", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 31}	Transaction Audit Message	Backend Error	
			Service Context	{ "service": "My API test", "client": "3bde5ea8-ff29-4f38-8fb5-29442716e260", "method": "400", "status": "failure", "monitor": true, "org": "API Development", "app": "Test APP 1", "duration": 75}	

API Gateway REST API Broker Invocation

API Shunt	Response	TM Filter Execution Path of API Manager Transaction	TM Filter Execution Path of API Manager Transaction	Changes
		Pre the February 26 Release	Post the February 26 Release	

n/a

200 OK

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
healthcheck			
Set service context	✓		0
Routing	✓		4
Health Check			
Set Message	✓	Success in setting the message	0
Reflect	✓		0

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	✓	Success	2
healthcheck			
Set service context	✓		0
Context Validation	✓		0
Routing	✓		0
Health Check			
Set Message	✓		0
Reflect	✓		0

Source	Value
Error Trace	n/a
circuit.failure.reason	n/a
Transaction Audit Message	Success in setting the message (*) (*) The message is not API specific, inconsistent and specific to a filter of current Routing policy. And depending on filter implementation it can be unset (e.g. see Reflect filter in the example above).
Service Context	{ "service": "My Test API", "client": null, "method": "healthcheck", "status": "success", "monitor": true, "org": null, "app": null, "duration": 34 }

Source	Value
Error Trace	n/a
circuit.failure.reason	n/a
Transaction Audit Message	Success
Service Context	{ "service": "My Test API", "client": null, "method": "healthcheck", "status": "success", "monitor": true, "org": null, "app": null, "duration": 2 }

- The API Broker now reports Transaction Audit Message as "Success"
- The "Context Validation" filter is introduced for transparency of policy execution time and failures.

noMethodMatched

404 No match found for request

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
No Match For Request	⚠	No Match For Request	0
No Match For Request	✓		6

Source	Value
Error Trace	Filter that caused failure: No Match For Request Policy 'API Broker' { Filter 'Set service context' Status: PASSED Filter 'No Match For Request' Status: FAILED } No Match For Request filter failed
circuit.failure.reason	No Match For Request filter failed
Transaction Audit Message	No Match For Request
Service Context	n/a

Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	No match found for request	1
Internal API Fault handler	✓	No match found for request	19

Source	Value
Error Trace	Filter that caused failure: API Transaction Policy 'API Broker' { Filter 'API Transaction' Status: FAILED } API Transaction filter failed
circuit.failure.reason	API Transaction filter failed
Transaction Audit Message	No match found for request
Service Context	n/a

- Clearer and standardized filter execution path when an error happened prior API method policy invocation.

invokableMethodParameterFailure

400 Bad Request

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
Set service context	✓		0
Bad Request	⚠	Bad Request	0
Bad Request	✓		31

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	⚠	Bad Request	6
healthcheck			
Set service context	✓		0
Context Validation	⚠		2
Internal API Fault handler	✓	Bad Request	28

Source	Value
Error Trace	Filter that caused failure: Bad Request Policy 'API Broker' { Filter 'Set service context' Status: PASSED Filter 'Bad Request' Status: FAILED } Bad Request filter failed
circuit.failure.reason	Bad Request filter failed
Transaction Audit Message	Bad Request
Service Context	{"service":"My Test API", "client":null, "method":"healthcheck", "status":"failure", "monitor":true, "org":null, "app":null, "duration":42}

Source	Value
Error Trace	Filter that caused failure: Context Validation Policy 'API Broker' { Filter 'API Transaction' Status: FAILED Policy 'healthcheck' { Filter 'Set service context' Status: PASSED Filter 'Context Validation' Status: FAILED } } Context Validation filter failed
circuit.failure.reason	Context Validation filter failed
Transaction Audit Message	Bad Request
Service Context	{"service":"My Test API", "client":null, "method":"healthcheck", "status":"failure", "monitor":true, "org":null, "app":null, "duration":34}

- Clearer and standardized filter execution path when an error happens during API method policy invocation.

Scenarios when API Broker's process CORS Requests

This section describes the differences processing API Manager and API Gateway CORS requests by API Broker

Default API Key inbound security is configured.

API Manager REST API with default CORS profile configuration

- OPTIONS request to existing API method

Http Request CORS Headers	TM Filter Execution Path of API Manager Transaction Pre the February 26 Release	TM Filter Execution Path of API Manager Transaction Post the February 26 Release	Changes

none

The filter execution path is **unavailable**

Source	Value
ApiShunt	optionsRequest
Response	200 OK Allow: GET, HEAD, OPTIONS Content-Type: text/plain
Error Trace	n/a
<code>circuit.failure.reason</code>	n/a
Transaction Audit Message	unavailable
Service Context	unavailable
Transaction Status	unavailable

FILTER EXECUTION PATH			
Filter	Status	Transaction Audit Message	Execution Time (ms)
API Broker			
API Transaction	✓	Options Request	5


Source	Value
ApiShunt	optionsRequest
Response	204 No Content Allow: GET, HEAD, OPTIONS
Error Trace	n/a
<code>circuit.failure.reason</code>	n/a
Transaction Audit Message	Options Request
Service Context	<pre>{"service": "My API test", "client": null, "method": "healthcheck", "status": "success", "monitor": true, "org": null, "app": null, "duration": 2}</pre>
Transaction Status	Pass

- The new API Broker meets OPTIONS request defined by [RFC 9110](#)

Origin:
<https://localhost:8065/>

The filter execution path is **unavailable**

Source	Value
ApiShunt	optionsRequest
Response	200 OK Allow: GET, HEAD, OPTIONS Content-Type: text/plain
Error Trace	n/a
circuit.failure.reason	n/a
Transaction Audit Message	unavailable
Service Context	unavailable
Transaction Status	unavailable

▼ FILTER EXECUTION PATH		
Filter	Status	Transaction Audit Mess
 API Broker	API Transaction ✓	Options Request

Source	Value
ApiShunt	optionsRequest
Response	200 OK Allow: GET, HEAD, OPTIONS Content-Length: 0
Error Trace	n/a
circuit.failure.reason	n/a
Transaction Audit Message	Options Request
Service Context	{"service":"My API test", "client":null, "method":"healthcheck", "status":"success", "monitor":true, "org":null, "app":null, "duration":2}
Transaction Status	Pass

- The new API Broker meets OPTIONS request defined by [RFC 7231](#)

API Gateway REST API with CORS profile configuration

- CORS profile configuration - Origin is *

- OPTIONS request to existing API method exhibit no change in behaviour

Http Request CORS Headers	TM Filter Execution Path of API Manager transaction Pre the February 26 Release	TM Filter Execution Path of API Manager transaction Post the February 26 Release	Changes																																
none	<p>The filter execution path is unavailable</p> <table border="1" data-bbox="422 423 1035 1138"> <thead> <tr> <th>Source</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>ApiShunt</td> <td>n/a</td> </tr> <tr> <td>Response</td> <td>204 No Content Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE</td> </tr> <tr> <td>Error Trace</td> <td>n/a</td> </tr> <tr> <td>circuit.failure.reason</td> <td>n/a</td> </tr> <tr> <td>Transaction Audit Message</td> <td>n/a</td> </tr> <tr> <td>Service Context</td> <td>unavailable</td> </tr> <tr> <td>Transaction Status</td> <td>unavailable</td> </tr> </tbody> </table>	Source	Value	ApiShunt	n/a	Response	204 No Content Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE	Error Trace	n/a	circuit.failure.reason	n/a	Transaction Audit Message	n/a	Service Context	unavailable	Transaction Status	unavailable	<p>The filter execution path is unavailable</p> <table border="1" data-bbox="1062 423 1686 1138"> <thead> <tr> <th>Source</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>ApiShunt</td> <td>n/a</td> </tr> <tr> <td>Response</td> <td>204 No Content Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE</td> </tr> <tr> <td>Error Trace</td> <td>n/a</td> </tr> <tr> <td>circuit.failure.reason</td> <td>n/a</td> </tr> <tr> <td>Transaction Audit Message</td> <td>n/a</td> </tr> <tr> <td>Service Context</td> <td>unavailable</td> </tr> <tr> <td>Transaction Status</td> <td>unavailable</td> </tr> </tbody> </table>	Source	Value	ApiShunt	n/a	Response	204 No Content Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE	Error Trace	n/a	circuit.failure.reason	n/a	Transaction Audit Message	n/a	Service Context	unavailable	Transaction Status	unavailable	<ul style="list-style-type: none"> • The API Broker meets OPTIONS request defined by RFC 9110 • API Gateway does not invoke message processing
Source	Value																																		
ApiShunt	n/a																																		
Response	204 No Content Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE																																		
Error Trace	n/a																																		
circuit.failure.reason	n/a																																		
Transaction Audit Message	n/a																																		
Service Context	unavailable																																		
Transaction Status	unavailable																																		
Source	Value																																		
ApiShunt	n/a																																		
Response	204 No Content Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE																																		
Error Trace	n/a																																		
circuit.failure.reason	n/a																																		
Transaction Audit Message	n/a																																		
Service Context	unavailable																																		
Transaction Status	unavailable																																		

<p>Origin: http://localhost:8080/</p>	<p>The filter execution path is unavailable</p> <table border="1"> <thead> <tr> <th>Source</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>ApiShunt</td> <td>n/a</td> </tr> <tr> <td>Response</td> <td>200 OK Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE Content-Length: 0</td> </tr> <tr> <td>Error Trace</td> <td>n/a</td> </tr> <tr> <td>circuit.failure.reason</td> <td>n/a</td> </tr> <tr> <td>Transaction Audit Message</td> <td>n/a</td> </tr> <tr> <td>Service Context</td> <td>unavailable</td> </tr> <tr> <td>Transaction Status</td> <td>unavailable</td> </tr> </tbody> </table>	Source	Value	ApiShunt	n/a	Response	200 OK Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE Content-Length: 0	Error Trace	n/a	circuit.failure.reason	n/a	Transaction Audit Message	n/a	Service Context	unavailable	Transaction Status	unavailable	<p>The filter execution path is unavailable</p> <table border="1"> <thead> <tr> <th>Source</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>ApiShunt</td> <td>n/a</td> </tr> <tr> <td>Response</td> <td>200 OK Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE Content-Length: 0</td> </tr> <tr> <td>Error Trace</td> <td>n/a</td> </tr> <tr> <td>circuit.failure.reason</td> <td>n/a</td> </tr> <tr> <td>Transaction Audit Message</td> <td>n/a</td> </tr> <tr> <td>Service Context</td> <td>unavailable</td> </tr> <tr> <td>Transaction Status</td> <td>unavailable</td> </tr> </tbody> </table>	Source	Value	ApiShunt	n/a	Response	200 OK Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE Content-Length: 0	Error Trace	n/a	circuit.failure.reason	n/a	Transaction Audit Message	n/a	Service Context	unavailable	Transaction Status	unavailable	<ul style="list-style-type: none"> • The API Broker meets OPTIONS request defined by RFC 7231 • API Gateway does not invoke message processing
Source	Value																																		
ApiShunt	n/a																																		
Response	200 OK Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE Content-Length: 0																																		
Error Trace	n/a																																		
circuit.failure.reason	n/a																																		
Transaction Audit Message	n/a																																		
Service Context	unavailable																																		
Transaction Status	unavailable																																		
Source	Value																																		
ApiShunt	n/a																																		
Response	200 OK Allow: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT, TRACE Content-Length: 0																																		
Error Trace	n/a																																		
circuit.failure.reason	n/a																																		
Transaction Audit Message	n/a																																		
Service Context	unavailable																																		
Transaction Status	unavailable																																		

**Axway can and will update this page with further examples upon client feedback*