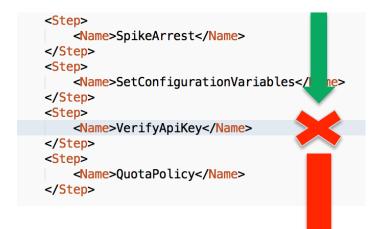


Foundational Training

Fault Handling

Apigee Fault Handling



Faults in Apigee are similar to exceptions in programming languages. When a fault is raised, current policy processing is aborted and switches to error processing called <FaultRules>This is supported in ProxyEndpoint and TargetEndpoint configurations only.

```
<DefaultFaultRule>
         <Step>
             <Name>SyslogPolicy</Name>
         </Step>
         <AlwaysEnforce>true</AlwaysEnforce>
    </DefaultFaultRule>
    <FaultRules>
         <FaultRule name ="Fault.InvalidKey">
             <Step>
10
                 <Name>Add-WWW-Authenticate-Header</Name>
11
             </Step>
         <Condition>(fault.name =="invalid_consumer_key")</Condition>
12
13
         </FaultRule>
    </FaultRules>
14
    <ht><htTTPProxyConnection></h>
             <BasePath>/certification/v1/weather/BasePath>
16
17
             <VirtualHost>default</virtualHost>
             <VirtualHost>secure</VirtualHost>
18
    </HTTPProxyConnection>
```

Apigee Fault Handling (cont'd)

<FaultRule>

- Support for multiple fault rules, executed conditionally
- Raised manually or automatically upon policy failure
- If multiple conditional FaultRules defined, their conditions are evaluated in reverse order (bottom up).

```
<DefaultFaultRule>
            <Name>SyslogPolicy</Name>
        <AlwaysEnforce>true</AlwaysEnforce>
    </DefaultFaultRule>
    <FaultRules>
        <FaultRule name ="Fault.InvalidKey">
                <Name>Add-WWW-Authenticate-Header</Name>
        <Condition>(fault.name =="invalid consumer key")</Condition>
        </FaultRule>
    </FaultRules>
15 <HTTPProxvConnection>
             <BasePath>/certification/v1/weather</BasePath>
17
            <VirtualHost>default</VirtualHost>
            <VirtualHost>secure</VirtualHost>
19 </HTTPProxyConnection>
```

<DefaultFaultRule>

- A catch-all / post processing fault rule is available using the flow
- If no FaultRule is matched, this flow will execute.
- Use <AlwaysEnforce>true</AlwaysEnforce> to use this flow for post processing in error flows. This will ensure the flow executes after a matched fault rule has completed its processing.

3 scenarios where processing will switch to fault processing:

- Using RaiseFault policy
- Any policy failure when contineOnError=false (default setting)
- Non-success response received from service callout or backend request (4XX, 5XX status codes).

Rewriting Backend Error Responses

Apigee Edge has a pre-defined fault response format:

You can either:

- 1. Rewrite backend error responses to match Apigee's format
- Rewrite Apigee fault responses to match the backend error response format

Day 3 – Lab 11 Fault Rules

Create a fault rule in Proxy Endpoint

```
<FaultRules>
    <FaultRule name="SpikeArrestViolation">
         <Condition>fault.name == "SpikeArrestViolation"</Condition>
         <Step><Name>Spike-Arrest-Violation</Name></Step>
         </FaultRule>
</FaultRules>
```

Create a Assign Message Policy named "Spike-Arrest-Violation"

Test your proxy using Postman

Thank you

