

Raise Fault Policy

Generates a custom message in response to an error condition. Use Raise Fault to define a fault response that is returned to the requesting app when a specific condition arises.

In the most common usage, Raise Fault is used to return a custom fault response to the requesting app. For example to return a 404.

```
<RaiseFault name="404">
  <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
  <FaultResponse>
    <Set>
      <StatusCode>404</StatusCode>
      <ReasonPhrase>The resource requested was not found</ReasonPhrase>
    </Set>
  </FaultResponse>
</RaiseFault>
```

Apigee Edge enables you to perform custom exception handling using a policy of type Raise Fault. The Raise Fault policy, which is a variation of the [Assign Message policy](#), lets you generate a custom fault response in response to an error condition.

Use the Raise Fault policy to define a fault response that is returned to the requesting app when a specific error condition arises. The fault response can consist of HTTP headers, query parameters, and a message payload. A custom fault response can be more useful to app developers and app end users than generic error messages or HTTP response codes.

When executed, the Raise Fault policy transfers control from the current flow to the Error flow, which then returns the designated fault response to the requesting client app. When the message Flow switches to the Error flow, no further policy processing occurs. All remaining processing Steps are bypassed, and the fault response is returned directly to the requesting app.

There are two main places where you use the Raise Fault policy:

- In a ProxyEndpoint/TargetEndpoint flow to trigger a fault in response to a condition or set of conditions. That fault can trigger a specific fault rule or, if there are no fault rules defined, it terminates processing of the proxy.

- In a fault rule if you want to detect an error during processing of a fault. For example, your fault handler itself could cause an error that you want to signal by using Raise Fault.

Step-1

First you have to add the Raise Fault Policy into your Application.

The screenshot shows the Apigee API Proxy console interface. The breadcrumb navigation at the top indicates the path: Dashboard / API Proxies / Raise_Fault_Policy. The main heading is 'Raise_Fault_Policy'. Below this, there are tabs for Project, Save, Revision 1, New, New Policy, Attach Policy, Tools, and Deployment. A 'Help for Selected' button is also present, with a tooltip for 'Raise Fault Policy'. On the left, a 'Navigator' pane shows a tree view of the application's structure, including Policies, Proxy Endpoints, and Scripts. The 'Policies' section is expanded, showing 'Raise Fault 1' and 'Raise Fault 2'. The 'New Policy' dropdown menu is open, displaying a list of policy categories: Access Control, Generate SAML Assertion, Validate SAML Assertion, MEDIATION (JSON to XML, XML to JSON, Raise Fault, XSL Transform, SOAP Message Validation), Assign Message, Extract Variables, Access Entity, Key Value Map Operations, EXTENSION (JavaScript, Service Callout, Statistics Collector). The 'Raise Fault' option under the MEDIATION category is highlighted with a red rectangle. The main workspace shows a diagram of the policy flow, with a 'Request' box pointing to a 'Server' box, and a 'Response' box. Below the diagram, there is a code editor showing the XML configuration for the 'Raise-Fault-1' policy:

```
<code>coding="UTF-8" standalone="yes">>
se" continueOnError="false" enabled="true" name="Raise-Fault-1">
Fault 1</DisplayName>
```

The screenshot shows the 'New Policy: Raise Fault' dialog box. It has a title bar with a close button. The dialog contains the following fields and controls:

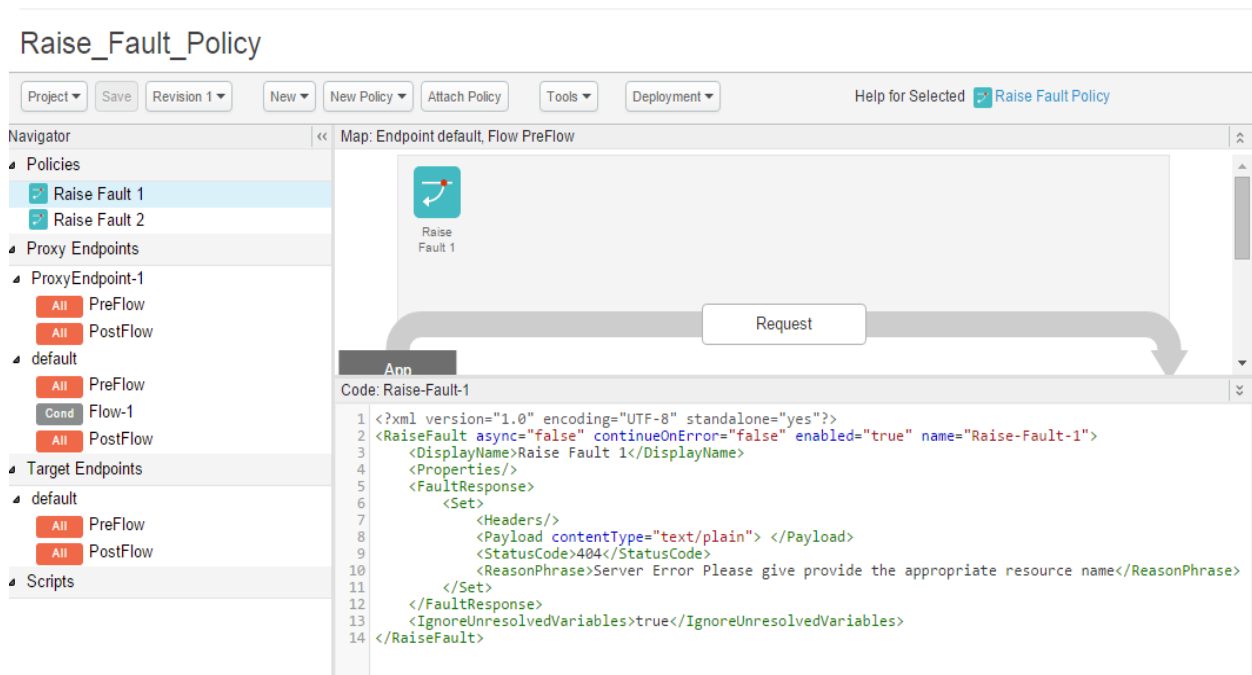
- Policy Display Name:** A text input field containing 'Raise Fault 3'.
- Policy Name:** A text input field containing 'Raise-Fault-3'.
- Attach Policy:** A checkbox that is checked.
- Flow:** A dropdown menu showing 'Flow PreFlow, Proxy Endpoint default'.
- Segment:** Two radio buttons, 'Request' (selected) and 'Response'.

At the bottom right, there are two buttons: 'Cancel' and 'Add'.

- Fill the fields as per your desired requirement.
- Click on “add” button.

Step-2

After adding Raise Fault Policy the Policy looks like this-



Code-

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<RaiseFault async="false" continueOnError="false" enabled="true" name="Raise-Fault-2">
```

```
  <DisplayName>Raise Fault 2</DisplayName>
```

```
  <Properties/>
```

```
  <FaultResponse>
```

<Set>

<Headers/>

<Payload contentType="text/plain"> </Payload>

<StatusCode>500</StatusCode>

<ReasonPhrase>Server Error</ReasonPhrase>

</Set>

</FaultResponse>

<IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>

</RaiseFault>

Step-3

- Whenever you will pass some wrong parameter (or) any server error ,or may be when you are passing resources from the browser there it may be chance of error and it will show you the error.

https://enterprise.apigee.com/platform/#/karanmeheta1/apis/Raise_Fault_Policy/1?section=trace

AppsBookmarksLive Cricket Streami...DD national hindi ch...Gmail - InboxAptitude Questions...Aptitude Questions ...Latest Tutorials for L...By Mr Nataraj Sir | J...Other bookmarks

apigeeDashboardAPIsPublishAnalyticsAdminHelpsamirkantla@rediffmail...API Management

Dashboard / API Proxies / Raise_Fault_PolicyOrganization karanmeheta1

Raise_Fault_PolicyACCESS THE NEW VERSION OF PROXY EDITOROVERVIEWDEVELOPTRACE

Deployment to TraceEnvironment test, Revision 1Stop Trace SessionRemaining Time 07:10Download Trace SessionNode.js Logs

Filters

Transactions

	Status	Method	URI	Elapsed
7	404	GET	/v1/raise_fault...	8 ms
6	404	GET	/v1/raise_fault...	5 ms
5	200	GET	/	165 ms
4	404	GET	/v1/raise_fault...	5 ms
3	404	GET	/v1/raise_fault...	9 ms
2	404	GET	/v1/raise_fault...	6 ms
1	404	GET	/v1/raise_fault...	8 ms

View Options

Transaction Map

- Show Disabled Policies (none)
- Show Skipped Phases (3)
- Show All FlowInfos (3)

Phase Details

- Automatically Compare Selected Phase
- Show Variables
- Show Properties

Send Requests

Method URL

GEThttp://karanmeheta1-test.apigee.net/v1/raise_fault_policy/appSend404

Or Send with the API Console

Transaction Map

1ms

BackNext

Phase Details

Raise Fault 1

404 Server Error Please give provide the appropriate resource name

Variables Read and Assigned

raisefault.failed = true

Output from all Transactions