Google Cloud

Edge Fundamentals

Conditions and Route Rules

Conditions

- Conditions enable API proxies to behave dynamically at runtime.
- Conditions define operations on variables, which are evaluated by the Apigee Edge processing pipeline.
- The basic structure of a conditional statement is

```
<Condition>{variable.name}{operator}{"value"}</Condition>
```

- Conditional statements are boolean and always evaluate to true or false
- Flow variables and custom variables can be used for evaluating conditions and executing appropriately.
- Conditions can be chained

Conditions

Conditions can be used to control the following behaviour:

- Policy execution - to control the enforcement of the policies

```
<Step>
  <Condition>request.header.accept = "application/json"</Condition>
  <Name>XMLToJSON</Name>
</Step>
```

- Flow execution - to control the execution of named flows in ProxyEndpoints and TargetEndpoints

```
<Flow name="GetRequests">
  <Condition>request.verb = "GET"</Condition>
  <Request>
    <Step>
      <Condition>request.path MatchesPath "/statuses/**"</Condition>
      <Name>StatusesRequestPolicy</Name>
    </Step>
  </Request>
  <Response/>
</Flow>
```

Conditions

- Target end point route selection - control the target endpoint invoked by proxy endpoint configuration. control the target endpoint invoked by proxy endpoint configuration

```
<RouteRule name="xmlTarget">
  <Condition>request.header.Content-Type = "text/xml"</Condition>
  <TargetEndpoint>XmlTargetEndpoint</TargetEndpoint>
</RouteRule>
<RouteRule name="default">
   <TargetEndpoint>target</TargetEndpoint>
</RouteRule>
```

- URI path matching can be done using “*” for single path element and “**” for multiple URI levels. “%” is treated as the escape character

- *null*, *true*, and *false* are the literals available in conditions

Pattern matching in Conditions

The different types of operators with patterns are:

- Matches - gives you two possibilities either match the string literally, or do a wildcard match with "*"

```
<Condition>(proxy.pathsuffix Matches "/cat")</Condition>
<Condition>(proxy.pathsuffix Matches "/*at")</Condition>
<Condition>(proxy.pathsuffix Matches "/cat*")</Condition>
<Condition>(proxy.pathsuffix Matches "/c%*at")</Condition>
```

- JavaRegex - allows regular expression pattern matching, and Edge follows the same rules as the classes in the java.util.regex package in the Java language

```
<Condition>(proxy.pathsuffix JavaRegex "/cat")</Condition>
<Condition>(proxy.pathsuffix JavaRegex "/c*t")</Condition>
<Condition>(proxy.pathsuffix JavaRegex "/ca?t")</Condition>
<Condition>(proxy.pathsuffix JavaRegex "/[cbr]at)</Condition>
```

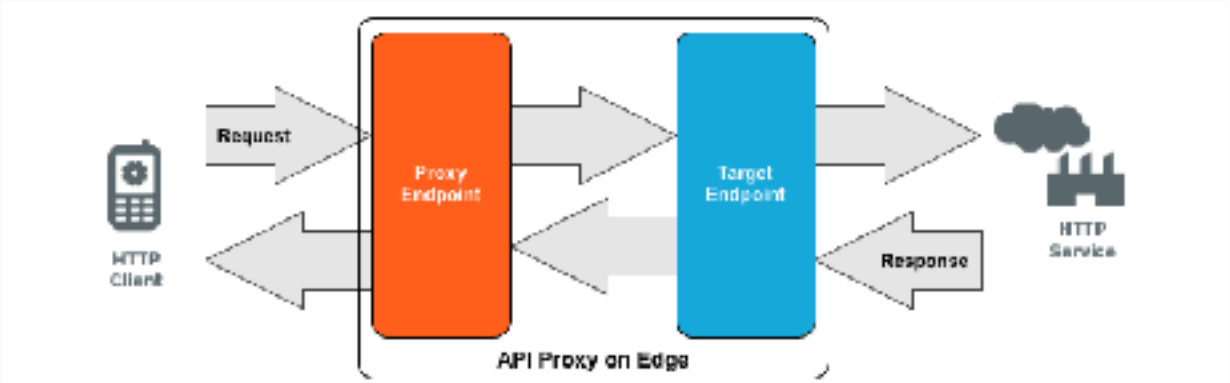
- MatchesPath - lets you use two wildcard notations: a single asterisk (*) and a double asterisk (**). The single asterisk matches one path element. The double asterisk matches one or many path elements

```
<Condition>(proxy.pathsuffix MatchesPath "/animals/*)</Condition>
<Condition>(proxy.pathsuffix MatchesPath "/animals/**)</Condition>
<Condition>(proxy.pathsuffix MatchesPath    "/animals/*/wild/**)</
Condition>
```

Understanding Routes

A route determines the path of a request from the Proxy Endpoint to the Target Endpoint.

Included in the route is the URL used to access the API Proxy Endpoint and the URL of the backend service defined by the Target Endpoint.

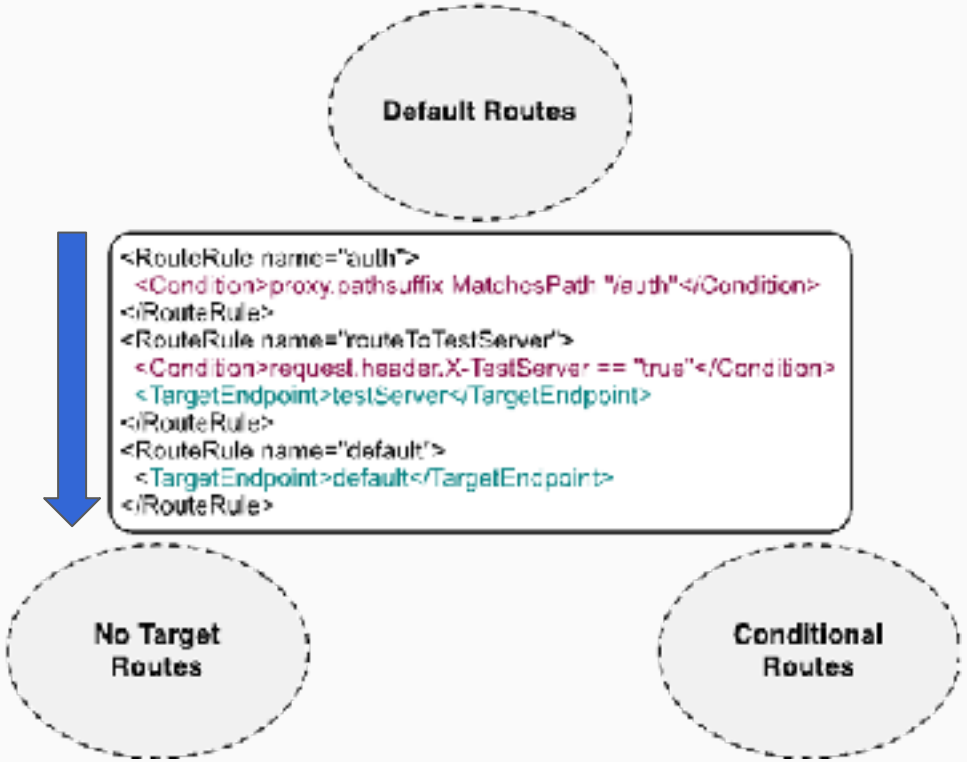



```
<ProxyEndpoint name="default">
    <PreFlow/>
    <Flows>
      <Flow name="forecast">
        <Condition>
          proxy.pathsuffix MatchesPath "/forecastrss"
        </Condition>
      </Flow>
    </Flows>
    <PostFlow/>
    <HTTPProxyConnection>
        <VirtualHost>default</VirtualHost>
        <BasePath>/v1/weather</BasePath>
    </HTTPProxyConnection>
    <RouteRule name="default">
        <TargetEndpoint>default</TargetEndpoint>
    </RouteRule>
</ProxyEndpoint>
```

```
<TargetEndpoint name="default">
    <PreFlow/>
    <Flows/>
    <PostFlow/>
    <HTTPTargetConnection>
        <URL>http://weather.yahooapis.com</URL>
    </HTTPTargetConnection>
</TargetEndpoint>
```

Route Rules

- The <RouteRule> tag in a Proxy Endpoint definition determines the target of the API proxy.
- Target can be defined as:
  - Direct URL
  - Single target definition
  - Null route
  - Multiple targets based on condition
- For multiple targets, the order is important, the first condition match gets executed



THANK YOU