Google Cloud

Edge Fundamentals

Policy Overview

Proprietary + Confidential

What is a policy ?

- A policy is like a module that implements a specific, limited management function.
- Edge enables you to 'program' API behavior without writing any code, by using 'policies'.
- Policies are designed to let you add common types of management capabilities to an API easily and reliably.
- Policies provide features like security, rate-limiting, transformation, and mediation capabilities - saving you from having to code and maintain this functionality on your own.
- Edge's out-of-the-box policies enable you to augment your API with sophisticated features to control traffic, enhance performance, enforce security, and increase the utility of your APIs, without requiring you to write any code or to modify any backend services.
- Extension policies enable you to implement custom logic in the form of JavaScript, Python, Java, and XSLT.

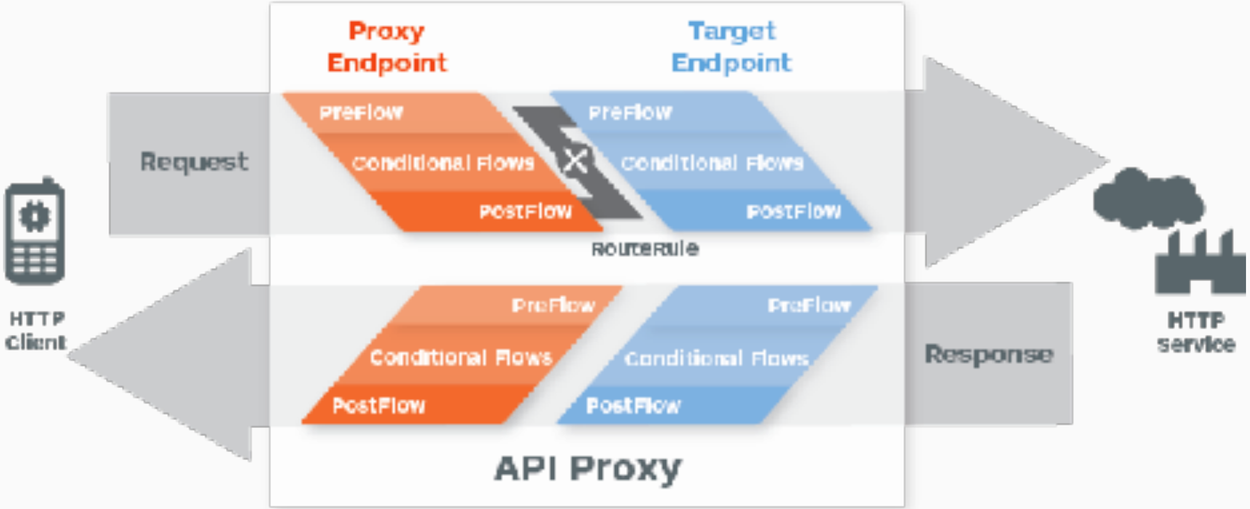
Out of the box policies

Traffic management policies	Mediation policies	Security policies	Extension policies
● Cache policies ● Concurrent Rate Limit policy ● Quota policy ● Reset Quota policy ● Spike Arrest policy	● Access Entity policy ● Assign Message policy ● Extract Variables policy ● JSON to XML policy ● Key Value Map Operations policy ● Raise Fault policy ● SOAP Message Validation policy ● XML to JSON policy ● XSL Transform policy	● Access Control policy ● Basic Authentication policy ● JSON Threat Protection policy ● LDAP policy *† ● OAuth v2.0 policies ● OAuth v1.0a policy ● Regular Expression Protection policy ● SAML Assertion policies ● Verify API Key policy ● XML Threat Protection policy	● Java Callout policy * ● JavaScript policy ● Message Logging policy ● Python Script policy * ● Service Callout policy ● Statistics Collector policy

* Cloud Enterprise only

† On-Premises installation only

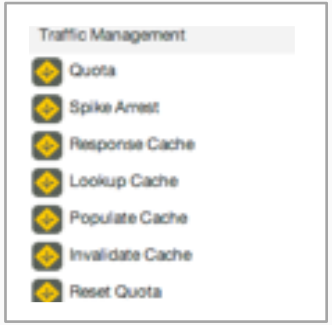
Enable fine grain controls in the API proxy



Many different points where policies can be configured to execute

Build APIs faster to help..

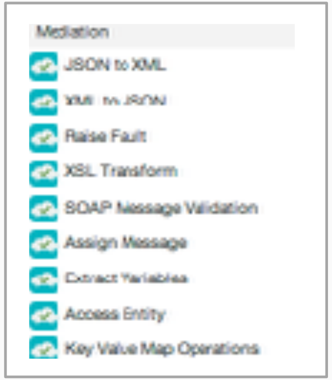
Manage interactions with API consumers and optimize performance



Traffic Management

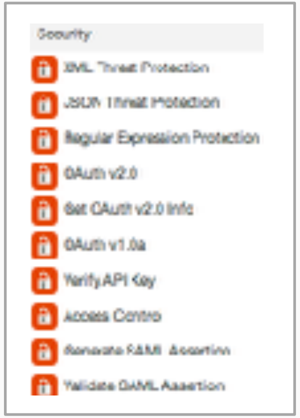
- Quota
- Spike Arrest
- Response Cache
- Lookup Cache
- Populate Cache
- Invalidate Cache
- Reset Quota

Transform, translate and reformat data for easy consumption



Mediation

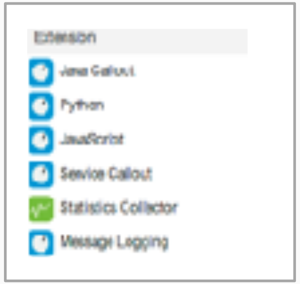
- JSON to XML
- XML to JSON
- Raise Fault
- XSL Transform
- SOAP Message Validation
- Assign Message
- Extract Variables
- Access Entity
- Key Value Map Operations



Security

- XML Threat Protection
- JSON Threat Protection
- Regular Expression Protection
- OAuth v2.0
- Get OAuth v2.0 Info
- OAuth v1.0a
- Verify API Key
- Access Control
- Generate SAML Assertion
- Validate SAML Assertion

Secure APIs and protect back-end systems from attack



Extension

- Java Callout
- Python
- JavaScript
- Service Callout
- Statistics Collector
- Message Logging

Extend with programming when you need it

Traffic Management

Spike Arrest

Spike Arrest

- help protect your API proxy’s target backend against severe traffic spikes and denial of service attacks
- used to control requests by the second and minute
- typically used in the “preflow”
- http://docs.apigee.com/api-services/reference/spike-arrest-policy

Quota

- Used to limit number of requests over a period of time such as a minute, hour, day, week, or month.
- Could be the same limit for everyone, or configure based on product, developer, etc
- Typically used in the “preflow”
  - http://docs.apigee.com/api-services/reference/quota-policy
- “Reset Quota” is used to reset the quota during the request
  - http://docs.apigee.com/api-services/reference/reset-quota-policy

Concurrent Rate Limit

Concurrent Rate Limit

- Throttles inbound connections from Edge to your backend services
- Used to limit the number of concurrent connections
- Not typically used but is available

- Needs to be attached to the both the request and response flows in the Target endpoint

- http://docs.apigee.com/api-services/reference/concurrent-rate-limit-policy

Comparison of Spike Arrest, Quota and Concurrent Rate Limit

- http://docs.apigee.com/api-services/content/comparing-quota-spike-arrest-and-concurrent-rate-limit-policies

Response Cache

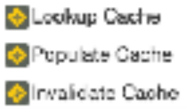
Response Cache

- Used to cache the whole HTTP response (including body, headers, status code, etc)
- Response Cache policy can improve performance by retrieving response from the cache instead of back end

- Policy attached in both the request and response flows
- Typically only used with GET calls

- http://docs.apigee.com/api-services/reference/response-cache-policy

Cache



Lookup Cache

Populate Cache

Invalidate Cache

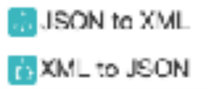
- Used to cache specific pieces of data within the proxy
- Examples of using cache:
  - tokens, service callout responses, data from previous calls, etc

- Use the “populate”, “lookup” and “invalidate” policies to control the cache

- http://docs.apigee.com/api-services/reference/populate-cache-policy
- http://docs.apigee.com/api-services/reference/lookup-cache-policy
- http://docs.apigee.com/api-services/reference/invalidate-cache-policy
- http://docs.apigee.com/api-services/reference/working-cachekeys

Mediation

JSON to XML, XML to JSON



JSON to XML

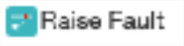
XML to JSON

- Used to convert a JSON payload to XML or an XML payload to JSON
- Convenient if the backend is XML and a JSON REST API is needed
- There are several “out of the box” types of conversion available

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/json-xml-policy
- http://docs.apigee.com/api-services/reference/xml-json-policy

Raise Fault

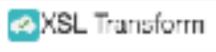

Raise Fault

- Raises a “fault” in a proxy
- A technique to force the execution of FaultRules
- Could also be used to send an error response back to the client

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/raise-fault-policy
- http://docs.apigee.com/api-services/content/fault-handling

XSL Transform



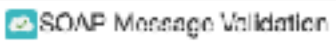
XSL Transform

- Applies a custom XSLT to an XML payload
- This allows the transformation to another format such as XML, HTML, etc

- Policy can only be used if the “Content-Type” is XML
- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/xsl-transform-policy

SOAP Message Validation


SOAP Message Validation

- Used to validate an XML message against an XSD schema
- Validate SOAP messages against a WSDL
- Confirm message is well-formed

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/message-validation-policy

Assign Message

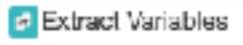
Assign Message

- Creates or modifies HTTP request or response messages
- Used to modify the request message before it is sent to the back end
- Used to modify the response message before it is sent to the client
- Used to create/modify custom request/response objects for a Service Callout policy
- Used to create and modify flow variables

- Policy can be used at any point in the proxy
- One of the most used policies

- http://docs.apigee.com/api-services/reference/assign-message-policy

Extract Variables


Extract Variables

- Used to extract data from a variable
- Often, that variable is the HTTP Request or Response
- Headers, URI paths, form/query parameters, JSON/XML payloads can be extracted using this policy
- Use a text pattern to extract the data

- Policy can be used at any point in the proxy
- One of the most used policies

- http://docs.apigee.com/api-services/reference/extract-variables-policy

Access Entity

Access Entity

- Retrieves entity profiles configured in Edge
- Use this policy to retrieve metadata from:
  - App
  - API Product
  - Company
  - Company developer
  - Consumer Key
  - Developer
- Policy can be used at any point in the proxy
- http://docs.apigee.com/api-services/reference/access-entity-policy

Key Value Map (KVM)

Access Entity

- Provides access to a Key Value Map store in Edge
- Supports PUT, GET, DELETE operations
- The creation of the KVM is done in a different step
  - Management UI (APIs → Environment Configuration)
  - Management API

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/key-value-map-operations-policy

Security

Basic Authentication

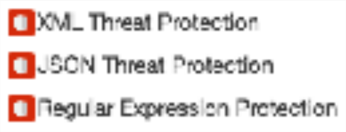
Basic Authentication

- Provides encoding and decoding to use for Basic Authentication
- Will base64 encode a username and password for backend authentication
- Will base64 decode the Authorization header to retrieve username and password for validation

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/basic-authentication-policy

XML, JSON, Regex Threat Protection



XML Threat Protection

JSON Threat Protection

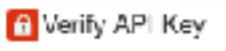
Regular Expression Protection

- These policies provide an easy way to inspect the payload coming in (from a POST) or from a backend response from threats
- The XML and JSON policies will validate the message structure meets predefined standards
- The Regex threat protection will evaluate the payload against predefined regular expressions (ex: SQL injection)

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/xml-threat-protection-policy
- http://docs.apigee.com/api-services/reference/json-threat-protection-policy
- http://docs.apigee.com/api-services/reference/regular-expression-protection

Verify API Key



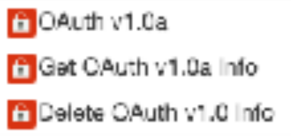
Verify API Key

- This policy will enforce verification of the API key at runtime
- The purpose is to restrict access to your API to only those with a valid API Key
- The key is created when a Developer App is created with a specific API Product
- The API Key can be passed in through the header or as a query parameter

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/verify-api-key-policy

OAuth v1.0a



- These policies are used to support OAuth v1.0a
- Actions supported:
    - Generating a request token and an access token, associate token verification code with a request token, verifying an access token, customize an access token

- Policy can be used at any point in the proxy
- These policies are rarely used because OAuth v1.0a has been replaced by OAuth v2.0

- http://docs.apigee.com/api-services/reference/oauth-10-policy
- http://docs.apigee.com/content/retrieve-token-attributes-using-getoauthv1info
- http://docs.apigee.com/api-services/content/delete-oauth-v1-info

OAuth v2.0

OAuth v2.0

Get OAuth v2.0 Info

Set OAuth v2.0 Info

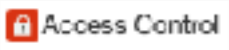
Delete OAuth v2.0 Info

- These policies are used to support OAuth v2.0a
- Actions supported:
  - Generate Access Token, Verify Access Token, Generate Authorization Code, Refresh Access Token
  - Retrieve additional metadata related to the access token or authorization code

- Policy can be used at any point in the proxy
- These policies are rarely used because OAuth v1.0a has been replaced by OAuth v2.0

- http://docs.apigee.com/api-services/content/oauthv2-policy
- http://docs.apigee.com/api-services/reference/get-oauth-v2-info-policy
- http://docs.apigee.com/api-services/reference/set-oauth-v2-info-policy
- http://docs.apigee.com/api-services/content/delete-oauth-v2-info
- http://docs.apigee.com/api-services/content/oauth-home

Access Control Policy



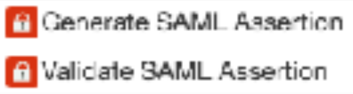
Access Control

- This policy empowers the developer to allow or deny access based on IP addresses
- The following configurations are supported:
  - Allow or Deny Specific IP Address
  - Allow or Deny Multiple IP Address
  - Allow or Deny IP Addresses using basic matching rules (ranges not supported)

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/access-control-policy

SAML



Generate SAML Assertion

Validate SAML Assertion

- This policy allows the support of SAML in the proxy
- The validate policy is used to validate a SAML assertion that is attached to an inbound XML message
- The generate policy is used to generate and attach a SAML assertion to an outbound XML message

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/saml-assertion-policy

Extension

Javascript Callout

- This policy allows custom Javascript code to execute within the context of an API proxy flow
- The “context” object provides runtime access to the current context of the flow
- There is a built-in Javascript object model available

- Policy can be used at any point in the proxy
- This is a commonly used policy

- http://docs.apigee.com/api-services/reference/javascript-policy
- http://docs.apigee.com/api-services/reference/javascript-object-model

Python Callout

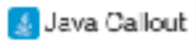
Python

- This policy allows custom Python code to execute within the context of an API proxy flow
- The “flow” variable provides runtime access to the current context of the flow
- Best used for existing custom Python code

- Policy can be used at any point in the proxy
- It is recommended to use a Javascript policy where possible

- http://docs.apigee.com/api-services/reference/python-script-policy

Java Callout



Java Callout

- This policy allows custom Java code to execute within the context of an API proxy flow
- The java code must be compiled and packaged as a jar file and then uploaded into the proxy
- It must follow the guidelines as specified in the documentation

- Policy can be used at any point in the proxy
- This is only available on a paid org or on-prem

- http://docs.apigee.com/api-services/reference/java-callout-policy

Service Callout

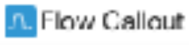
Service Callout

- This policy allows another service to be called from the API proxy flow
- Calls can be made to an external or internal service
- Using the “AssignMessage” policy, a new request message can be created for this policy
- Using the “ExtractVariables” policy, the response can be parsed

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/service-callout-policy

Flow Callout



Flow Callout

- This policy calls out to a “shared flow” from a proxy or another shared flow
- The “shared flow” must first be created and deployed
- Shared flows can call other shared flows

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/flow-callout-policy
- http://docs.apigee.com/api-services/content/shared-flows

Statistics Collector

Statistics Collector

- This policy enables the developer to collect statistics based on data in the message.
- The data can come from flow variables or custom variables that have been defined by previous policies.
- The data is processed by the analytics service and will appear in the analytics reports.
- When creating an analytic report, the custom stats data will be available to select.

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/statistics-collector-policy

Message Logging

Message Logging

- This policy allows custom logging features
- For cloud, the policy will gather data and send to a logging service asynchronously
- For onprem, the policy can be configured to log custom messages to the local file system

- Policy can be used at any point in the proxy

- http://docs.apigee.com/api-services/reference/message-logging-policy

THANK YOU