

APPSENTINELS API SECURITY

APIGEE DEPLOYMENT

Revision History

Version	Date	Revision Description
1.0	22/08/2022	The first version of the guide
1.1	14/19/2022	Service callout only based post client implementation
1.2	25/10/2022	Review comment incorporation

Scope

This guide contains approach and steps to deploy APIGee with Appsentinels controller.

AppSentinels.ai Doc Center

The content in this guide is an excerpt from the topics in the [AppSentinels.ai Doc Center](#). Visit the Doc Center for full context and more topics relevant to the AppSentinels API Security solution.

Table of Contents

Contents

Revision History	i
Scope.....	i
AppSentinels.ai Doc Center.....	i
<i>Table of Contents</i>	<i>0</i>
<i>Appsentinels logging requirements</i>	<i>1</i>
Approach	1
Shared Flows:	2
Steps	2
Creating KeyStore and TrustStore.....	2
Creating Shared Flows.....	2
Attach Shared Flows.....	3

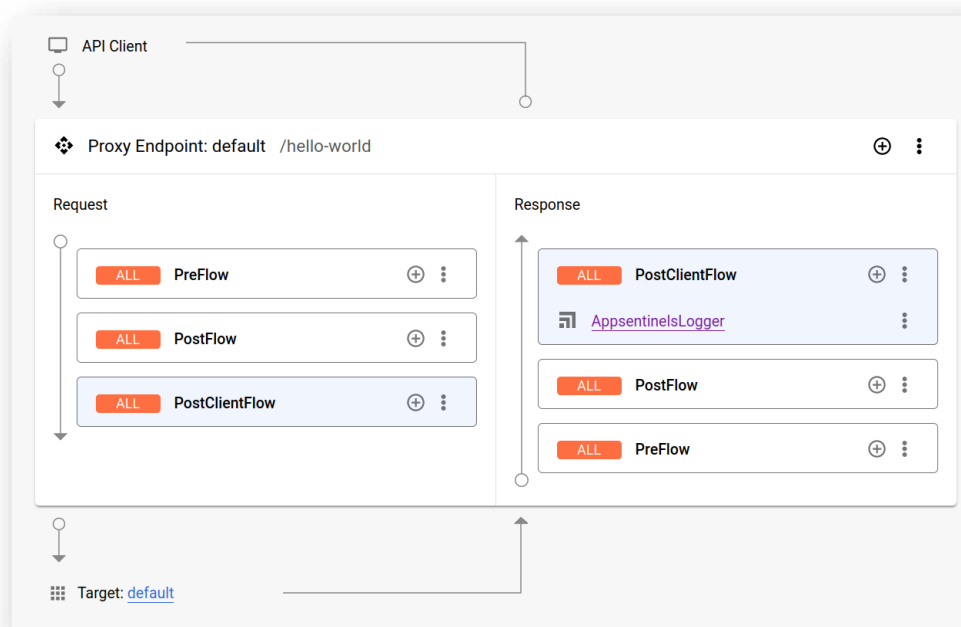
Appsentinels logging requirements

Appsentinels will employ a controller on your premise to collect below information from APIGee,

1. Headers (name + values)
2. Query (name + values)
3. Path
4. Payloads of request and response

Approach

In PostClientFlow, service callouts for request and response will independently pass on the headers along with other information (path, query, payloads) to Appsentinels controller endpoint. We will be using 2 Service Callouts since we cannot create one single request containing request and response information.



Shared Flows:

1. Shared flows allow for easier integration into flows belonging to multiple API proxies
2. Service Callouts will be implemented as part of a shared flow which can be attached to any proxy's post client flow

Steps

Creating KeyStore and TrustStore (in case of mTLS)

1. Create truststore **AppsentinelLoggingTrustStore** (only needed for self-signed certs in case server, aka Appsentinel controller is running a self signed certificate)
2. Create keystore **AppsentinelLoggingKeyStore** with alias **LoggingKeyStore** (<https://docs.apigee.com/api-platform/system-administration/creating-keystores-and-truststore-cloud-using-edge-ui>). Load client's private key and public cert into AppsentinelLoggingKeyStore
3. These will be used for mTLS during Service Callouts

Creating Shared Flows

1. There will be a shared flow bundles called AppsentinelLoggingSharedFlow available (provided separately). This performs mTLS based request and response log sending to Appsentinel controller. This will be inserted into response PostClientFlow.
2. Now go to Apigee dashboard and Shared Flows. Next click on Upload Bundle
3. Select the AppsentinelLoggingSharedFlow_*.zip bundle

Shared Flows			CREATE NEW	UPLOAD BUNDLE
Filter Enter property name or value				
Name ↑	Environments	Last Modified		
AppsentinelLoggingSharedFlow	eval	51 minutes ago		

4. Configure the endpoint for sending logs. Go to Service Callout-Request policy and configure the URL under HTTPTargetConnection
5. Go to Service Callout-Response and configure the URL under HTTPTargetConnection

The below example configures URL to where Appsentinel's controller will be reachable. This will be deployment specific.

```
<HTTPTargetConnection>
  <Properties/>
  <URL>https://<Controller-URL-here>:9006</URL>
  <SSLInfo>
    <Enabled>true</Enabled>
    <ClientAuthEnabled>true</ClientAuthEnabled>
    <KeyStore>Appsentinel'sLoggingKeyStore</KeyStore>
    <KeyAlias>LoggingKeyStore</KeyAlias>
    <IgnoreValidationErrors>false</IgnoreValidationErrors>
    <Ciphers/>
    <Protocols/>
  </SSLInfo>
</HTTPTargetConnection>
</ServiceCallout>
```

6. Now deploy the shared flows

Attach Shared Flows

1. Go to your API proxy. We will be creating Flow Callouts and attach the above shared flows under Proxy Endpoints
2. In Response PostClientFlow, create a new policy of type Flow Callout. Rename it as Appsentinel'sLogger. Under SharedFlow choose "Appsentinel'sLoggingSharedFlow". Add this policy unconditionally

Create policy



Apigee's out-of-the-box policies augment your APIs to control traffic, enhance performance, enforce security, and increase the utility of your APIs. See the [Policy reference overview](#) for a list of all policies and how to use them.

Select policy type *
Flow Callout

Details

Name *
AppsentinelsLogger

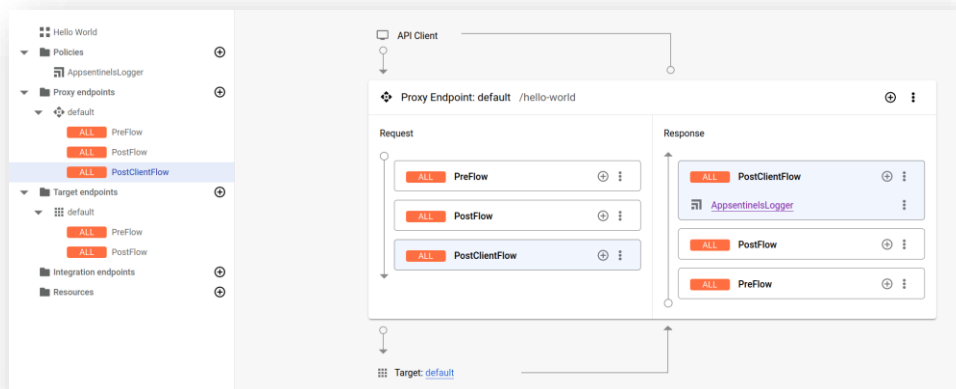
❗ Policy name must be unique

Display name *
AppsentinelsLogger

Used in addition to the name attribute to label the policy in the Proxy Editor with a different, natural-language name.

Sharedflow *
AppsentinelsLoggingSharedFlow

3. The proxy flow should end up looking as below,



4. Deploy your API proxy with the changes