Service Callouts

Google Cloud

Placeholder for Example Illustration

apigee

Service Callout

- The Service Callout policy lets you call to another service from your API proxy flow.
- The policy supports requests over HTTP and HTTPS. Also has Request and Response elements which is separate from the actual proxy request and response
- A callout is typically used with two other policies: Assign Message (for the service callout request) and Extract Variables (to extract the service callout response)

apigee

Service Callout Targets

Target connection configurations supported in Service Callout

- External use case - call 3rd party APIs
  - HTTPTargetConnection
    - URL
    - Target Server (Load Balancer)
    - Target Server with SSLInfo
- Internal use case - call a proxy in the same organization and environment
  - LocalTargetConnection
    - APIProxy (proxy in the same organization and environment)
    - Proxy endpoint (proxy endpoint that should be the target of calls)
    - Path (path to the endpoint that is being targeted)

Mashing up the Responses

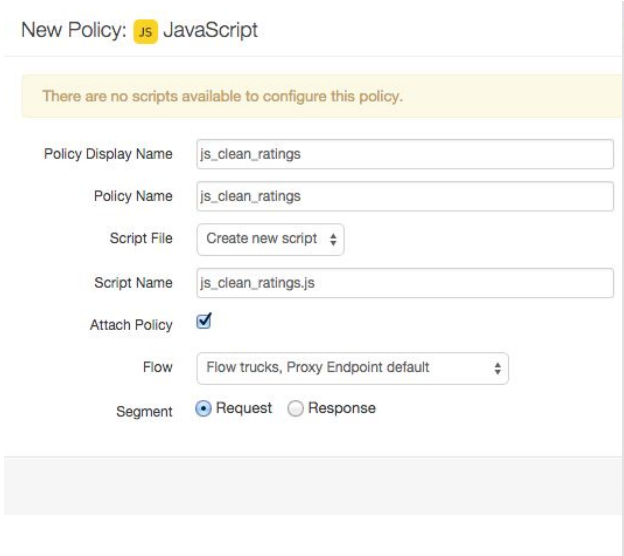
- Strategies for Mashups
  - Assign Message (simple, most constrictive)
  - Extract Variables, Assign Message (simple, constrictive)
  - JavaScript (flexible, complex)
  - Java (flexible, complex)
  - XSLT (flexible, complex)

  - Choose a format that works for you!!

apigee

A Mashed Up Response Sure but…..

- There is a lot of extra metadata we don't care about in the ratings response
- One method of cleansing your JSON of unwanted fields is to use the JavaScript policy, or you could extract out the fields you want with an Extract Variables policy, or a combination of both.



Code: js_clean_ratings.js

```
try{
  var servicecalloutcontent = context.getVariable('ratingsResponse.content');
  var ratingsSourceObj = JSON.parse(servicecalloutcontent);
  var cleanRatings = {};
  cleanRatings = ratingsSourceObj.entities;
  for(var i =0;i< cleanRatings.length; i++){
    delete(cleanRatings[i].uuid);
    delete(cleanRatings[i].type);
    delete(cleanRatings[i].created);
    delete(cleanRatings[i].modified);
    delete(cleanRatings[i].metadata);
    delete(cleanRatings[i].truck);
  }
  context.setVariable('ratingsResponse.content', JSON.stringify(cleanRatings));
}catch(exception){
  context.setVariable('js_error', exception.message);
}
```

Thank You

Google Cloud