Passing data to and from the backend service

When an API client make a request to your API deployed on API Gateway, the client can pass any or all of the following information as part of the request:

- Request headers
- Query params
- Form data
- XML or JSON payloads
- Request paths

In building its response to the API request, the backend service can return data to the API client, including:

- Response headers
- XML or JSON payloads

This document describes how this data is passed to and from the backend service.

How is request data passed to the backend service?

All data in the request from the API client is passed unchanged to the backend service. It is then up to the backend service to parse the request data as part of handling the request.

How is response data returned to the API client?

All data received in the response from the backend service is passed unchanged to the API client. It is then up to the API client to process any returned data in the response.

How is the request URL passed to the backend service?

The URL used to make a request to the backend service is controlled by the `x-google-backend` extension. This section describes the options for configuring the backend service URL.

Setting the backend service address and path in the OpenAPI spec

In the OpenAPI spec that you use to create an API config, you use the `x-google-backend` extension to specify the URL of the backend service. For example, you specify the backend service in the form:

Backend   **x-google-backend**

Backend	x-google-backend
Cloud Functions	```x-google-backend:\n address: https://GCP_REGION-PROJECT_ID.cloudfunctions.net/hello```
Cloud Run	```x-google-backend:\n address: https://hello-HASH.a.run.app```
App Engine standard environment	```x-google-backend:\n address: https://PROJECT_ID.appspot.com```

In these examples:

1. *GCP_REGION* specifies the GCP region for the deployed backend.
2. *PROJECT_ID* specifies the Google Cloud project ID.
3. *HASH* specifies the unique hash code generated when the Cloud Run service is created.

In addition, the `path` parameter in the OpenAPI spec specifies the endpoint, or *resource,* supported by your API. You can specify an absolute path or one that uses path parameters:

Path	Path with parameters
```path\ns:\n\n/hel\nlo:```	```paths:\n /hello/{name}:```

Generating the backend service URL from an API request

As API Gateway handles a request from the API client, it takes the request URL sent by the API client and translates it into the URL used to make the request to the backend service. Exactly how this translation occurs depends on which *path translation strategy* you are using.

The `path_translation` option to the `x-google-backend` extension supports two path translation strategies:

- `APPEND_PATH_TO_ADDRESS` : The backend service URL is generated by appending the resource path from the client request to the `address` URL of the `x-google-backend` extension.

  Most backend services use `APPEND_PATH_TO_ADDRESS` because it means the backend receives the same resource path as specified by the API client.

- `CONSTANT_ADDRESS` : The backend service URL is constant, as defined by the `address` URL of the `x-google-backend` extension. If the client request contains a resource path, the resource path is added to the backend service URL by using query parameters.

  This method is typically used by <u>Cloud Functions</u>.

For example:

- `APPEND_PATH_TO_ADDRESS`
  - `address` : `https://PROJECT_ID.appspot.com`
  - Without OpenAPI path parameters:
    - OpenAPI path: `/hello`
    - API client request resource path: `/hello`
    - Backend service request URL:
      `https://PROJECT_ID.appspot.com/hello`
  - With OpenAPI path parameters:
    - OpenAPI path: `/hello/{name}`
    - API client request resource path: `/hello/Dave`
    - Backend service request URL:
      `https://PROJECT_ID.appspot.com/hello/Dave`
- `CONSTANT_ADDRESS`
  - `address` : `https://GCP_REGION-PROJECT_ID.cloudfunctions.net/hello`
  - Without OpenAPI path parameters
    - OpenAPI path: `/hello`
    - API client request resource path: `/hello`
    - Backend service request URL: `https://GCP_REGION-PROJECT_ID.cloudfunctions.net/hello`
  - With OpenAPI path parameters
    - OpenAPI path: `/hello/{name}`
    - API client request resource path: `/hello/Dave`
    - Backend service request URL: `https://GCP_REGION-PROJECT_ID.cloudfunctions.net/hello?name=Dave`

Setting `path_translation`

Set `path_translation` as part of setting `x-google-backend` :

```
x-google-backend:
  address: https://GCP_REGION-PROJECT_ID.cloudfunctions.net/hello
  path_translation: [ APPEND_PATH_TO_ADDRESS | CONSTANT_ADDRESS ]`
```

The default value of `path_translation` depends on where you set `x-google-backend` in your OpenAPI spec:

- When `x-google-backend` is used at the top level of the OpenAPI specification, `path_translation` defaults to `APPEND_PATH_TO_ADDRESS` .

- When `x-google-backend` is used at the operation level of the OpenAPI specification, `path_translation` defaults to `CONSTANT_ADDRESS` .

Rate and review