

Publisher Guidance

NOTE:

Onboarding list for new APIs that will go behind Apigee

Overview

1. What is the function/purpose of the API?
2. What type of API? (REST, GraphQL, SOAP?)
3. If REST, what level of [Richardson's Maturity Model](#)? (Level 3 preferred but level 2 is the minimum.)
4. Does the API use JSON, XML or both for their accepted data formats?
5. Will the API be stateless or stateful?
6. What types of data will the API be consuming & returning in payloads? PCI? PII? Proprietary? Binary? (like images, PDFs or other blobs?)
7. Will the API provide cache headers?

API Consumer

Who is the API Consumer?

1. Is the API Consumer a single internal user or internal system? (service account accessing the API? AA# user accessing the API?)
2. Is the API Consumer another internal or external system?
3. Will there ever be a use case of more than one consumer? (Is this API going to be an Application or Enterprise level service?)
4. What types of applications will be calling this API? SPA, PWA, mobile app or server based applications?
5. What CRUD operations are being performed by the consumer and what level of complexity is being used to regulate access control to API endpoints? Will there be various levels of authorization like consumer A can only do READ operations, but consumer B can do all CRUD operations?
6. What level of Security will be used to connect on the Northbound connection (from Consumer to Apigee) and where will authentication & authorization be performed?
 - a. Development Portal credentials?
 - b. OAuth2 with a separate IDP endpoint fronted by Apigee?
 - c. 3rd party?
7. Is there a plan to monetize your APIs and charge consumers? If so, what strategy is going to be used (per endpoint? Per volume? Other strategy?)

Who is the API Owner?

1. Is approval required to consume the API Endpoint?
 - a. If approval is required:
 - i. Is it required at the environment level? (for all API environments? Sandbox doesn't require it but UAT & Production does?)
 - ii. Is it for certain types of CRUD operations?
- Is it by URI endpoints.
1. Are there requirements that need to be met before approval and if so, what are they?
1. Who approves the consumption of the API? (Receive access notifications via email? Via Service Now? Via other method?)
2. Is there a business sponsor/owner?

Performance/SLAs

1. Must have clearly defined SLAs
 - a. What is the expected response time per endpoint operation?
 - b. What are the threshold limits that define a breach of SLA?
 - i. Are there various levels of SLA breach and what are those definitions?
 - c. When should Apigee assume a timeout has happened and terminate the socket & return a 500 status code? (2x max SLA? 3x max SLA?)
2. What are the expected call volumes?
 - a. What is the volume expected during peak hours?

- b. What is the volume expected during off-peak hours?
- c. What is the max calls allowed (Spike Quota look like?)
- 3. What are the expected service availability SLAs?
 - a. Is this API expected to be 24x7? 9am-9pm M-F?
 - b. What classification/category is the service? Platinum, Gold, Silver, or Bronze?
- 4. What Ops/DevOps team is supporting this API?
 - a. Ferguson provided? 3rd party?
 - b. What is their contact information? (Email address, Phone numbers, Manager contact info, Team designation within Service Now and afterhours contact info.)
 - c. What types of monitoring is the Ops/DevOps team doing on the API? SiteScope? Splunk Alerts? xmatters Alerts?

Backend API

- 1. What level of Security will be used to connect on the Southbound connection (from Apigee to backend API)? OAuth 2? mTLS? Basic Auth? appld & appKey?
- 2. Any version control being used? (within the URI itself? Within the request or response payloads?)
- 3. What endpoints does the backend API touch as secondary systems and are they internal, external or SaaS based? (example, ODS Data, Trilogie, HighJump, Oracle or MSSQL DB?)
- 4. What URI endpoints are available to call for the purpose of verifying the API is working properly?

Deliverables

- 1. Must provide an OpenAPI/Swagger document Preferably v2 but v3 is acceptable.
- 2. Must provide a Postman Collection.
- 3. Must provide Visio/Lucidchart diagrams and workflow diagrams.
- 4. Must provide "health checks" URI endpoints to verify API is functioning. (Could be a version endpoint that also hits the services' DB backend or read-only calls that always return guaranteed data.)