API Management Best Practices

API Development Process

When you start creating new elements in your API Management, consider the following recommendations.

Organizations

- We recommend to create organizations as fine-granular as possible, e.g. one organization for each logical group of APIs (purchase, order processing, billing).
- Use a separate, dedicated organization for testing or development.
- Do not test your API in an organization that holds productive data.

Policies

- Policies can be added to clients, plans and APIs, and this has impact on the policy chain. So, give a thought or two on where to add your policy.
- Testing APIs or verifying concepts with policies is much simpler with public APIs.

> The order of policies is important and has a huge impact on the applied rules, e.g. use the BASIC Authentication policy before the Rate Limiting policy. Otherwise unauthorized requests will reduce the rate limit.

Level	Recommendation
API	On API level, you will typically use modification policies, such as URL Rewriting or API Key.
Plan	On plan level, you will typically use limiting policies, such as Rate Limiting. This way, each plan will allow for a different amount of requests.
Client	On client level, you will typically apply authentication and authorization policies, such as BASIC Authentication or Authorization, or other security policies.

For detailed explanations about the usage of policies, the policy chain and the policies available in **Scheer PAS** *API Management*, go to page Policies.

API Key

API keys are not encrypted and visible:

- in the request header,
- to people who have access to API Management metrics/the Log Analyzer,
- in the logs of the integration component (Bridge) if you are using the API Key policy.

So, API keys need to be handled in a secure way - otherwise attackers may be able to use the API key to gain access to your system.

> As per definition, API keys are used to identify technical clients only and, subsequently, to apply related policies. **Do not use API keys to authenticate users.**

Authentication should always be implemented via a dedicated security policy (see Policy Configuration, category **Security Policies** and page API Security: Authentication and Authorization).

Tips and Tricks

Do not delete APIs, plans, or clients and recreate them if you want to change policies or settings. Instead:

- As concerns APIs and clients: Retire the old version, so it will not be callable anymore. Alternatively, skip this step if you want both versions - old and new - to be available.
- Create a new version of the element you want to change.
- Re-publish or re-register the API or client.

API Documentation

On this Page:

Related Pages:

- The API Management User Interface
- The Concepts of API Management
- Getting Started With API Management
- Organizations
- Policies

Related Documentation:

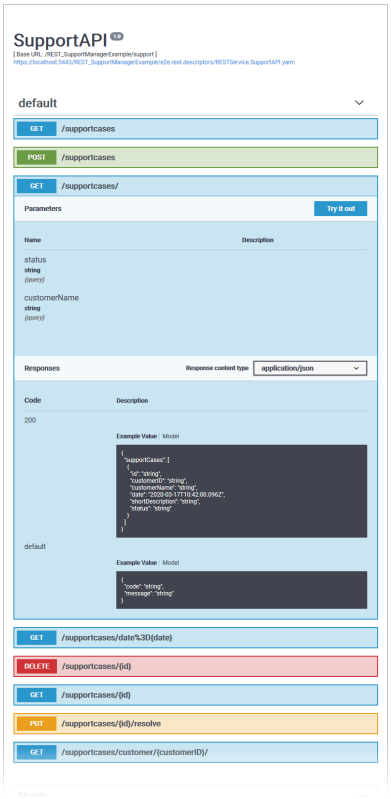
- BRIDGE Documentation
  - Documenting a REST Service
- Swagger Homepage

If your clients have been granted access to a developer portal (see Developer Access to APIs), they can try out your APIs using a testing UI. To be able to easily use your APIs, it is important to provide them with extensive API documentation.

> Adding or removing policies does not enrich the Open API documentation. You need to adjust your documentation manually.

API With Good Documentation

API Lacking Documentation



Refer to Documenting a REST Service for more information on REST documentation and how to add documentation to xUML services.