**Developer Portal**

* API portals provide a centralized place for consumers to discover interactive documentation, tutorials, code snippets, examples and other tools developers need to successfully use your API
* It is a part of anypoint exchange
* Used to share assets to public or within organization
* It allows to view APIs, document, test
* It is used to add detail description. This description has information such as how the api is used, wiki, contact information incase support is needed
* It is easy to use and basic features are provided out of the box
* It allows user to **request for the access** of an API
* It allows to **share the API in public portal**
* It allows to **share resources** between two or more organization

Why?

* It is used to share with different people and get feedback from them. These users cane be customers, stakeholders or developers
* In near future the application will be driven by developers because the dependency on the developers are increasing day by day

So if a developers sees that the API is easy to consume than they will try to implement that API

So every **organization wants to have developer friendly portal**

* This helps to increase collaboration and improve reusability

Features:

* Api notebook
* Api endpoints
* Access to public portal

**API NoteBook**

* API Notebook lets you document an API’s functionality with explanations that accompany code windows.
* Users can change the code and execute it by clicking the Play button in each code block.
* You can create a new API Notebook from the Exchange editor and use JavaScript code in the code blocks.
* It is used by the developers to test
* **It is for interactive documentation. You can add scripts**

[**https://docs.mulesoft.com/api-manager/1.x/tutorial-create-an-api-notebook**](https://docs.mulesoft.com/api-manager/1.x/tutorial-create-an-api-notebook)

**Share Resources**

* **To public**
* **To other organization**
  + Goto access management
  + Goto external access
  + Add domain name of the organization with whom you want to share
  + Then assets can be shared between organization

**Downside of this Portal**

* It is limited in terms of functionality example you can only change the logo
* To customize the portal you need to use API community manager Example of such implementation in HSBC

API Community Manager

* It gives a complete customization capability like provide custom templates, forums, chat rooms
* It allows to custom grouping of your assets
* Split API based on country

Why?

* In order to encourage developers to use your API it is essential to have a nice and user friendly API portals
* To have well documented
* Collaborate with other users and received feedback
* Used for branding

**It is build using salesforce community cloud product**

Why deprecate an API version

1. The API is insecure
2. The API has too many bugs
3. The API does not support important use cases
4. The API is inefficient

While transitioning consumers of your API to an updated version, you can prevent developers from signing up for access to your old API version. As an API administrator, you mark an API as deprecated to remove the Request API Access from the portal page. Existing application contracts remain active but no new contracts can be created for that API version. Deprecated APIs have an indicator on the portal page in place of the request access button, on the API version details page, and on the Developer portal for your organization. A badge in search results indicates that the version is deprecated.

1. In Anypoint Platform, click API Manager.
2. On the API Administration page, click the API version link.
3. On the API version details page, select Deprecate version from the dropdown.