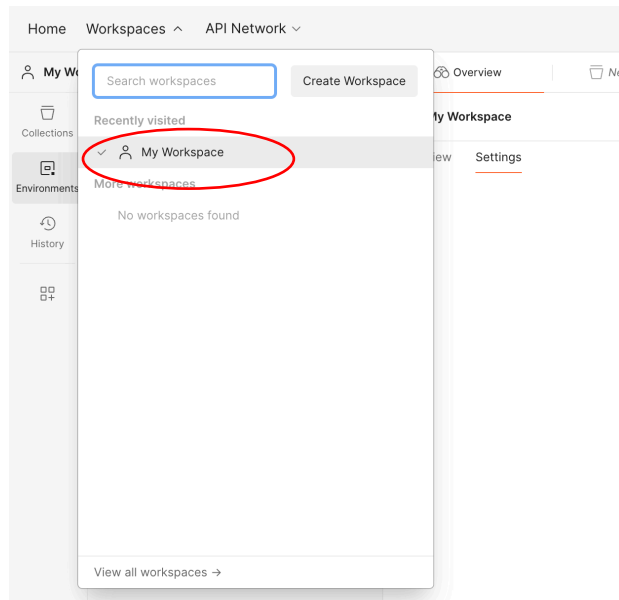


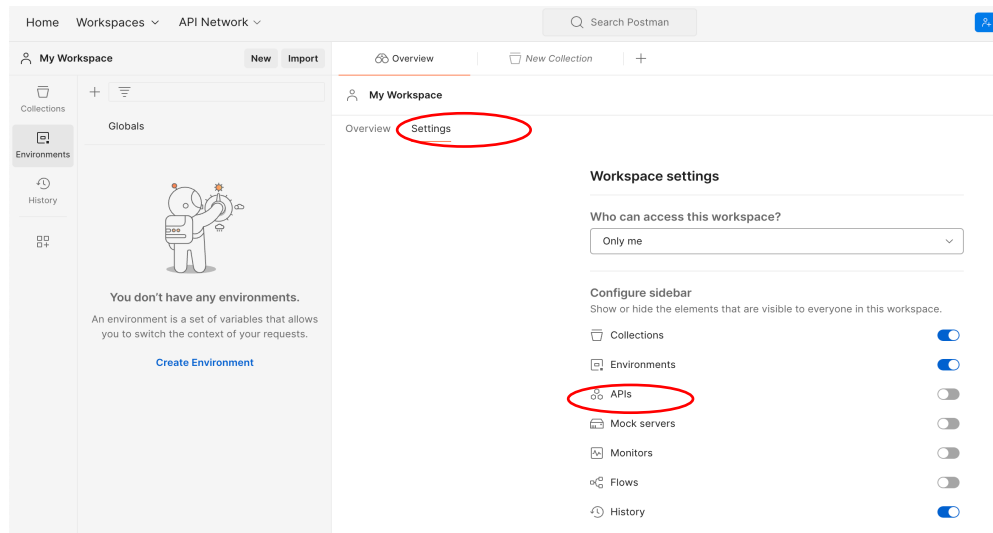
Steps for Creating an API and Documentation in Postman (updated Oct 15, 2024)

1. Create your new API using the Open API Specification 3.0 in your PERSONAL workspace.

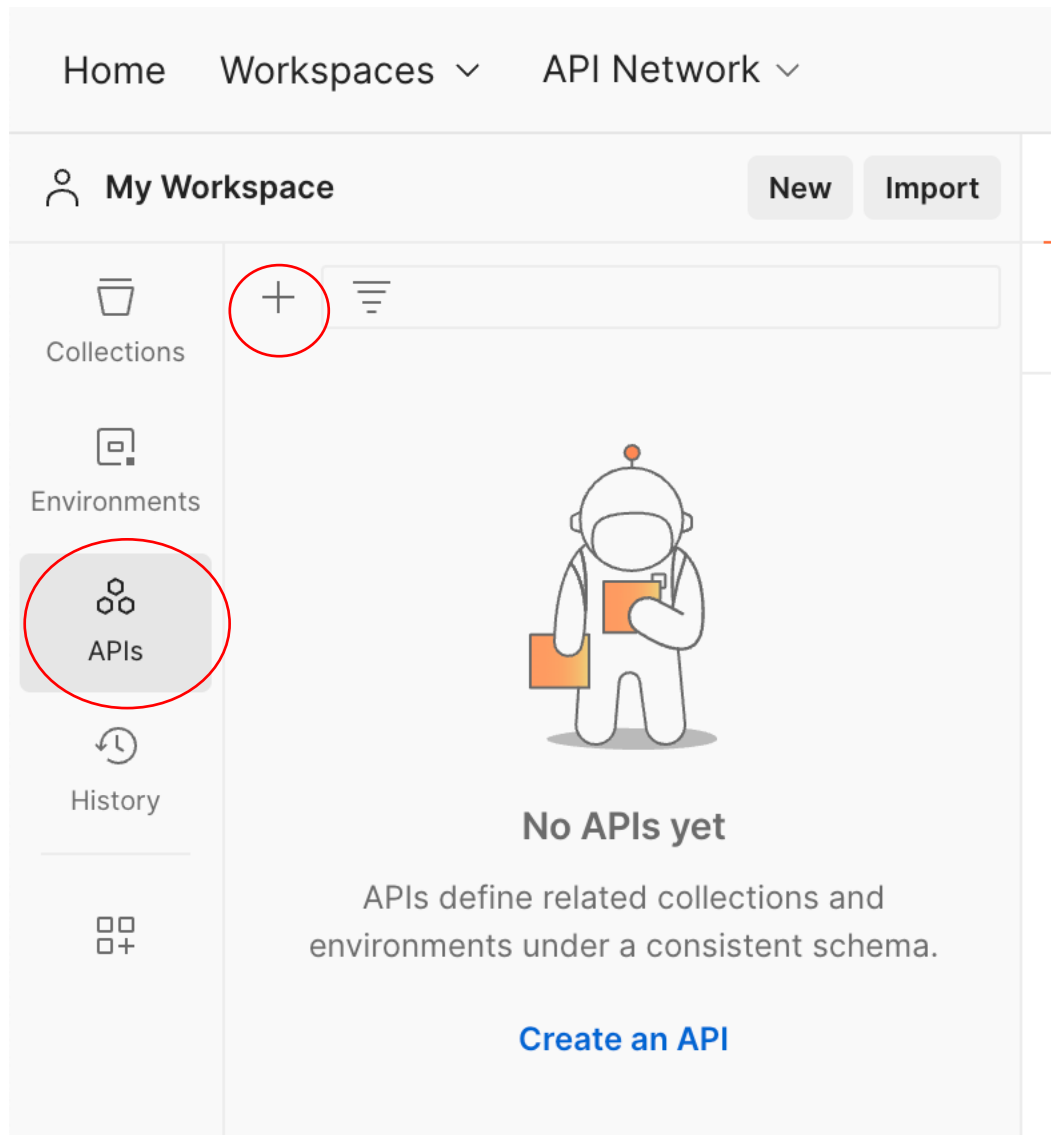
- In Postman , Go to My Workspace.



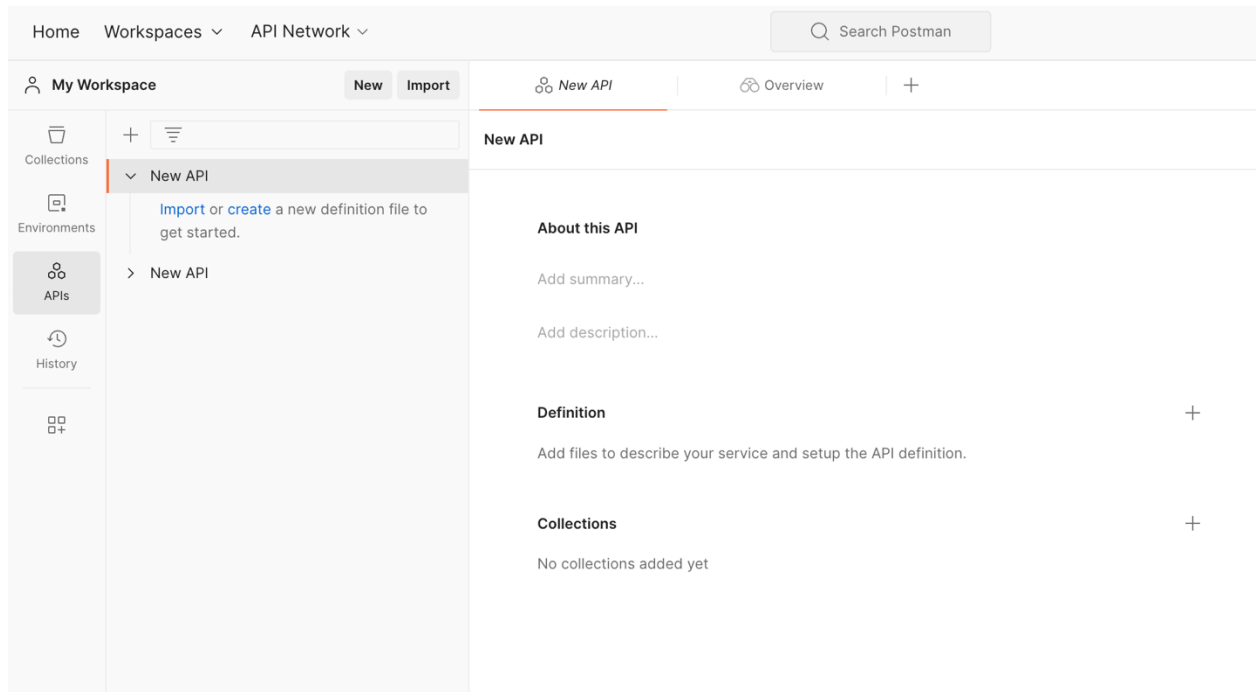
If you don't see the API option in the sidebar then go to settings and add the API option. If you already have it, then skip this step.



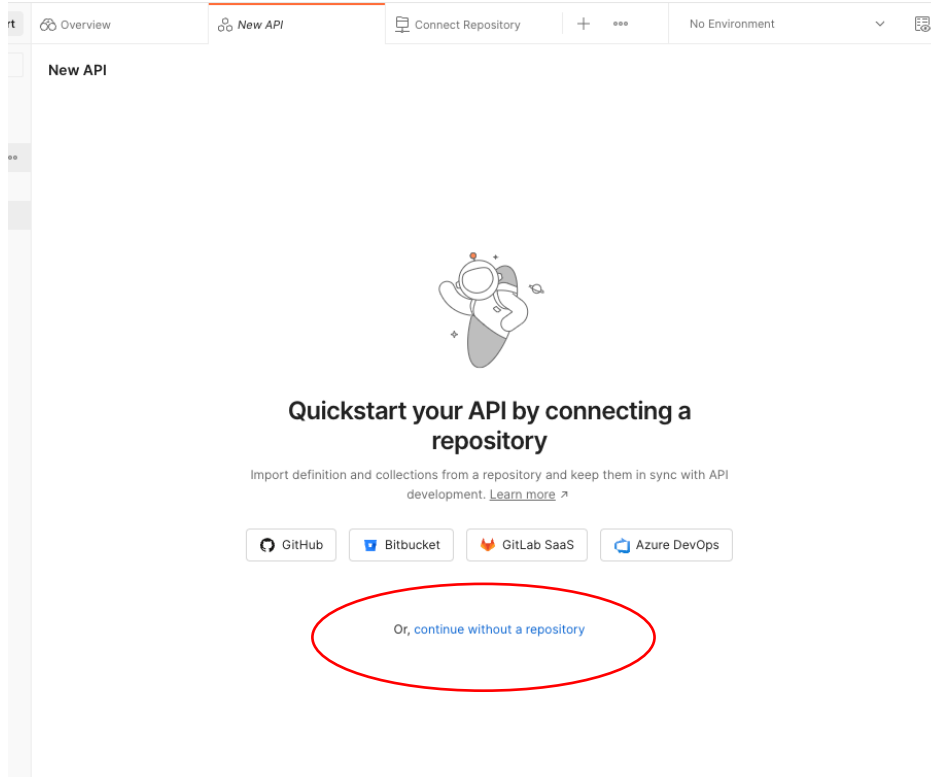
Now go to the API option on the side bar and click the + symbol to create an API



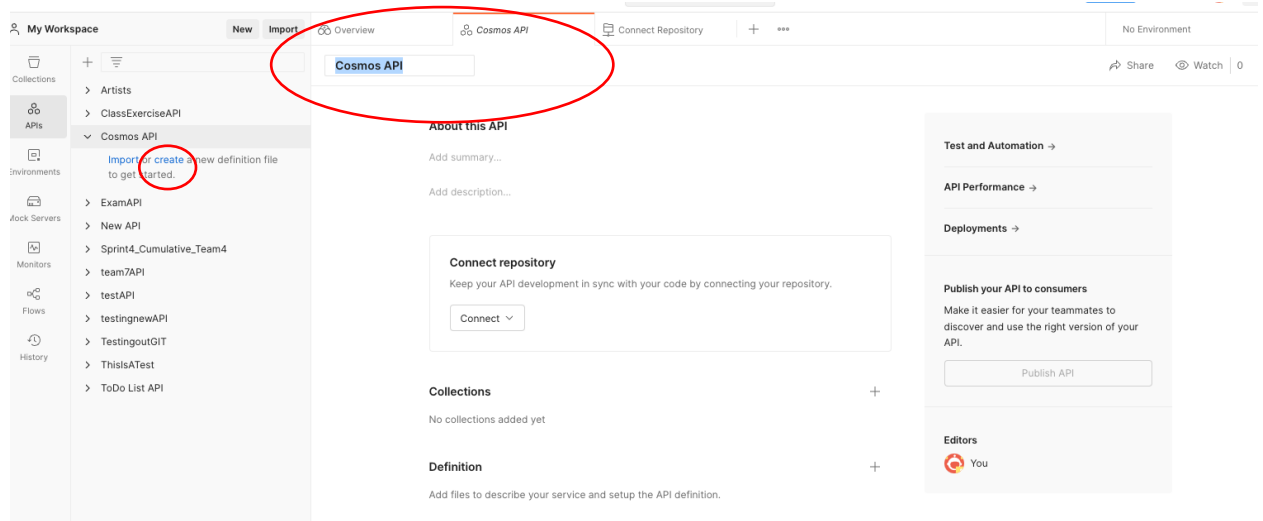
If you are using your personal mail and not joined in the Clemson computer science postman team, Postman assumes you're working on the API by yourself, so it gives you the more standard, direct API creation page like this



- If you join in the Clemson computer science postman team you will get this page
The differences you're seeing in Postman's API creation interface likely stem from **team workspace features**. Connect this API with a repo – SKIP THIS (optional)



- Change the name of your API and then select **Create API** from left hand Side bar (you could also import from a file if you have the definition somewhere saved)



- In the pop-up you need give the Def type = OpenAPI 3.0 and definition format = YAML and then Create Definition

Author definition from scratch

Describe your service using a spec format

Definition type ⓘ

OpenAPI 3.0

Definition format

YAML

☐ Use a boilerplate ⓘ

Create Definition

Cancel

- Note that you can use a boilerplate definition, A prefilled API skeleton will be auto generated then you can fill it in or cut and paste your definition from elsewhere.

Author definition from scratch

Describe your service using a spec format

Definition type ⓘ

OpenAPI 3.0

Definition format

YAML

☒ Use a boilerplate ⓘ

Create Definition

Cancel

- After your definition is complete, Save it

Cosmos API / Definition / index.yaml

OpenAPI 3.0

Save Cancel YAML {}

General

Nothing here yet.

Servers

No servers defined.

Paths

No paths defined.

Components

No components defined.

Security

No security schemes defined.

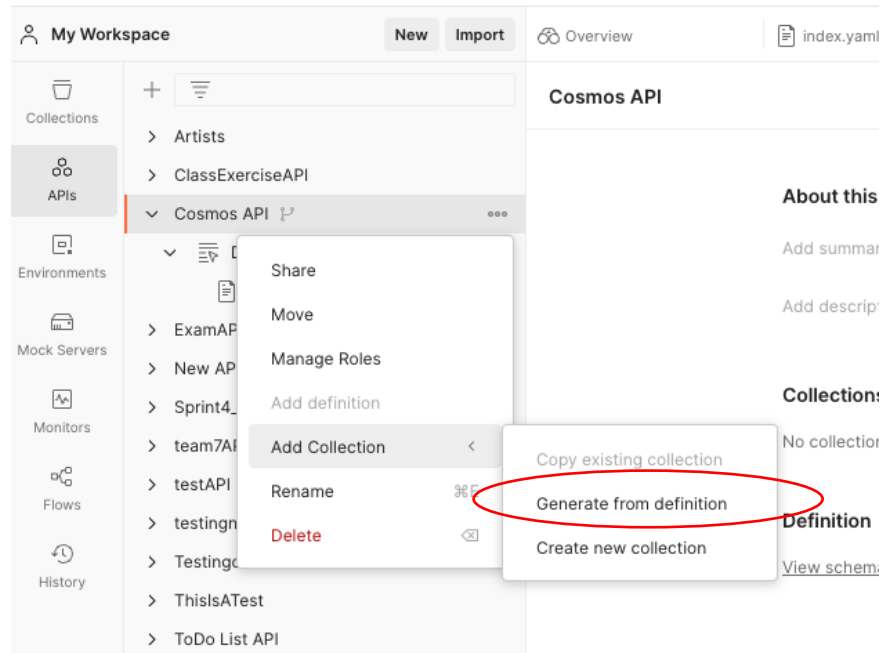
⚠ Outline will be generated once you have a valid definition.

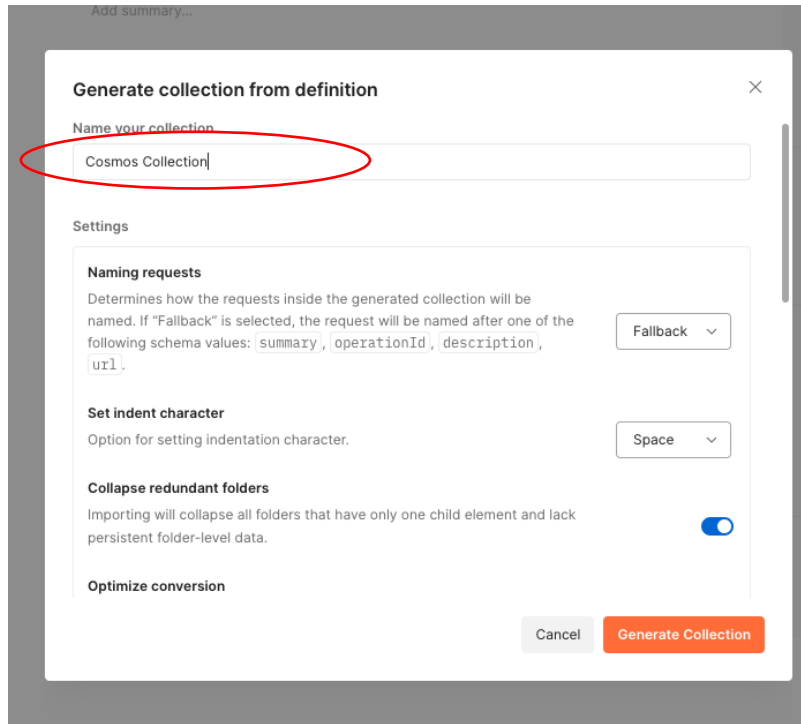
```
1 openapi: "3.0.0"
2 info:
3   version: 1.0.0
4   title: Cosmos
5   description: A sample API that retrieves constellations as an example to demonstrate features in the Open
6   license:
7     name: MIT
8 servers:
9   - url: https://e8c086f7-a5c9-4752-b81f-bfac0d0dc5d2.mock.pstmn.io
10 paths:
11   /cosmos:
12     get:
13       description: Returns all constellations in this sample dataset
14       summary: List all cosmos
15       operationId: listCosmos
16       tags:
17         - cosmos
18       responses:
19         "200":
20           description: An array of constellations
21           content:
22             application/json:
23               schema:
24                 $ref: "#/components/schemas/Cosmos"
25           default:
26             description: unexpected error
27             content:
28               application/json:
29                 schema:
30                   $ref: "#/components/schemas/Error"
31 /cosmos/{cosmoId}:
32   get:
33     description: Returns a constellation based on a single ID
34     summary: Info for a specific constellation
35     operationId: showCosmoById
36     tags:
37       - cosmos
38     parameters:
39       - name: cosmoId
40         in: path
```

Violations found in definition: Syntax (0)

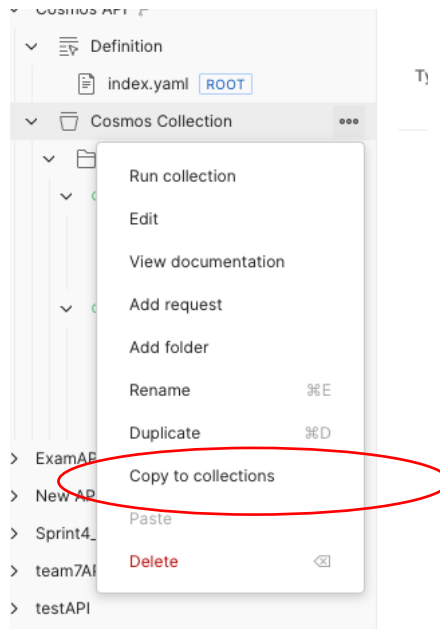
2. Create a Collection from your API Definition

- Select the API in the left Nav and click on ... next to the API name
- On the Add Collection pop-up, select Generate from definition and Name the collection

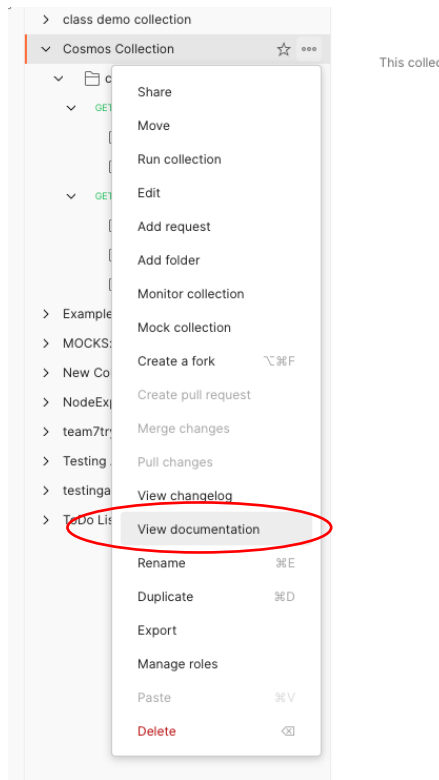




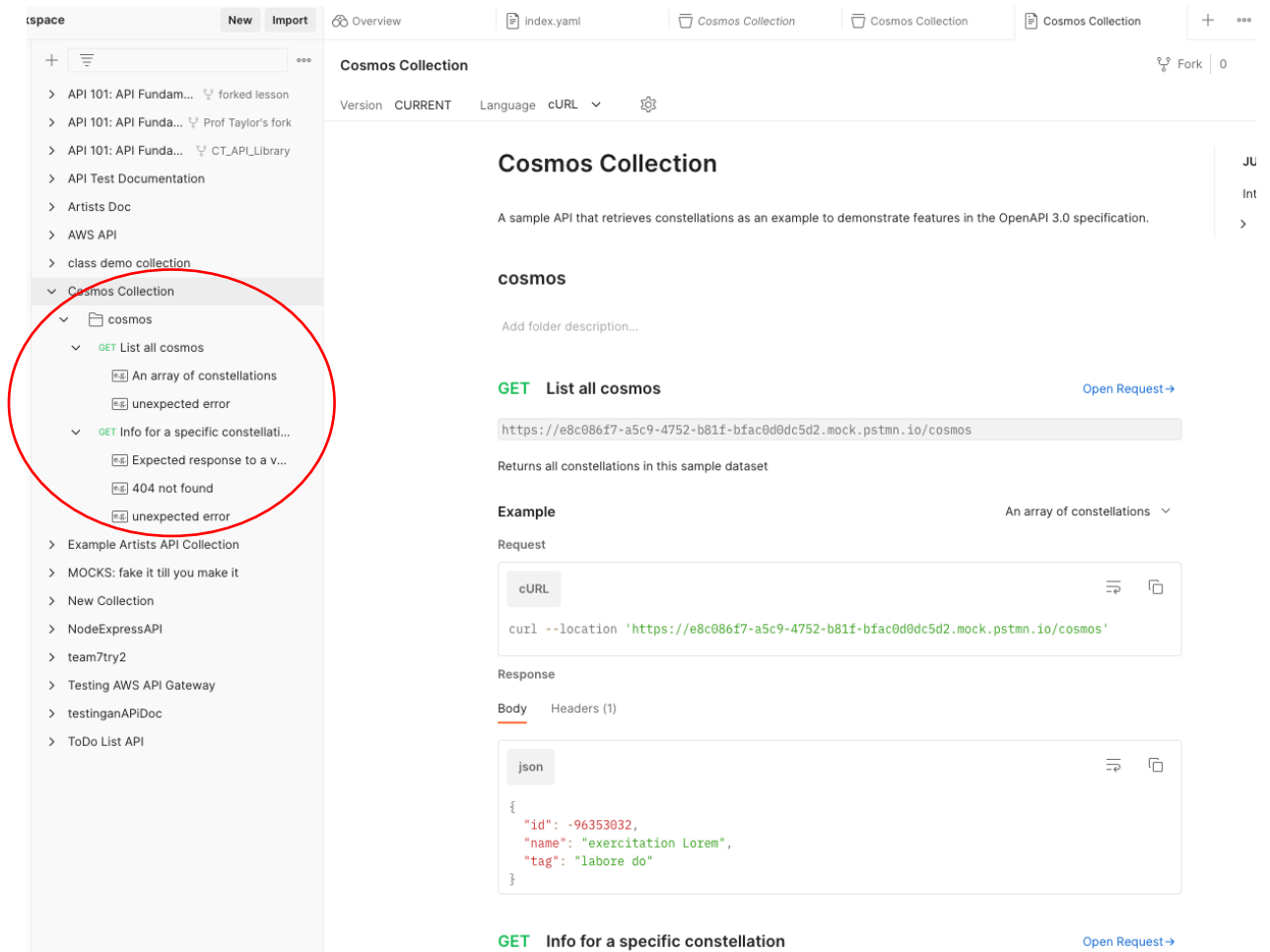
Your collection will show up under the API but you can copy it to the Collections area



The collection itself is what I will use to look at documentation of the API by selecting View Documentation; note that the documentation is only as good as your specification (garbage in -> garbage out)



Here is the Cosmos Collection and documentation:



space **New** **Import** Overview index.yaml Cosmos Collection Cosmos Collection Cosmos Collection + ***

Cosmos Collection Fork 0

Version CURRENT Language cURL ⚙️

Cosmos Collection

A sample API that retrieves constellations as an example to demonstrate features in the OpenAPI 3.0 specification.

cosmos

Add folder description...

GET List all cosmos [Open Request →](#)

`https://e8c086f7-a5c9-4752-b81f-bfac0d0dc5d2.mock.pstmn.io/cosmos`

Returns all constellations in this sample dataset

Example [An array of constellations](#)

Request

cURL

```
curl --location 'https://e8c086f7-a5c9-4752-b81f-bfac0d0dc5d2.mock.pstmn.io/cosmos'
```

Response

Body Headers (1)

json

```
{  "id": -96353032,  "name": "exercitation Lorem",  "tag": "labore do"}
```

GET Info for a specific constellation [Open Request →](#)