API Methods

Using Postsman

Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

https://www.postman.com

Signup at Create Postman Account and Create a free account

Optional: Download Postman, alternatively just use it in your browser.

Workalong

In **Postman**, make a new Collection, name it whatever you want (eg. "JSONPlaceholder")

We will be using {JSON} Placeholder, the free fake API for testing and prototyping, and make requests to that using Postman.

Use the JSONPlaceholder Guide as Documentation.

Important: resources will not be really updated on the server but it will be faked as if.

HTTP GET Request Method

1. Make a GET request to `https://jsonplaceholder.typicode.com/todos`, name it "Get all Todos". Save. Press [Send] to fetch all todos.
2. Make a GET request to `https://jsonplaceholder.typicode.com/posts/1`, name it "Get a single Todo"...
3. Make a GET request to `https://jsonplaceholder.typicode.com/posts`, name it "Get all Posts"; then try adding a parameter: `/posts?userId=1`
4. Make a GET request to `https://jsonplaceholder.typicode.com/posts/2`, name it "Get all Posts"

> Tip: When you've made the first request, you can use "duplicate" and make some small changes to reduce typing...

5. Make a GET request to `https://jsonplaceholder.typicode.com/posts/2/comments`, name it "Get all comments for a single post" (this is an example of a *nested route*)
6. Make a GET request to `https://jsonplaceholder.typicode.com/comments?postId=2`, name it "Get all comments for a single post with parameter"

> Quastion: *Are there any differences in the result of the last two?*

7. Make a GET request to `https://jsonplaceholder.typicode.com/photos`, name it "Get all Photos"
8. Make a GET request to `https://jsonplaceholder.typicode.com/photos/13`, name it "Get a single Photo"

> Tip: Checkout the "images" that are in Photos.

HTTP POST Request Method

9. Make a POST request to `https://jsonplaceholder.typicode.com/posts`, and name it "Post new Post".

> Note: If you [Send] it without a body, it will make a new Post, but it will have no content (except an id), so we need to add a body.

Go to the Body-tab, choose "raw" and make sure the dropdown is "JSON", then add the body:

```
{
    "title": "foo",
    "body": "bar",
    "userId": 1
}
```

Now, you get a new Post.

10. Make a POST request to `https://jsonplaceholder.typicode.com/todos`, and name it "Post new Todo", and add the following body:

```
{
    "title": "Whatever",
    "completed": false,
    "userId": 2
}
```

HTTP PUT Request Method

11. Make a PUT request to `https://jsonplaceholder.typicode.com/posts/3`, name it "Update Post", and add the body:

```
{
    "id": 3,
    "title": "Updated post",
    "body": "We put this here",
    "userId": 1
}
```

> Note the naming. Do we really need to add the Method to the name of each request?

12. Make a PUT request to `https://jsonplaceholder.typicode.com/todos/3`, name it "Update Todo", and add a body:

```
{
    "title": "New todo",
    "completed": false,
    "userId": 2
}
```

> Notice we didn't use the ID, but it accepted it without and updated the correct one because of the `id` in the URL.
> But if we omit `userId`, that will not be "created" for us.

HTTP PATCH Request Method

13. Make a PATCH request to `https://jsonplaceholder.typicode.com/posts/1`, name it "Update Post", and add the body:

```
{
    "title": "Updated title"
}
```

> Notice: Here we *just* updated the title, while the rest of the Post was left as it was.

14. Make a PATCH request to `https://jsonplaceholder.typicode.com/todos/1`, name it "Update Todo", and add the body:

```
{
    "completed": true
}
```

HTTP DELETE Request Method

15. Make a DELETE request to `https://jsonplaceholder.typicode.com/posts/2`, name it "Delete Post".

> Note: It returns with Status 200 OK and an empty object.

16. Make a DELETE request to `https://jsonplaceholder.typicode.com/todos/3`, name it "Delete Todo".

Noroff API

Noroff API Documentation

- Making a new Collection (named "Noroff API" or similar)
- Making a variable of the Base URL, found on: About
- Add requests for Basic Endpoints > Books
- Add requests for Basic Endpoints > Cat Facts
- Add requests for Basic Endpoints > Jokes

> Now, look under **E-commerce Endpoints**

Todos

Postman

1. Make a Postman Collection for your Project
2. Familiarize yourselves with the documentation for your projects API:
    - Rainy Days
    - GameHub
    - Square Eyes
3. Make and save a request for each of the (main) endpoints for your project
    - All of them have one for all entries and another for a single entry
4. Make Collections for other *free* APIs you want to use, like the AmiiboAPI.

Help: Intro to Postman | Part 1: Send a Request (09:04)

Todos

Mollify

Read Postman, and do the Lesson Task

Read HTTP GET Request Method

Read HTTP POST Request Method

Read HTTP PATCH Request Method

Read HTTP DELETE Request Method**

Read HTTP PUT Request Method

Read Handling `fetch` Errors, and do the Lesson Task**

(** may be wrongly named as `HTTP POST Request Method` in the menu)