How to test Axios in Jest by Example

NOVEMBER 18, 2019 BY ROBIN WIERUCH - EDIT THIS POST

Every once in a while we need to test API requests. Axios is one of the most popular JavaScript libraries to fetch data from remote APIs. Hence, we will use Axios for our data fetching example -- however, the following tutorial should make it possible to exchange axios with any other data fetching library.

```
import axios from 'axios';

export const API = 'https://hn.algolia.com/api/v3';

export const fetchData = async query => {
  const url = `${API}/search?query=${query}`;

  return await axios.get(url);
};
```

```
fetchData('react');
```

In this data fetching example with axios, we provide a query parameter to the function to search on Hacker News via its API. If the request is successful, we will return the response. If the request runs into a network error, the function will throw an error which we would have to capture outside of it.

Now we are going to use Jest to test the asynchronous data fetching function. Jest is used as a test runner (alternative: Mocha), but also as an assertion utility (alternative: Chai). In addition, it comes with utilities to spy, stub, and mock (asynchronous) functions. That's how we will use Jest to mock Axios.

```
import { fetchData } from './';

describe('fetchData', () => {
  it('fetches successfully data from an API', async () => {

  });

  it('fetches erroneously data from an API', async () => {

  });
});
```

First, we will mock Axios with Jest. And second, we will give Axios' get method a mock implementation to resolve and reject a promise for both tests.

```
import axios from 'axios';

import { fetchData } from './';

jest.mock('axios');

describe('fetchData', () => {
  it('fetches successfully data from an API', async () => {
    const data = {
      data: {
        hits: [
          {
            objectID: '1',
            title: 'a',
          },
          {
            objectID: '2',
            title: 'b',
          },
        ]
```

```
      ],
    },
  };

  axios.get.mockImplementationOnce(() => Promise.resolve(data));
});

it('fetches erroneously data from an API', async () => {
  const errorMessage = 'Network Error';

  axios.get.mockImplementationOnce(() =>
    Promise.reject(new Error(errorMessage)),
  );
});
});
```

Last but not least, we will make our assertion with Jest in the cases of resolving the promise successfully or erroneously:

```
import axios from 'axios';

import { fetchData } from './';

jest.mock('axios');

describe('fetchData', () => {
  it('fetches successfully data from an API', async () => {
    const data = {...};

    axios.get.mockImplementationOnce(() => Promise.resolve(data));

    await expect(fetchData('react')).resolves.toEqual(data);
  });

  it('fetches erroneously data from an API', async () => {
    const errorMessage = 'Network Error';

    axios.get.mockImplementationOnce(() =>
      Promise.reject(new Error(errorMessage)),
    );

    await expect(fetchData('react')).rejects.toThrow(errorMessage);
  });
});
```

As bonus, we can also assert that Axios' get has been called with the correct URL:

```
import axios from 'axios';
```

```
import { fetchData, API } from './';

jest.mock('axios');

describe('fetchData', () => {
  it('fetches successfully data from an API', async () => {
    const data = {...};

    axios.get.mockImplementationOnce(() => Promise.resolve(data));

    await expect(fetchData('react')).resolves.toEqual(data);

    expect(axios.get).toHaveBeenCalledWith(
      `${API}/search?query=react`,
    );
  });

  ...
});
```

That's it for creating a Jest mock for Axios by going through one example. You can find this Axios mocking with Jest example in this GitHub repository. A few more thoughts:

- If you want to mock a post instead of a get request for Axios, just apply the `mockImplementationOnce()` for `axios.post` instead of `axios.get`.
- If you want to have a proper network error, for instance a status code 404, then you have to create a custom JavaScript error for it.
- Last but not least, head over to this tutorial if you want to see how data fetching with Axios can be tested in React with Jest and Enzyme.

Show Comments

KEEP READING ABOUT REACT

HOW TO FETCH DATA IN REACT

Newcomers to React often start with applications that don't need data fetching at all. Usually they are confronted with Counter, Todo or TicTacToe applications. That's good, because data fetching adds...

HOW TO FETCH DATA WITH REACT HOOKS?

In this tutorial, I want to show you how to fetch data in React with Hooks by using the state and effect hooks. We will use the widely known Hacker News API to fetch popular articles from the...