

# Exercise 2 - Async/Await Error Handling with Fetch API

## Objective:

To create a JavaScript function using async/await that fetches users from the JSONPlaceholder API and displays them on the webpage. If there's an error, it will display a nicely styled error message on the screen.

## Preparation:

- Observe the provided HTML and CSS, focusing on the users' container and individual user structures.
- Familiarize yourself with the endpoint `https://jsonplaceholder.typicode.com`, specifically the `/users` endpoint.
- You can use some of the code from the third exercise in the 'Promises' section.

## Instructions:

1. **Get the Containers:** Start by selecting the div elements that will contain the users and the error message using the `document.getElementById` method.
2. **Create an Async Function:** Define an asynchronous function `fetchUsers` that will contain the following steps.
3. **Fetch the Data:** Within the function, utilize the `fetch` method with async/await to get users from the JSONPlaceholder API. The endpoint is `'https://jsonplaceholder.typicode.com/users'`.
4. **Handle the Response:** After fetching, check if the response is OK using an if statement. If not, throw an error with a message like `'Network response was not ok'`.
5. **Parse the JSON:** If the response is OK, parse it to JSON using the `await response.json()` method.
6. **Iterate Through Users:** Once you have the JSON, iterate through the users, creating new div elements for each user. Include their name and email in the content.

7. **Append Users to Container:** Append the created div elements to the users' container on the page.
8. **Handle Errors:** Use a `catch` block within a try-catch statement in your async function to handle errors. This block will catch any errors thrown in the previous steps, including response errors and JSON parsing errors. Display the error message in the error container on the page.
9. **Invoke the Function:** Call the `fetchUsers` function to initiate the fetching process.

## Error Handling Insights:

Errors can occur in several ways:

- **Network Errors:** Such as no internet connection or failure to reach the server.
- **Response Errors:** The server might respond with an error status code.
- **Parsing Errors:** If the server does not return valid JSON, parsing it will result in an error.

## Tips for Debugging:

- Use `console.error` to log the errors in the catch block.
- Use the browser's developer tools to see the network requests and responses.

## Expected Outcome:

On successful execution, the page should display the users fetched from the API. If an error occurs, a nicely styled error message should be displayed on the screen.