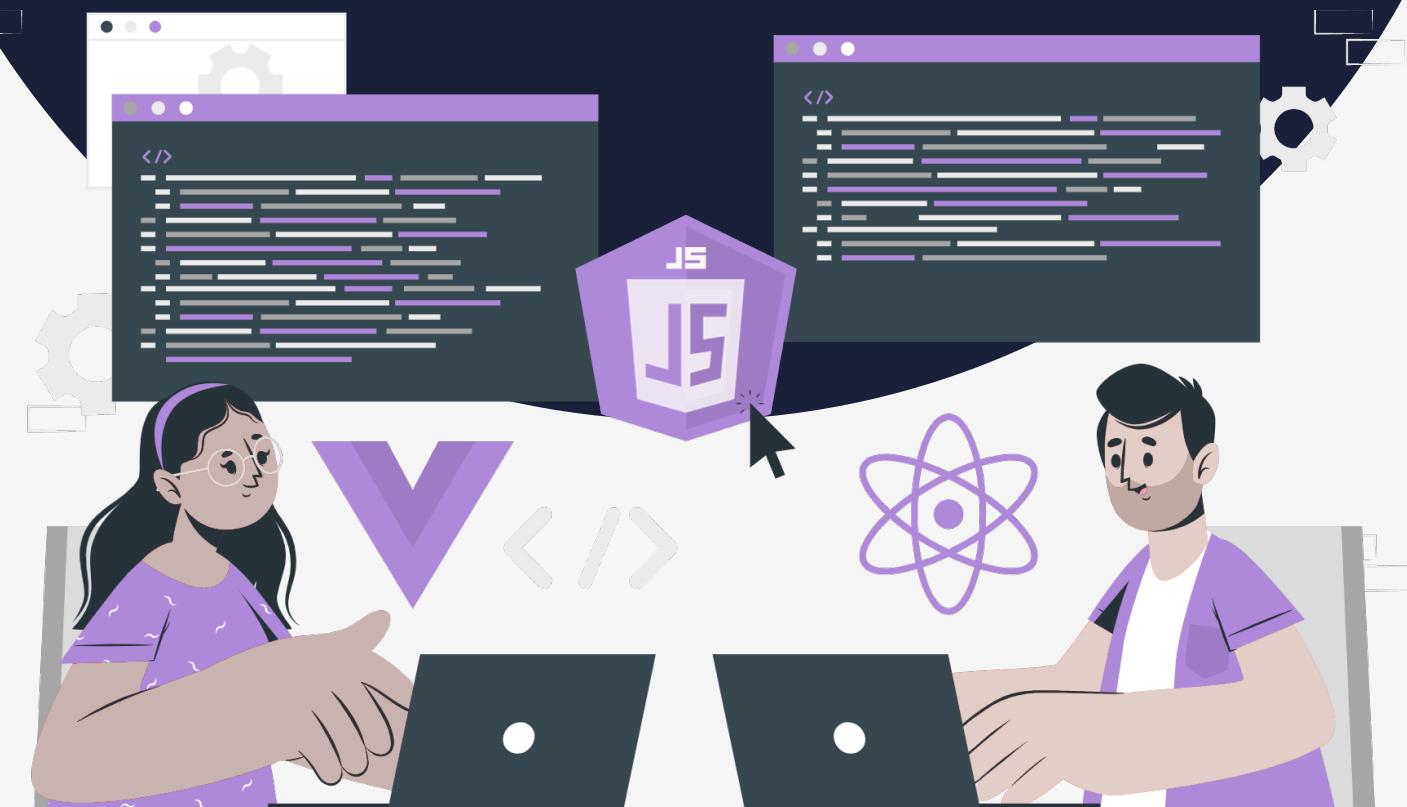


Lesson:

Fetch API



Topics Covered:

1. Introduction to Networks and APIs.
2. Introduction to Fetch API.
3. Features of Fetch.
4. How to use Fetch?
5. Implementation.

Network requests and APIs are fundamental to modern web development. In simple terms, a network request is a process of sending a message from a client (usually a web browser) to a server and receiving a response. This process is often used to retrieve data from a server, such as HTML pages, images, or other resources, or to send data to the server, such as form submissions or user input.

APIs (Application Programming Interfaces) are interfaces that define how software components should interact with each other. In the context of web development, an API is typically a set of rules and protocols that enable a client to interact with a server and retrieve or manipulate data. This data can be in various formats, such as JSON or XML, and is often used to create dynamic and interactive web applications.

We will be looking into the api's their creation, formats, and type of requests in further lectures.

The Fetch API was introduced as part of the web standards effort to modernize and simplify web development by providing a more flexible and powerful way to make network requests in JavaScript. Before the Fetch API, developers primarily used the XMLHttpRequest (XHR) API to make network requests. However, XHR had some limitations so it was replaced by Fetch API.

Fetch API provides a simpler and more powerful way to make network requests. It uses a promise-based syntax and it is a more flexible request and response API. The Fetch API is now widely used in modern web development and has helped to streamline the process of making network requests and processing responses in JavaScript.

Features of Fetch API.

1. Simpler and more intuitive syntax: The Fetch API uses a simpler and more intuitive syntax, with a promise-based API that makes it easier to work with asynchronous data.
2. Better support for new features: The Fetch API is designed to support newer web features, which are not well-supported by earlier network requests.
3. Improved error handling: The Fetch API provides better error handling, with clear and consistent error messages that are easier to understand and troubleshoot.
4. Request and response objects: The Fetch API provides a request object that represents the request being made and a response object that represents the response returned by the server.

How to Use Fetch?

To use the Fetch API, we need to call the `fetch()` method on the client side.

The `fetch()` method requires one parameter, the URL to request, and returns a promise.

```
fetch("URL") // URL of the API end-point
  .then((response) => response.json())
  .then((data) => console.log(data));
```

The `fetch()` methods optionally takes another parameter of options array. We will be looking into it the further lectures.

The `fetch()` method returns a promise that resolves to a Response object when the request is completed. The Response object represents the HTTP response returned by the server and provides access to the response headers, status, and body.

The `fetch()` method is not available by default in Node.js, to use the `fetch()` method in a Node.js application, we need to install the `node-fetch` package using `npm`. We will be looking into it in further sections. For demonstration, we will be implementing the `fetch()` method within the script tag of an HTML document.

Now, let's look at an example where we will be using the `fetch()` method to get quotes from an open-source API.

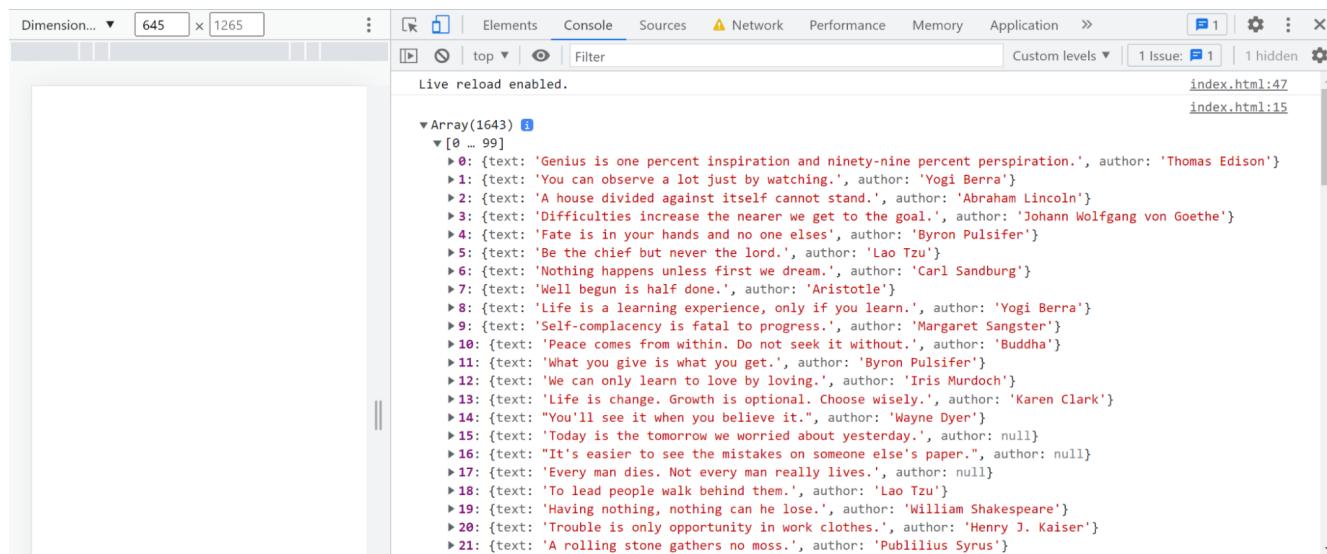
We will be using "<https://type.fit/api/quotes>" API endpoint. The API endpoint returns an array of inspirational quotes and their authors.

Let's make use of them and print the quotes onto the console.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    ←— JS Starts —→

    <script>
      fetch("https://type.fit/api/quotes")
        .then((response) => response.json())
        .then((data) => console.log(data));
    </script>

    ←— JS Ends —→
  </body>
</html>
```



```

Array(1643)
  ▼[0 ... 99]
    ▷ 0: {text: 'Genius is one percent inspiration and ninety-nine percent perspiration.', author: 'Thomas Edison'}
    ▷ 1: {text: 'You can observe a lot just by watching.', author: 'Yogi Berra'}
    ▷ 2: {text: 'A house divided against itself cannot stand.', author: 'Abraham Lincoln'}
    ▷ 3: {text: 'Difficulties increase the nearer we get to the goal.', author: 'Johann Wolfgang von Goethe'}
    ▷ 4: {text: 'Fate is in your hands and no one else's', author: 'Byron Pulsifer'}
    ▷ 5: {text: 'Be the chief but never the lord.', author: 'Lao Tzu'}
    ▷ 6: {text: 'Nothing happens unless first we dream.', author: 'Carl Sandburg'}
    ▷ 7: {text: 'Well begun is half done.', author: 'Aristotle'}
    ▷ 8: {text: 'Life is a learning experience, only if you learn.', author: 'Yogi Berra'}
    ▷ 9: {text: 'Self-complacency is fatal to progress.', author: 'Margaret Sangster'}
    ▷ 10: {text: 'Peace comes from within. Do not seek it without.', author: 'Buddha'}
    ▷ 11: {text: 'What you give is what you get.', author: 'Byron Pulsifer'}
    ▷ 12: {text: 'We can only learn to love by loving.', author: 'Iris Murdoch'}
    ▷ 13: {text: 'Life is change. Growth is optional. Choose wisely.', author: 'Karen Clark'}
    ▷ 14: {text: "You'll see it when you believe it.", author: 'Wayne Dyer'}
    ▷ 15: {text: 'Today is the tomorrow we worried about yesterday.', author: null}
    ▷ 16: {text: "It's easier to see the mistakes on someone else's paper.", author: null}
    ▷ 17: {text: 'Every man dies. Not every man really lives.', author: null}
    ▷ 18: {text: 'To lead people walk behind them.', author: 'Lao Tzu'}
    ▷ 19: {text: 'Having nothing, nothing can he lose.', author: 'William Shakespeare'}
    ▷ 20: {text: 'Trouble is only opportunity in work clothes.', author: 'Henry J. Kaiser'}
    ▷ 21: {text: 'A rolling stone gathers no moss.', author: 'Publius Syrus'}

```

The code fetches data from the quotes API using the `fetch` method, which returns a Promise. The response from the API is converted to JSON format using the `response.json()` method, which also returns a Promise. The data returned from the API is then logged to the console using the `console.log()` method.

Using `async-await`.

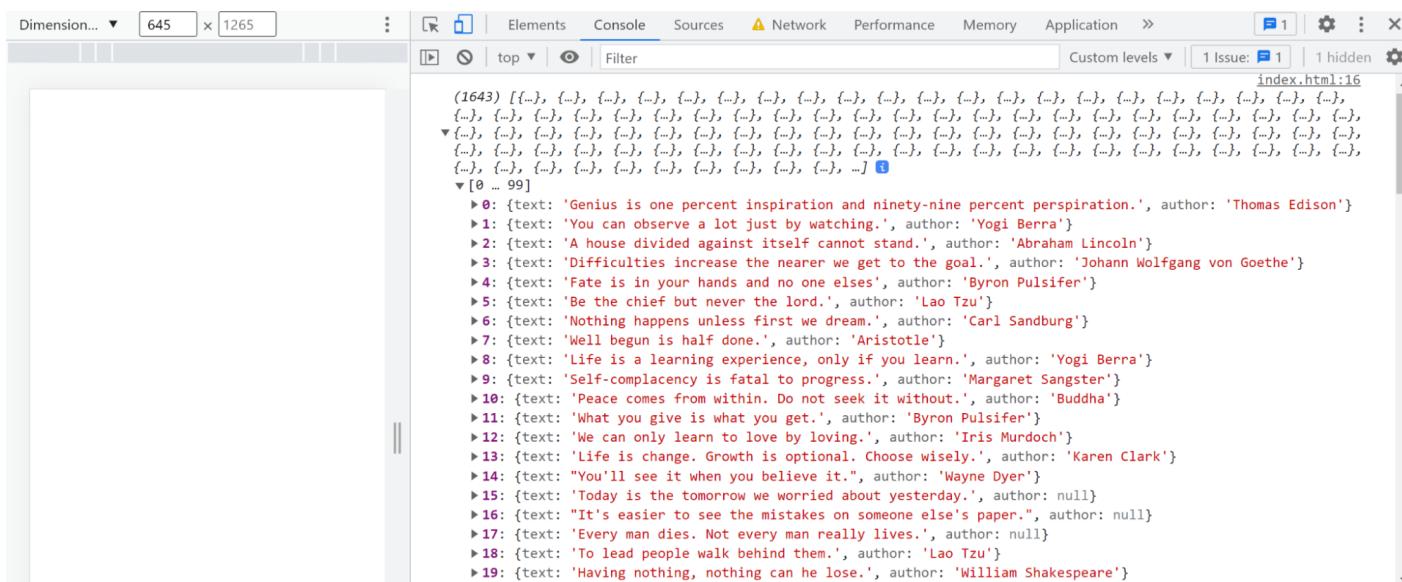
```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <!-- JS Starts -->
    <script>
      async function getQuotes() {
        const response = await fetch("https://type.fit/api/quotes");
        const data = await response.json();
        console.log(data);
      }

      // Calling the function
      getQuotes();
    </script>

    <!-- JS Ends -->
  </body>
</html>

```



```
(1643) [{}]
  ↴ 0: {text: 'Genius is one percent inspiration and ninety-nine percent perspiration.', author: 'Thomas Edison'}
  ↴ 1: {text: 'You can observe a lot just by watching.', author: 'Yogi Berra'}
  ↴ 2: {text: 'A house divided against itself cannot stand.', author: 'Abraham Lincoln'}
  ↴ 3: {text: 'Difficulties increase the nearer we get to the goal.', author: 'Johann Wolfgang von Goethe'}
  ↴ 4: {text: 'Fate is in your hands and no one else', author: 'Byron Pulsifer'}
  ↴ 5: {text: 'Be the chief but never the lord.', author: 'Lao Tzu'}
  ↴ 6: {text: 'Nothing happens unless first we dream.', author: 'Carl Sandburg'}
  ↴ 7: {text: 'Well begun is half done.', author: 'Aristotle'}
  ↴ 8: {text: 'Life is a learning experience, only if you learn.', author: 'Yogi Berra'}
  ↴ 9: {text: 'Self-complacency is fatal to progress.', author: 'Margaret Sangster'}
  ↴ 10: {text: 'Peace comes from within. Do not seek it without.', author: 'Buddha'}
  ↴ 11: {text: 'What you give is what you get.', author: 'Byron Pulsifer'}
  ↴ 12: {text: 'We can only learn to love by loving.', author: 'Iris Murdoch'}
  ↴ 13: {text: 'Life is change. Growth is optional. Choose wisely.', author: 'Karen Clark'}
  ↴ 14: {text: "You'll see it when you believe it.", author: 'Wayne Dyer'}
  ↴ 15: {text: 'Today is the tomorrow we worried about yesterday.', author: null}
  ↴ 16: {text: "It's easier to see the mistakes on someone else's paper.", author: null}
  ↴ 17: {text: 'Every man dies. Not every man really lives.', author: null}
  ↴ 18: {text: 'To lead people walk behind them.', author: 'Lao Tzu'}
  ↴ 19: {text: 'Having nothing, nothing can he lose.', author: 'William Shakespeare'}
```

The code is to fetch data from the quotes API using the `async/await` syntax in JavaScript.

The `getQuotes()` function is defined as an asynchronous function, which means it uses the `await` keyword to wait for a Promise to resolve before continuing with the next line of code. Within the function, the `fetch()` method is used to make a request to the API, and the response is saved to the `response` variable using the `await` keyword.

Then, the response is parsed as JSON using the `response.json()` method and saved to the `data` variable using the `await` keyword. Finally, the resulting data is logged to the console using `console.log()`.

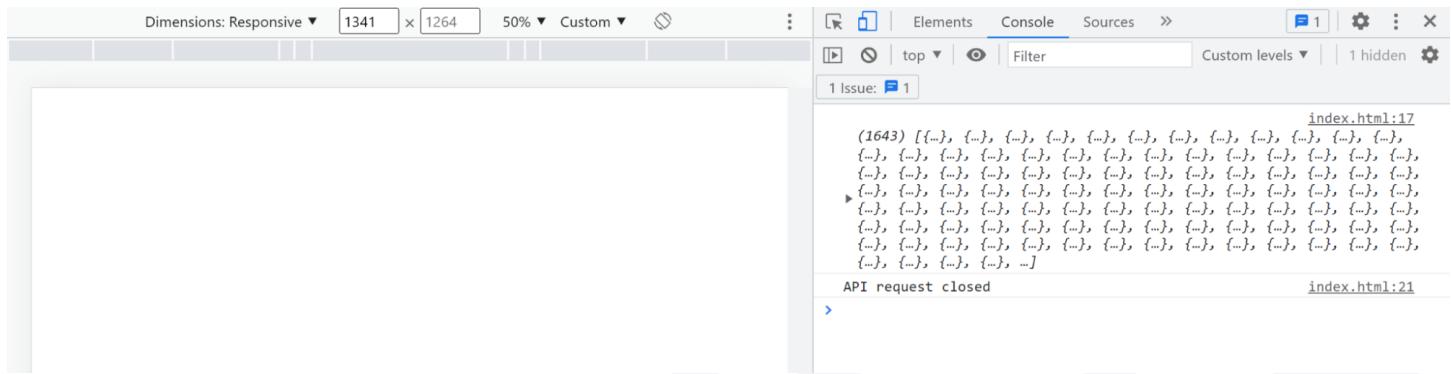
Error Handling in `async-await`.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <!-- JS Starts -->
    <script>
      async function getQuotes() {
        try {
          const response = await fetch("https://type.fit/api/quotes");
          const data = await response.json();
          console.log(data);
        } catch (error) {
          console.log(error);
        } finally {
          console.log("API request closed");
        }
      }
    </script>
  </body>
</html>
```

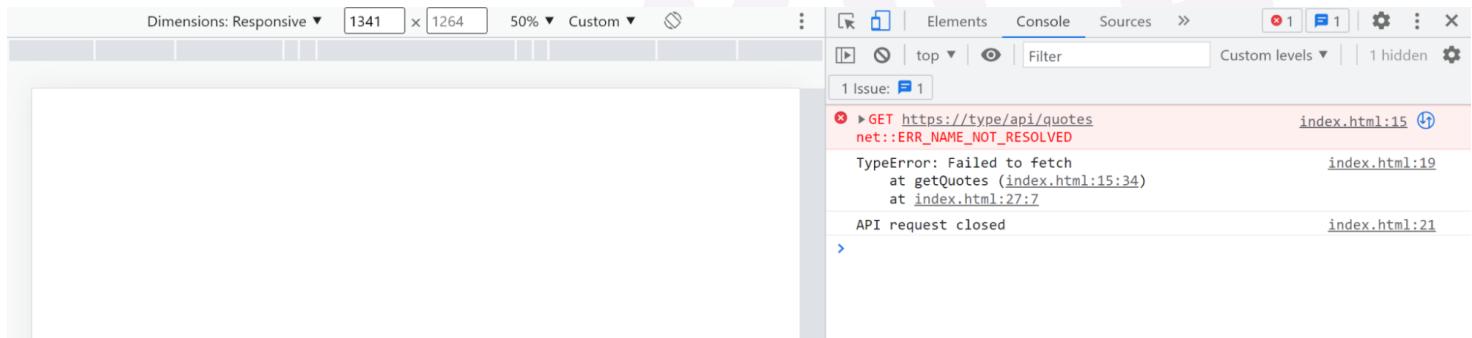
```
// Calling the function
getQuotes();
</script>

</body>
</html>
```

- Returns data if it is a valid request.



- When errors are found, the errors are logged to the console.



The above code uses `async/await` syntax in JavaScript, with added error handling using the `try/catch` block and a `finally` block to run code after the `try/catch` block regardless of whether an error occurred or not.

The `getQuotes()` function starts with a `try` block, within which the `fetch()` method is used to make a request to the API, and the response is saved to the `response` variable using the `await` keyword. Then, the response is parsed as JSON using the `response.json()` method and saved to the `data` variable using the `await` keyword. Finally, the resulting data is logged to the console using `console.log()`.

If an error occurs during this process, it is caught by the `catch` block, and the error message is logged to the console using `console.log(error)`. The `finally` block is then executed, and a message is logged to the console indicating that the API request is closed.