upGrad

#RahoAmbitious

Full Stack Software Development

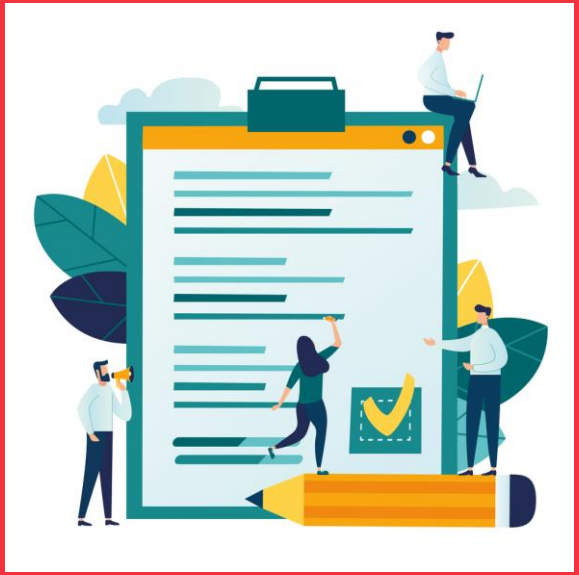
upGrad

Lecture On: AJAX and Backend Integration

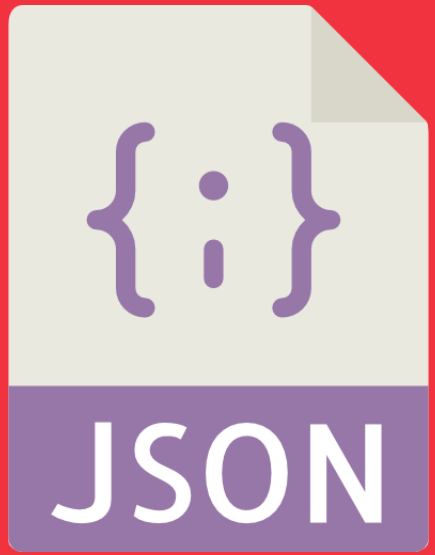
Instructor: Aquib Ajani

Today's Agenda

- Introduction to JSON and how to parse JSON files
- Introduction to AJAX
- Introduction to HTTP Requests and Responses
- Differences between GET, POST, PUT and DELETE
- HTTP Headers
- The XMLHTTPRequest



Introduction to JSON



Introduction to JSON

upGrad

- **JSON** stands for **JavaScript Object Notation**.
- To transfer data between servers and browsers, the data needs to be in text format only.
- JSON is a text format, which can be converted to object notation; hence, it is called JavaScript Object Notation.
- Therefore, JSON is used for storing and exchanging data. It can be stored as JavaScript (JS) objects. JSON can be converted to text while being sent across the network, and they can be converted back to objects when they are received.

```
{
"name": "John Doe",
"age": 27,
"nationality": "English"
}
```

JSON Data Types

Data Type	Description
number	This data type is used for all numeric types, be it integers or floating-point numbers.
string	This data type is used for a string of Unicode characters. You need to enclose the value in double-quotes.
boolean	This data type is used for a true/false value.
array	This data type contains an ordered list of 0 or more values. The values are wrapped in square brackets.
object	This is an unordered collection of key-value pairs. You can have multiple objects embedded in a JSON object.
null	This data type is used for null values.

JSON Parsing and Converting JSON to Strings

```
let myJson = '{"name": "John Doe", "age": 27, "nationality": "English", "cars": ["BMW", "Audi"]}'
let obj1 = JSON.parse(myJson);
console.log(obj1); //{name: "John Doe", age: 27, nationality: "English", cars: Array(2)}
let obj2 = {
   name: “John Doe”,
   age: 27
}
let jsonText = JSON.stringify(obj2);
console.log(jsonText); //{"name":"John Doe","age":27}
```

- JSON objects received from servers are in text or string format. To convert them to object format, so you can access and use the key-value pair inside the object, you can use **JSON.parse()**.
- Similarly, if you want to convert JSON objects to string format, so you can send them across the server, you can use the **JSON.stringify()** method.

Asynchronous Programming

Introduction to AJAX

Introduction to AJAX

upGrad

- **AJAX** stands for Asynchronous JavaScript and XML. It is not a language, but a process to interact with the server and update the elements on the DOM.
- Updation of UI is not done directly by AJAX. AJAX is also useful for sending and receiving data from the server.
- AJAX allows web pages to be updated asynchronously by exchanging data with the web server in the background. This essentially ensures that only a part of the web page is loaded without reloading the entire page.
- AJAX uses a combination of XMLHttpRequest object along with JavaScript and DOM.

XMLHttpRequest

The **XMLHttpRequest** object can be used to exchange data with the server in the background. However, for security reasons, modern browsers do not allow access across domains(This is called CORS). This means both the website and the data need to be on the same server.

```
let xhhtp = new XMLHttpRequest();
```

To send a request to the server, we use the open() and send() methods of the XMLHttpRequest object.

- The **open()** method specifies the type of request. It takes three parameters – *method*, *url* and *async*. The *method* parameter specifies the type of request – GET or POST. The *url* parameter specifies the server (file) location, and ***async***(*this is a boolean value and true corresponds to asynchronous request and false corresponds to synchronous request*).

```
xhhtp.open("GET", "https://tripadvisor1.p.rapidapi.com/hotels/get-details?lang=en_US&location_id=12345", true);
```

- The **send()** method is used for sending the GET request to the server, and the variation **send(*string*)** method is used for sending the POST request.

```
xhhtp.send();
```

Introduction to AJAX

upGrad

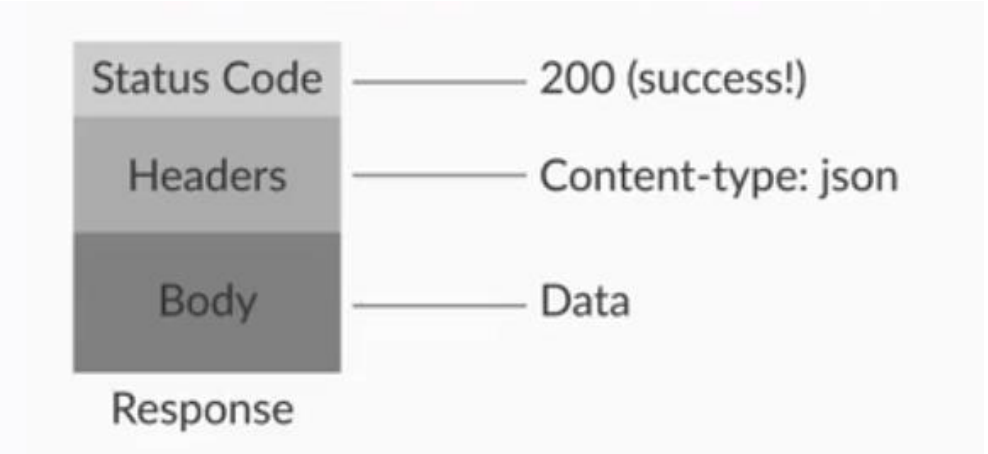
AJAX Response

- The **readyState** property returns the current state of the XMLHttpRequest.

  The readyState property has five values – **0** (The XHR client has been created, but the open() method hasn't been called yet.), **1** (open() has been called.), **2** (send() has been called, and headers and status are available.), **3** (Response's body is being received.) and **4** (The fetch operation is complete).

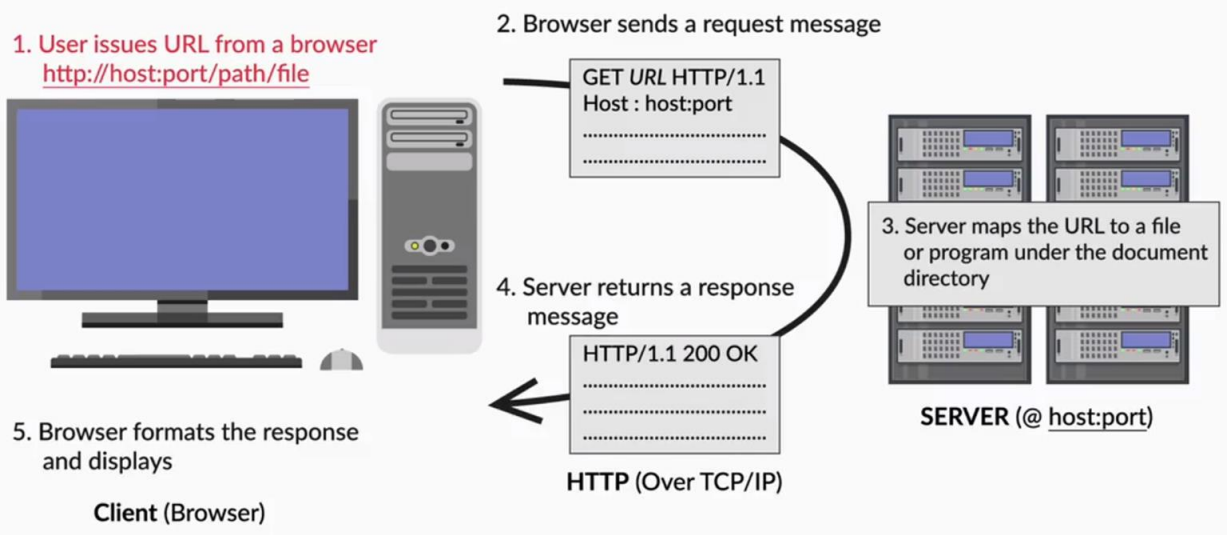
- The **onreadystatechange** property defines a callback function to be executed when the readyState changes.

- The **status** property shows the status of a response. A few important response codes include **200** (OK), **401** (Unauthorized), **403** (Forbidden) and **404** (Not found).




```
let loadDoc = () => {
     let xhhtp = new XMLHttpRequest();
     xhhtp.open("GET",
"https://www.googleapis.com/books/v1/volumes?q=harry", true);
     xhhtp.send();
     xhhtp.onreadystatechange = () => {
         if (xhhtp.readyState === 4 && xhhtp.status === 200)
alert("Here");
     };
}
```

AJAX Response



upGrad

GET, POST, PUT and DELETE Requests

GET, POST, PUT and DELETE

- **GET** – The GET method is used to retrieve data from the server. Since it is a read-only method, there is no risk of the data getting mutated or corrupted.

```
xhr.open('GET', url);
xhr.send();
```

- **POST** – The POST method sends data to the server, and it creates a new resource. This method is essentially used to create new data entries.

```
xhr.open('POST', url);
xhr.send(dataToSend); // dataToSend is the data to be sent to the server
```

GET, POST, PUT and DELETE

- **PUT** – The PUT method is used to update an existing resource. Whenever you need to update data, you can call this resource along with the new data that needs to be updated.

```
xhr.open('PUT', url);
xhr.send(dataToSend); // dataToSend is the data to be sent to the server
```

- **DELETE** – The DELETE method is used to delete a specific resource.

```
xhr.open('DELETE', url);
xhr.send(dataToSend); // dataToSend is the data to be sent to the server
```

upGrad

HTTP Headers

HTTP Headers

upGrad

- **HTTP headers** allow the client and the server to send additional information with an HTTP request or response.

- An HTTP header consists of its case-insensitive name followed by a colon and then its value.

  HTTP headers are of the following four types:

- **General Header:** It is an HTTP header that can be used in both request and response messages, but it is not related to the data sent with the request. Example: Connection header

- **Request Header:** It is a header that can be used in an HTTP request, but it is not related to the body of the request. A few request headers include Host, User-Agent, Accept, Cache-Control, Accept-Language, etc.

- **Response Header:** It is a header that can be used in an HTTP response, but it is not related to the content itself. A few response headers include Access-Control-Allow-Origin, Date, Keep-Alive, Server, Set-Cookie, etc.

- **Entity Header:** It is a header that describes the body of the resource. A few entity headers include Content-Length, Content-Encoding, Content-Language, etc. It can be used in both request and response headers.

Hands-On Exercise (10 mins)

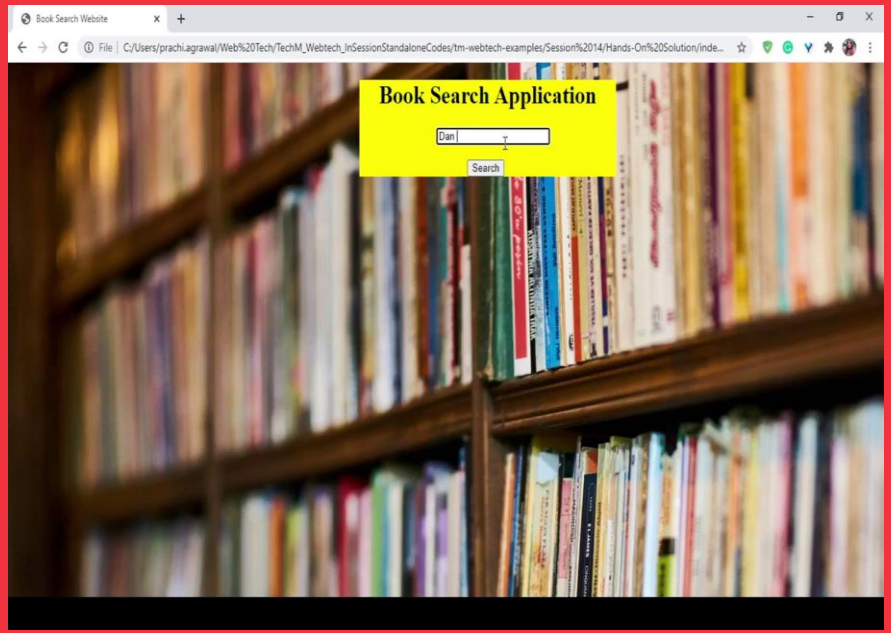
upGrad

Write a JavaScript program that uses AJAX calls to fetch the titles of books from a given title or author string in the search bar. For this purpose, you are required to use the Google Books API.

Use 'https://www.googleapis.com/books/v1/volumes?q=' and concatenate the title or the author string used to search the books to the given URL in your JavaScript code.

The required HTML and CSS files are provided to you as a stub code.

Note: The website should dynamically list down the book titles from the given search.



Key Takeaways

- JSON (JavaScript Object Notation) is used for storing and sending data across a network.
- JSON is a text format, but it can be converted to object notation using *JSON.parse()*.
- Similarly, to convert a JavaScript (JS) object to JSON text format, you can use *JSON. stringify()*.
- AJAX (Asynchronous JavaScript and XML) is used to either update only specific parts of a web page without reloading the entire page, or send and receive data between servers and clients.
- GET is a read-only method that is used for getting data. POST is used for adding a new entry. PUT and PATCH are used for updating data, while DELETE is used for deleting entries.
- To fix errors in code, we use debuggers.

upGrad

#RahoAmbitious

Thank You!