2.8 AJAX Calls

This section will guide you to:

* Create a JavaScript project in your IDE
* Write a program in JavaScript to work with AJAX calls, fetch, and promise

This lab has three subsections, namely:

2.8.1 Writing a program in JavaScript to verify implementations of AJAX, fetch, and promise

2.8.2 Executing the program and verifying working of AJAX calls, fetch, and promise

2.8.3 Pushing the code to your GitHub repositories

**Step 2.8.1:** Writing a program in JavaScript to verify implementations of AJAX, fetch, and promise

* Open Visual Studio Code
* *[Right click]* on the **src** folder of the project
* Select *New File* -> Enter the filename as **ajax\_demo1.html**
* Write the code shown below resolving the warning and errors due compatibility-related issues

<html>

<body>

<header>

<h1>MEAN Stack</h1>

<p> Lesson 2 Demos </p>

<p> Ajax without Fetch and Promise</p>

</header>

<script language="javascript" type="text/javascript">

function ajax\_call\_demo( ){

        if (window.XMLHttpRequest || window.ActiveXObject) {

         if (window.ActiveXObject) {

             try {

                 xhttp = new ActiveXObject("Msxml2.XMLHTTP");

             } catch(exception) {

                 xhttp = new ActiveXObject("Microsoft.XMLHTTP");

             }

         } else {

xhttp = new XMLHttpRequest();

         }

        } else {

         alert("Your browser does not support XMLHttpRequest...!");

        }

xhttp.open("GET", "https://images.pexels.com/photos/853168/pexels-photo-853168.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500", true);  // Make sure file is in same server

xhttp.overrideMimeType('text/plain; charset=x-user-defined');

        xhttp.send(null);

xhttp.onreadystatechange = function() {

         if (xhttp.readyState == 4){

         if ((xhttp.status == 200) || (xhttpr.status == 0)){

         var image = document.getElementById("get\_img");

         image.src = "data:image/gif;base64," + encode64(xhttp.responseText);

         }else{

         alert("Something misconfiguration : " +

             "\nError Code : " + xhttp.status +

             "\nError Message : " + xhttp.responseText);

         }

         }

        };

}

function encode64(inputStr){

var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";

var outputStr = "";

var i = 0;

while (i<inputStr.length){

var byte1 = inputStr.charCodeAt(i++) & 0xff;

var byte2 = inputStr.charCodeAt(i++) & 0xff;

var byte3 = inputStr.charCodeAt(i++) & 0xff;

var enc1 = byte1 >> 2;

var enc2 = ((byte1 & 3) << 4) | (byte2 >> 4);

var enc3, enc4;

if (isNaN(byte2)){

            enc3 = enc4 = 64;

} else{

enc3 = ((byte2 & 15) << 2) | (byte3 >> 6);

if (isNaN(byte3)){

enc4 = 64;

} else {

enc4 = byte3 & 63;

}

}

outputStr += b64.charAt(enc1) + b64.charAt(enc2) + b64.charAt(enc3) + b64.charAt(enc4);

}

return outputStr;

}

</script>

<button onClick="ajax\_call\_demo()">Click here to get an image</button><br />

<img id="get\_img" />

</body>

</html>

* *[Right click]* on the **src** folder of the project
* Select *New File* -> Enter the filename as **index.html**
* Write the code shown below resolving the warning and errors due compatibility-related issues

<!DOCTYPE html>

<html lang="en">

<body>

<h1>MEAN Stack</h1>

<p> Lesson 2 Demos </p>

<div id="main"></div>

<script src="index.js"></script>

</body>

</html>

* *[Right click]* on the **src** folder of the project
* Select *New File* -> Enter the filename as **index.js**
* Write the code shown below resolving the warning and errors due compatibility-related issues

const URL = "https://ghibliapi.herokuapp.com/people";

const main = document.getElementById("main");

main.innerHTML = "<p>Loading...";

fetch(URL)

.then((response) => response.json())

.then((people) => main.innerHTML = getListOfNames(people));

const getListOfNames = (people) => {

const names = people

.map((person) => `<li>${person.name}</li>`)

.join("\n");

return `<ul>${names}</ul>`;

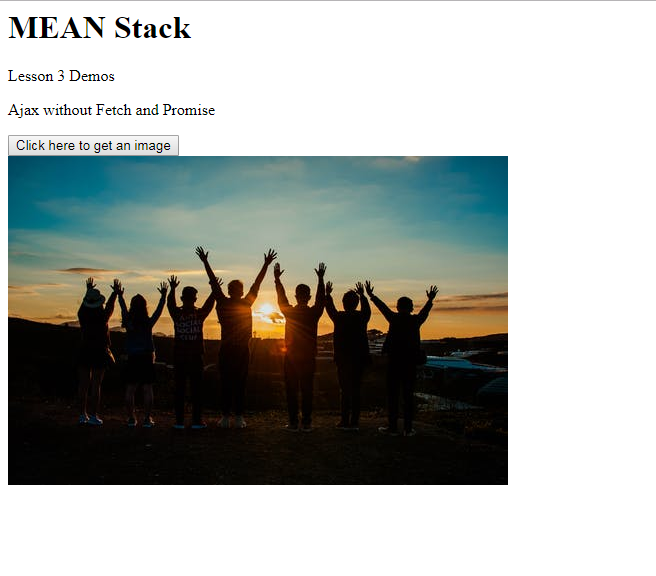
};

* In this demo, *json()* method actually returns a promise

**Step 2.8.2:** Executing the program and verifying working of AJAX calls, fetch, and promise

Before you execute the program, check for syntactical corrections. If no errors are found, follow the steps mentioned below:

* Go to Extensions and download **Live Server**
* *[Right click]* on the **ajax\_demo1.html** file of the project
* Select *Open with Live Server*
* Right click when the server starts running. Select *Inspect Element*.Click on **Console**



* *[Right click]* on the **index.html** file of the project
* Select *Open with Live Server*
* Right click when the server starts running. Select *Inspect*.Click on **Console**



**Step 2.8.3:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master