



Project 6: PostMan Clone - Creating a Get/Post Request Website | JavaScript Tutorial In Hindi #63

[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Project 6: PostMan Clone - Creating a Get/Post Request Website | JavaScript Tutorial In Hindi #63

Now it's time for the 6th project. The project we are doing to do is creating a "PostMan Clone". The project details are given below:

Project Explanation:-

Postman is a popular application programming interface that makes it easy for developers to create, share, test and document APIs. This is done by allowing the users to create and save simple and complex HTTP/s requests, as well as read their responses. It can make various types of HTTP requests like GET, POST, PUT, PATCH, saving environments for later use, converting the **API** to code for various languages like JavaScript, Python.

The website we are going to build in this tutorial looks like the following:

Setting Up the Project:-

For this project, we will use the VS Code. If you have not installed the Visual Studio Code yet, then click on the link below for installation guidance:

<https://codewithharry.com/videos/javascript-tutorials-in-hindi-1>

For project setup, start by creating a new project folder in VS Code, and inside that project folder, create two empty new files of HTML (filename.html) and JavaScript (filename.js). The *Html file* is the entry point for our website and contains the HTML code. In this project, we are also using **Bootstrap**.

Bootstrap:-

Bootstrap is used for applying styling the user interface components. The easiest way to include Bootstrap is to add it from <https://getbootstrap.com/docs/4.0/getting-started/introduction/>.

First add the following in the head section:

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
      integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

Then include the following code in the body section, before the closing </body> tag:

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
      crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
      integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1c1HTMga3JDZwrnQq4sF86dIHNDz0W1"
      crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
      integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
      crossorigin="anonymous"></script>
```

Bootstrap is bundled with many components that can be used to provide good user experience and user interactions in a web page. In this project, we are using bootstrap components for making the front end of website. So, let's start coding!!!

Code as described/written in the video

```
<!doctype html>
<html lang="en">
```

```
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <link rel="stylesheet" href="prism.css">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

  <style>
    #responsePre{
      max-height: 500px;
    }
  </style>
  <title>PostMaster by CodeWithHarry!</title>
</head>

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a class="navbar-brand" href="#">PostMaster</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText"
      aria-controls="navbarText" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarText">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Contact us</a>
        </li>
      </ul>
    </div>
  </nav>
```

```
</div>
</nav>

<div class="container">
  <h1 class="my-3">Welcome to PostMaster</h1>
</div>

<div class="container">

  <!-- URL box -->
  <div class="form-group row">
    <label for="url" class="col-sm-2 col-form-label">URL</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" id="url" placeholder="Enter URL here">
    </div>
  </div>

  <!-- Request type box -->
  <fieldset class="form-group">
    <div class="row">
      <legend class="col-form-label col-sm-2 pt-0">Request Type</legend>
      <div class="col-sm-10">
        <div class="form-check">
          <input class="form-check-input" type="radio" name="requestType" id="get" value="GET" checked>
          <label class="form-check-label" for="get">
            GET
          </label>
        </div>
        <div class="form-check">
          <input class="form-check-input" type="radio" name="requestType" id="post" value="POST">
          <label class="form-check-label" for="post">
            POST
          </label>
        </div>
      </div>
    </div>
  </fieldset>
```

```
<!-- Content type box -->
<fieldset class="form-group">
  <div class="row">
    <legend class="col-form-label col-sm-2 pt-0">Content Type</legend>
    <div class="col-sm-10">
      <div class="form-check">
        <input class="form-check-input" type="radio" name="contentType" id="jsonRadio" value="json"
          checked>
        <label class="form-check-label" for="json">
          JSON
        </label>
      </div>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="contentType" id="paramsRadio" value="params">
        <label class="form-check-label" for="params">
          Custom Parameters
        </label>
      </div>
    </div>
  </div>
</fieldset>

<!-- Parameters box - This will hide on clicking json option in content type -->
<div id="parametersBox">
  <div class="form-row">
    <label for="url" class="col-sm-2 col-form-label">Parameter 1</label>
    <div class="col-md-4">
      <input type="text" class="form-control" id="parameterKey1" placeholder="Enter Parameter 1 Key">
    </div>
    <div class="col-md-4">
      <input type="text" class="form-control" id="parameterValue1" placeholder="Enter Parameter 1 Value">
    </div>
    <button id="addParam" class="btn btn-primary">+</button>
  </div>
  <div id="params"></div>
</div>
```

```
<!-- Json Request box - This will hide on clicking parameters option in content type -->
<div class="my-3" id="requestJsonBox">
  <div class="form-group row">
    <label for="requestJsonText" class="col-sm-2 col-form-label">Enter Request Json</label>
    <div class="col-sm-10">
      <textarea class="form-control" id="requestJsonText" rows="3"></textarea>
    </div>
  </div>
</div>

<!-- Submit button which will trigger fetch api -->
<div class="form-group row my-2">
  <div class="col-sm-10">
    <button id="submit" class="btn btn-primary">Submit</button>
  </div>
</div>

<div class="my-3" id="responseJsonBox">
  <div class="form-group row">
    <label for="responseJsonText" class="col-sm-2 col-form-label">Response</label>
    <div class="col-sm-10">
      <!-- <textarea class="form-control" id="responseJsonText" rows="3">Your response will appear here</textarea> -

      <pre id='responsePre' class="language-javascript"> <code id='responsePrism' class="language-javascript"> Your
    </div>
  </div>
</div>

</div>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
  integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVZpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
```

```
    crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
    integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01c1HTMga3JDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
    integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
    crossorigin="anonymous"></script>

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

</html>
```

Code as described/written in the video

```
console.log('This is my project 6 from JavaScript course');

// Utility functions:
// 1. Utility function to get DOM element from string
function getElementFromString(string) {
    let div = document.createElement('div');
    div.innerHTML = string;
    return div.firstChild;
}

// Initialize no of parameters
let addedParamCount = 0;

// Hide the parameters box initially
let parametersBox = document.getElementById('parametersBox');
parametersBox.style.display = 'none';

// If the user clicks on params box, hide the json box
let paramsRadio = document.getElementById('paramsRadio');
paramsRadio.addEventListener('click', () => {
    document.getElementById('requestJsonBox').style.display = 'none';
    document.getElementById('parametersBox').style.display = 'block';
});
```

Copy

```
    })

    // If the user clicks on json box, hide the params box
    let jsonRadio = document.getElementById('jsonRadio');
    jsonRadio.addEventListener('click', () => {
        document.getElementById('requestJsonBox').style.display = 'block';
        document.getElementById('parametersBox').style.display = 'none';
    })

    // If the user clicks on + button, add more parameters
    let addParam = document.getElementById('addParam');
    addParam.addEventListener('click', () => {
        let params = document.getElementById('params');
        let string = `<div class="form-row my-2">
            <label for="url" class="col-sm-2 col-form-label">Parameter ${addedParamCount + 2}</label>
            <div class="col-md-4">
                <input type="text" class="form-control" id="parameterKey${addedParamCount + 2}" placeholder="Enter Parameter $
            </div>
            <div class="col-md-4">
                <input type="text" class="form-control" id="parameterValue${addedParamCount + 2}" placeholder="Enter Parameter
            </div>
            <button class="btn btn-primary deleteParam"> - </button>
        </div>`;

        // Convert the element string to DOM node
        let paramElement = getElementFromString(string);
        params.appendChild(paramElement);
        // Add an event listener to remove the parameter on clicking - button
        let deleteParam = document.getElementsByClassName('deleteParam');
        for (item of deleteParam) {
            item.addEventListener('click', (e) => {
                // TODO: add a confirmation box to confirm parameter deletion
                e.target.parentElement.remove();
            })
        }
        addedParamCount++;
    })

    // If the user clicks on submit button
```



```
let submit = document.getElementById('submit');
submit.addEventListener('click', () => {
  // Show please wait in the response box to request patience from the user
  // document.getElementById('responseJsonText').value = "Please wait.. Fetching response...";
  document.getElementById('responsePrism').innerHTML = "Please wait.. Fetching response...";

  // Fetch all the values user has entered
  let url = document.getElementById("url").value;
  let requestType = document.querySelector("input[name='requestType']:checked").value;
  let contentType = document.querySelector("input[name='contentType']:checked").value;

  // If user has used params option instead of json, collect all the parameters in an object
  if (contentType == 'params') {
    data = {};
    for (let i = 0; i < addedParamCount + 1; i++) {
      if (document.getElementById('parameterKey' + (i + 1)) != undefined) {
        let key = document.getElementById('parameterKey' + (i + 1)).value;
        let value = document.getElementById('parameterValue' + (i + 1)).value;
        data[key] = value;
      }
    }
    data = JSON.stringify(data);
  }
  else {
    data = document.getElementById('requestJsonText').value;
  }

  // Log all the values in the console for debugging
  console.log('URL is ', url);
  console.log('requestType is ', requestType);
  console.log('contentType is ', contentType);
  console.log('data is ', data);

  // if the request type is get, invoke fetch api to create a post request
  if (requestType=='GET'){
    fetch(url, {
      method: 'GET',
    })
```

```
        .then(response=> response.text())
        .then((text) =>{
            // document.getElementById('responseJsonText').value = text;
            document.getElementById('responsePrism').innerHTML = text;
            Prism.highlightAll();
        });
    }

    else{
        fetch(url, {
            method: 'POST',
            body: data,
            headers: {
                "Content-type": "application/json; charset=UTF-8"
            }
        })
        .then(response=> response.text())
        .then((text) =>{
            // document.getElementById('responseJsonText').value = text;
            document.getElementById('responsePrism').innerHTML = text;
            Prism.highlightAll();
        });
    }

});
```

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

