



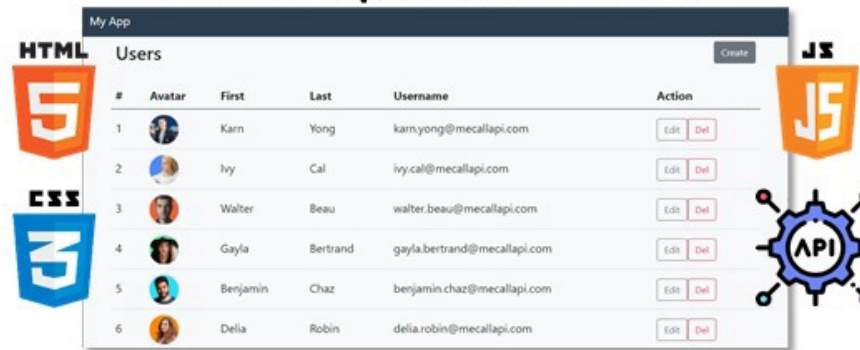
Karn Yongsiriwit

[Follow](#)

Jun 12, 2021 · 4 min read · [Listen](#)


# Let's Build a CRUD Website with HTML, CSS, JavaScript and an External API

## Basic HTML, CSS, JS (Bootstrap 5) for CRUD operations with API



In this article, we will give you a tutorial for creating a web application with just only basic **HTML**, **CSS** and **JavaScript** (based on **Bootstrap 5**) to perform **CRUD** operations. Well, **CRUD** operations are the four basic operations of manipulating data including Create/Construct, Read, Update and Delete. Furthermore, our **CRUD** operations will perform by the use of an external **API** from [MeCallAPI.com](#). If you want to try a mockup API for CRUD and authentication operations, feel free to check on the website.



## ME CALL API

Free REST-API with fake data, Ready to be called for simulating the real-world application. ME CALL API is suitable for developers who want to quickly build and test prototypes.

### TRY CALLING API!

```
USER LIST: api/users

Method: GET
URL: https://www.mecallapi.com/api/users
Response (200):
[
  {
    "id": 1,
    "fname": "karn",
    "lname": "rong",
    "username": "karn.yong@mecallapi.com",
    "avatar": "https://www.mecallapi.com/users/1.png"
  },
  {
    "id": 2,
    "fname": "xyy",
```

<https://www.mecallapi.com/>

You can also check an example of what we want to achieve in this article here:

<https://www.mecallapi.com/crud/>





Users						Create
#	Avatar	First	Last	Username	Action	
1		Karn	Yong	karn.yong@mecallapi.com	Edit	Del
2		Ivy	Cal	ivy.cal@mecallapi.com	Edit	Del
3		Walter	Beau	walter.beau@mecallapi.com	Edit	Del
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	Edit	Del
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	Edit	Del
6		Delia	Robin	delia.robin@mecallapi.com	Edit	Del
7		Hector	Graves	hector.graves@mecallapi.com	Edit	Del
8		Diego	Greene	diego.greene@mecallapi.com	Edit	Del
9		Izzie	Mooney	izzie.mooney@mecallapi.com	Edit	Del
10		Romeo	Garner	romeo.garner@mecallapi.com	Edit	Del

<https://www.mecallapi.com/crud/>

## Software Installation

You only need a Text Editor/IDE (VS Code, Notepad, etc.) and a web browser (Chrome, Firefox, Edge, etc.) to do this tutorial!

## Let's Code! (HTML and CSS)

Starting for the HTML which are documents designed to be displayed in a web browser. There are more than a hundred of HTML elements you can choose to put on your HTML file and make the magic happen.





- **Sweetalert** for easily creating nice popups using JavaScript (line 53)
- **index.css** for defining extra CSS (Cascading Style Sheets) to style your **index.html** in addition from the Bootstrap (line 13)
- **index.js** for JavaScript using in **index.html** to call API for **CRUD** operations (line 52)

```
1  <!doctype html>
2  <html lang="en">
3    <head>
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7
8      <!-- Bootstrap CSS -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="styl
10
11      <title>CRUD FROM API</title>
12
13      <link href="index.css" rel="stylesheet">
14    </head>
15    <body>
16
17      <nav class="navbar navbar-dark bg-mynav">
18        <div class="container-fluid">
19          <a class="navbar-brand" href="#">My App</a>
20        </div>
21      </nav>
22
23      <div class="container">
24        <div class="d-flex bd-highlight mb-3">
25          <div class="me-auto p-2 bd-highlight"><h2>Users</div>
26          <div class="p-2 bd-highlight">
27            <button type="button" class="btn btn-secondary" onclick="showUserCreateBox()">Create</bu
28          </div>
```





```
34         <tr>
35             <th scope="col">#</th>
36             <th scope="col">Avatar</th>
37             <th scope="col">First</th>
38             <th scope="col">Last</th>
39             <th scope="col">Username</th>
40             <th scope="col">Action</th>
41         </tr>
42     </thead>
43     <tbody id="mytable">
44         <tr>
45             <th scope="row" colspan="5">Loading...</th>
46         </tr>
47     </tbody>
48 </table>
49 </div>
50 </div>
51
52 <script src="index.js"></script>
53 <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11.0.16/dist/sweetalert2.all.min.js"></s
54 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" int
55 </body>
56 </html>
```

index.html hosted with ❤ by GitHub

[view raw](#)

Create **index.css** to define extra CSS (Cascading Style Sheets) for your **index.html** in addition from the CSS provided by Bootstrap.

```
1 .bg-mynav {
2     background-color: #2c3e50;
3 }
4
5 body {
6     font-size: 1.25rem;
```





12 }

index.css hosted with ❤ by GitHub

[view raw](#)

## Read operation (JavaScript)

Create file **index.js** to call an **API** for **CRUD** operations starting from **Read**.

API URL: <https://www.mecallapi.com/api/users>

Method: GET

To read data with JavaScript, we will use **AJAX (Asynchronous JavaScript And XML)** technique to have our web site update asynchronously. Concretely, we will have data exchange using **API** in the background. Thus, the data is retrieved, we will have our web page updated without doing a refresh.

We will use **XMLHttpRequest** to call an **API** for retrieving data in **JSON** and display/update such data on the web page. To update the data on the web page, we will manipulate **HTML DOM (Document Object Model)**. In our code, we will update the data in table by referring to the id of the **HTML** data table element, namely (**mytable**) in line 21.

```
1 function loadTable() {
2   const xhttp = new XMLHttpRequest();
3   xhttp.open("GET", "https://www.mecallapi.com/api/users");
4   xhttp.send();
5   xhttp.onreadystatechange = function() {
6     if (this.readyState == 4 && this.status == 200) {
7       console.log(this.responseText);
8       var trHTML = '';
```





```
14     trHTML += '<td>' + object['fname'] + '</td>';
15     trHTML += '<td>' + object['lname'] + '</td>';
16     trHTML += '<td>' + object['username'] + '</td>';
17     trHTML += '<td><button type="button" class="btn btn-outline-secondary" onclick="showUserEd
18     trHTML += '<button type="button" class="btn btn-outline-danger" onclick="userDelete(' + obje
19     trHTML += "</tr>";
20 }
21 document.getElementById("mytable").innerHTML = trHTML;
22 }
23 };
24 }
25
26 loadTable();
```

index.js hosted with ❤ by GitHub

[view raw](#)

Open **index.html** on a web Browser, you will see the result:

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<button>Edit</button> <button>Del</button>
6		Delia	Robin	delia.robin@mecallapi.com	<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>





URL: <https://www.mecallapi.com/api/users/create>

Method: POST

Sample body (JSON):

```
{
  "fname": "Cat",
  "lname": "Chat",
  "username": "cat.chat@mecallapi.com",
  "email": "cat.chat@mecallapi.com",
  "avatar": "https://www.mecallapi.com/users/cat.png"
}
```

Add the following code in **index.js**

```
1  function showUserCreateBox() {
2    Swal.fire({
3      title: 'Create user',
4      html:
5        '<input id="id" type="hidden">' +
6        '<input id="fname" class="swal2-input" placeholder="First">' +
7        '<input id="lname" class="swal2-input" placeholder="Last">' +
8        '<input id="username" class="swal2-input" placeholder="Username">' +
9        '<input id="email" class="swal2-input" placeholder="Email">',
10     focusConfirm: false,
11     preConfirm: () => {
12       userCreate();
13     }
14   })
15 }
16
17 function userCreate() {
18   const fname = document.getElementById("fname").value;
19   const lname = document.getElementById("lname").value;
20   const username = document.getElementById("username").value;
```







```
26 xhttp.send(JSON.stringify({
27   "fname": fname, "lname": lname, "username": username, "email": email,
28   "avatar": "https://www.mecallapi.com/users/cat.png"
29 }));
30 xhttp.onreadystatechange = function() {
31   if (this.readyState == 4 && this.status == 200) {
32     const objects = JSON.parse(this.responseText);
33     Swal.fire(objects['message']);
34     loadTable();
35   }
36 };
37 }
```

index.js hosted with ❤ by GitHub

[view raw](#)

Refresh **index.html** and try to perform the create operation:

The screenshot shows a web application titled "My App" with a "Users" section. The "Users" section contains a table with 10 rows of user data. Each row has columns for "#", "Avatar", "First", "Last", "Username", and "Action". The "Action" column contains "Edit" and "Del" buttons. A "Create" button is located in the top right corner of the "Users" section, highlighted with a red box.

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
2		Ivy	Cal	ivy.cal@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
3		Walter	Beau	walter.beau@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
6		Delia	Robin	delia.robin@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
7		Hector	Graves	hector.graves@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
8		Diego	Greene	diego.greene@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>
10		Romeo	Garner	romeo.garner@mecallapi.com	<a href="#">Edit</a> <a href="#">Del</a>

Click Create





Users					
#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy			<button>Edit</button> <button>Del</button>
3		Walter			<button>Edit</button> <button>Del</button>
4		Gayla			<button>Edit</button> <button>Del</button>
5		Benjamin			<button>Edit</button> <button>Del</button>
6		Delia			<button>Edit</button> <button>Del</button>
7		Hector			<button>Edit</button> <button>Del</button>
8		Diego			<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>

Create user

  
  
  
  
OK

Input data

CRUD FROM API

File | C:/medium-project/bootstrap-app/index.html

My App

Create

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<button>Edit</button> <button>Del</button>
6		Delia	Robin	delia.robin@mecallapi.com	<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>
11		Cat	Chat	cat.chat@mecallapi.com	<button>Edit</button> <button>Del</button>

New data is added





URL: <https://www.mecallapi.com/api/users/update>

Method: PUT

Sample body (JSON):

```
{
  "id": 11,
  "fname": "Cat",
  "lname": "Gato",
  "username": "cat.gato@mecallapi.com",
  "email": "cat.gato@mecallapi.com",
  "avatar": "https://www.mecallapi.com/users/cat.png"
}
```

Add the following code in **index.js**

```
1 function showUserEditBox(id) {
2   console.log(id);
3   const xhttp = new XMLHttpRequest();
4   xhttp.open("GET", "https://www.mecallapi.com/api/users/"+id);
5   xhttp.send();
6   xhttp.onreadystatechange = function() {
7     if (this.readyState == 4 && this.status == 200) {
8       const objects = JSON.parse(this.responseText);
9       const user = objects['user'];
10      console.log(user);
11      Swal.fire({
12        title: 'Edit User',
13        html:
14          '<input id="id" type="hidden" value='+user['id']+>' +
15          '<input id="fname" class="swal2-input" placeholder="First" value="'+user['fname']+'">' +
16          '<input id="lname" class="swal2-input" placeholder="Last" value="'+user['lname']+'">' +
17          '<input id="username" class="swal2-input" placeholder="Username" value="'+user['username']+'">' +
18          '<input id="email" class="swal2-input" placeholder="Email" value="'+user['email']+'">',
19        focusConfirm: false,
```





```
25     };
26   }
27
28   function userEdit() {
29     const id = document.getElementById("id").value;
30     const fname = document.getElementById("fname").value;
31     const lname = document.getElementById("lname").value;
32     const username = document.getElementById("username").value;
33     const email = document.getElementById("email").value;
34
35     const xhttp = new XMLHttpRequest();
36     xhttp.open("PUT", "https://www.mecallapi.com/api/users/update");
37     xhttp.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
38     xhttp.send(JSON.stringify({
39       "id": id, "fname": fname, "lname": lname, "username": username, "email": email,
40       "avatar": "https://www.mecallapi.com/users/cat.png"
41     }));
42     xhttp.onreadystatechange = function() {
43       if (this.readyState == 4 && this.status == 200) {
44         const objects = JSON.parse(this.responseText);
45         Swal.fire(objects['message']);
46         loadTable();
47       }
48     };
49   }
```

index.js hosted with ❤ by GitHub

[view raw](#)

Refresh **index.html** and try to perform the update operation:





Users					
#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<button>Edit</button> <button>Del</button>
6		Delia	Robin	delia.robin@mecallapi.com	<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>
11		Cat	Chat	cat.chat@mecallapi.com	<button>Edit</button> <button>Del</button>

Edit the newly added user

CRUD FROM API

My App

Users Create

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy			<button>Edit</button> <button>Del</button>
3		Walter			<button>Edit</button> <button>Del</button>
4		Gayla			<button>Edit</button> <button>Del</button>
5		Benjamin			<button>Edit</button> <button>Del</button>
6		Delia			<button>Edit</button> <button>Del</button>
7		Hector			<button>Edit</button> <button>Del</button>
8		Diego			<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>
11		Cat	Chat	cat.chat@mecallapi.com	<button>Edit</button> <button>Del</button>

Edit User

OK

Input data





Users					
#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<button>Edit</button> <button>Del</button>
6		Delia	Robin	delia.robin@mecallapi.com	<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>
11		Cat	Gato	cat.gato@mecallapi.com	<button>Edit</button> <button>Del</button>

Data is updated

## DELETE operation (JavaScript)

API URL: <https://www.mecallapi.com/api/users/delete>

Method: DELETE

Sample Body (JSON):

```
{
  "id": 11
}
```

Add the following code in **index.js**

```
1 function userDelete(id) {
```





```
7   });
8   xhttp.onreadystatechange = function() {
9     if (this.readyState == 4) {
10      const objects = JSON.parse(this.responseText);
11      Swal.fire(objects['message']);
12      loadTable();
13    }
14  };
15 }
```

index.js hosted with ❤ by GitHub

[view raw](#)

Refresh **index.html** and try to perform the delete operation:

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<button>Edit</button> <button>Del</button>
6		Delia	Robin	delia.robin@mecallapi.com	<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>
11		Cat	Gato	cat.gato@mecallapi.com	<button>Edit</button> <button>Del</button>

Delete the newly added user

You can now subscribe to get stories delivered directly to your inbox.

**Got it**





Users

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin			<button>Edit</button> <button>Del</button>
6		Delia			<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>

User with ID = 11 is deleted

OK

Popup is showing

My App

Users Create

#	Avatar	First	Last	Username	Action
1		Karn	Yong	karn.yong@mecallapi.com	<button>Edit</button> <button>Del</button>
2		Ivy	Cal	ivy.cal@mecallapi.com	<button>Edit</button> <button>Del</button>
3		Walter	Beau	walter.beau@mecallapi.com	<button>Edit</button> <button>Del</button>
4		Gayla	Bertrand	gayla.bertrand@mecallapi.com	<button>Edit</button> <button>Del</button>
5		Benjamin	Chaz	benjamin.chaz@mecallapi.com	<button>Edit</button> <button>Del</button>
6		Delia	Robin	delia.robin@mecallapi.com	<button>Edit</button> <button>Del</button>
7		Hector	Graves	hector.graves@mecallapi.com	<button>Edit</button> <button>Del</button>
8		Diego	Greene	diego.greene@mecallapi.com	<button>Edit</button> <button>Del</button>
9		Izzie	Mooney	izzie.mooney@mecallapi.com	<button>Edit</button> <button>Del</button>
10		Romeo	Garner	romeo.garner@mecallapi.com	<button>Edit</button> <button>Del</button>

Data is deleted







...and a deep understanding, this section will help you understanding the basic of developing web application by using just HTML, CSS, and JavaScript. With the basic that you have, I strongly believe that you will be able to faster adopt popular frameworks in the market, such as, React, Angular, Vue. And in some cases, you might not need to use those frameworks at all.

*Article by Karn Yongsiriwit, Ph.D. College of Digital Innovation Technology, Rangsit University*

