

Example 1

1. Install node <https://nodejs.org/en/download/> - Open your system terminal command on windows - Mac > spotlight with CMD+Space > search terminal.
2. Check Node version and NPM version `node -v npm -v`
3. Create db.json file locally - add valid JSON data setup custom routes if needed as properties in the data object
4. Install json-server on your system using the terminal `npm install -g json-server` (prefix with `sudo` if on a mac and you encounter permissions issue.)
5. Run the JSON server package - `json-server --watch db.json` - Open your web browser to `http://localhost:3000/posts`

```
{
  "posts": [
    {
      "id": 1,
      "title": "json-server",
      "author": "typicode"
    },
    {
      "id": 2,
      "title": "Test 1",
      "author": "Laurence Svekis"
    }
  ],
  "comments": [
    {
      "id": 1,
      "body": "some comment",
      "postId": 1
    }
  ],
  "profile": {
    "name": "typicode"
  },
  "tester": {
    "val": 1
  }
}
```

http://localhost:3000/

Example 2

```
<!DOCTYPE html>
<html>

<head>
  <title>JavaScript API</title>
  <style>
    * {
      box-sizing: border-box;
    }

    .container {
      text-align: center;
      margin: auto;
    }

    .message {
      font-size: 1em;
      color: #222;
      padding: 20px;
      border: 1px solid #ddd;
    }

    .loadPosts {
      display: block;
      width: 50%;
      font-size: 1.5em;
      background-color: blue;
      color: white;
      margin: auto;
    }

    .dashboard button {
      color: white;
      font-size: 1em;
      border-radius: 15px;
    }

    .dashboard button:hover {
      opacity: 0.7;
    }
  </style>
</head>

<body>
  <div class="container">
    <div class="message">
      <div class="loadPosts">
        <div class="dashboard">
          <button>Dashboard</button>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

.dashboard button:nth-child(1) {
    background-color: green;
}

.dashboard button:nth-child(2) {
    background-color: red;
}

.box {
    display: block;
    width: 70%;
    border: 1px solid #ddd;
    margin: 5px auto;
    padding: 10px;
}

.box input {
    width: 90%;
    display: block;
    padding: 3px;
    margin: 5px auto;
    font-size: 1.2em;
}

.box button {
    padding: 10px;
    width: 40%;
    cursor: pointer;
}

.sellInfo {
    width: 30%;
    display: inline-block;
    height: 40px;
    font-size: 1.5em;
}
</style>
</head>

<body>
<div class="container">
    <input type="text" class="val">
    <button class="btn">Click</button>

```

```

    <div class="output"></div>
  </div>
  <script src="apps1.js"></script>
</body>

</html>
const container = document.querySelector('.container');
const message = cme2(container, 'div', 'Message Area');
message.classList.add('message');
const val1 = document.querySelector('.val');
const output = document.querySelector('.output');
const baseurl = 'http://localhost:3000/';
const btn1 = document.querySelector('.btn');
const btn2 = cme2(container, 'button', 'Load Posts');
btn2.classList.add('loadPosts');
btn1.textContent = 'Create New';
btn1.classList.add('sellInfo');
val1.classList.add('sellInfo');
window.addEventListener('DOMContentLoaded', getAllPosts);
btn1.addEventListener('click', addPost);
btn2.addEventListener('click', getAllPosts);

function addPost(e) {
  console.log('ready');
  e.preventDefault();
  const title = val1.value || 'default title';
  val1.value = "";
  const url = baseurl + 'posts';
  const body = {
    title: title,
    author: 'Laurence Svekis'
  }
  const opts = {
    method: 'POST',
    body: JSON.stringify(body),
    headers: {
      'Content-type': 'application/json; charset=UTF-8'
    }
  };
  fetch(url, opts)
    .then(rep => rep.json())
    .then(data => {
      makeItem(data);
      myMessage(`New Item Made id = ${data.id}`);
    });
}

```

```
    })  
  }  
}
```

```
function myMessage(html) {  
  message.innerHTML = html;  
}
```

```
function getAllPosts(e) {  
  console.log('page ready');  
  const url = baseUrl + 'posts';  
  fetch(url)  
    .then(rep => rep.json())  
    .then(data => {  
      pageContents(data);  
    })  
}
```

```
function pageContents(data) {  
  console.log(data);  
  output.innerHTML = "";  
  data.forEach(el => {  
    makeItem(el);  
  });  
}
```

```
function makeItem(el) {  
  console.log(el);  
  const main = cme(output, 'div', "");  
  main.classList.add('box');  
  const myID = cme(main, 'div', el.id);  
  const in1 = cme(main, 'input', "");  
  in1.value = el.title;  
  const in2 = cme(main, 'input', "");  
  in2.value = el.author;  
  const btns = cme(main, 'div', "");  
  btns.classList.add('dashboard');  
  const bt1 = cme(btns, 'button', 'Update');  
  const bt2 = cme(btns, 'button', 'Delete');  
  bt1.addEventListener('click', (e) => {  
    const json = {  
      title: in1.value,
```

```

        author: in2.value
    };
    updateItem(json, el.id);
    myMessage(`Updated = ${el.id}`);
  })
  bt2.addEventListener('click', (e) => {
    myMessage(`Deleted = ${el.id}`);
    const url = baseUrl + 'posts/' + el.id;
    fetch(url, {
      method: 'DELETE'
    });
    main.remove();
  })
}

```

```

function updateItem(json, id) {
  const url = baseUrl + 'posts/' + id;
  const opts = {
    method: 'PUT',
    body: JSON.stringify(json),
    headers: {
      'Content-type': 'application/json; charset=UTF-8'
    }
  };
  fetch(url, opts)
    .then(rep => rep.json())
    .then(data => {
      console.log(data);
    })
}

```

```

function cme2(parent, typeEle, html) {
  const el = document.createElement(typeEle);
  parent.prepend(el);
  el.innerHTML = html;
  return el;
}

```

```

function cme(parent, typeEle, html) {
  const el = document.createElement(typeEle);
  parent.append(el);
  el.innerHTML = html;
  return el;
}

```

```
{
  "posts": [
    {
      "title": "UPDATE",
      "author": "Helloworld",
      "id": 1
    },
    {
      "title": "new bew",
      "author": "Laurence Svekis",
      "id": 12
    },
    {
      "title": "title",
      "author": "Laurence Svekis",
      "id": 13
    },
    {
      "title": "test",
      "author": "Laurence Svekis",
      "id": 14
    },
    {
      "title": "test",
      "author": "Laurence Svekis",
      "id": 15
    },
    {
      "title": "default title",
      "author": "Laurence Svekis",
      "id": 16
    },
    {
      "title": "Hello World 2",
      "author": "Laurence Svekis",
      "id": 17
    }
  ],
  "comments": [
    {
      "id": 1,
      "body": "some comment",
```

```
        "postId": 1
    },
    ],
    "profile": {
        "name": "typicode"
    },
    "tester": {
        "val": 1
    }
}
```