

JSON SERVER

npm init -y -> package.json erstellen

"start": "json-server --watch db.json" (in package scripts schreiben)

[npm install json-server]

npm start

JAVASCRIPT SERVER

npm init -y -> package.json erstellen

"start": "live-server --port=4000" (in package scripts schreiben=

[npm install json-server]

[npm install w3-css]

npm start

FEHLER

json oder live server nicht installiert sind -> npm install -g (json/live)-server -y

port passt nicht --port 0

"server": "json-server --watch db.json"

<https://www.npmjs.com/package/json-server>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>1. Test</title>
  <script src="./src/index.js" type="module"></script>
  <link rel="stylesheet" href="./node_modules/w3-css/w3.css"
</head>
<body>
<table id="posts" class="w3-table-all">
  <caption><h2>Posts</h2></caption>
  <thead>
    <tr>
      <th>ID</th>
      <th>userId</th>
      <th>Title</th>
      <th>Body</th>
    <tr>
  </thead>
  <tbody id="tbody">
  </tbody>
</table>
<div id="dlg" class="w3-modal">
  <div class="w3-modal-content w3-card-4">
    <header class="w3-container w3-light-gray">
      <span id="clse" class="w3-button w3-display-
topright">&times;</span>
      <h2>Edit Post</h2>
    </header>
    <div class="w3-container">
```

```

        <p><input id="title" type="text" style="width: 100%"/></p>
        <p><textarea rows="5" id="body" style="width: 100%"></textarea></p>
    </div>
    <div class="w3-container">
        <button id="save" class="w3-button w3-round">Save</button>
        <button id="cancel" class="w3-button w3-round">Cancel</button>
    </div>
    <footer class="w3-container w3-light-gray">
        <p>Footer</p>
    </footer>
</div>
</div>
</body>
</html>

```

```

import postsService from "./posts-service.js"
import PostsTable from "./posts-table.js"
import PostsService from "./posts-table.js"

const postsTable = new PostsTable()
window.onload = async () => {
    const posts = await postsService.getAll()
    postsTable.render(posts)
}

```

```

const URL = "http://localhost:3000/posts"

class PostsService {
    async getAll() {
        const resp = await fetch(URL)
        return await resp.json()
    }
    async save(post) {
        const resp = await fetch(`${URL}/${post.id}`, {
            method: "PUT",
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify(post)
        })

        return await resp.json()
    }
}

export default new PostsService()

```

```

import postsService from "../posts-service.js"

const rowTemplate = /*template*/ post => `
  <tr>
    <td>${post.id}</td>
    <td>${post.userId}</td>
    <td>${post.title}</td>
    <td>${post.body}</td>
  </tr>
`

function $(id) {
  return document.getElementById(id)
}

class PostsTable {
  render(posts) {
    console.log("render...", posts)
    const tbody = document.getElementById("tbody")
    const dlg = $("dlg")
    const title = $("title")
    const body = $("body")
    const save = $("save")
    posts.forEach(post => {
      const row = tbody.insertRow()
      row.innerHTML = rowTemplate(post)
      tbody.appendChild(row)
      row.addEventListener("click", () => {
        dlg.style.display = "block"
        //alert(`post ${post.title} clicked`)
        title.value = post.title
        body.value = post.body
        save.onclick = async () => {
          const toSave = {
            ...post,
            title: title.value,
            body: body.value
          }
          const saved = await postsService.save(toSave)
          window.location.reload()
        }
      })
    })
    $("cancel").onclick = () => close()
    $("close").onclick = () => close()
  }
}

function close() {
  dlg.style.display = "none"
}

export default PostsTable

```