My-Website

Development

|  |  |
| --- | --- |
| Abstract | Dieses Handbuch beschreibt |
| Version | 0.1 |
| Status | In Arbeit |
| Ausgabedatum | 25. August 2022 |
| Gültigkeitsbereich | bsd. Bern |
| Dokumentname | Handbuch.dotx |
| Ablage | C:\Users\javad\OneDrive\Schule\BSD\Deutsch\Handbuch.docx |

# Änderungskontrolle

## Änderung

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Ausführende Stelle** | **Bemerkung / Art der Änderung** |
| 0.1 | 22.01.2023 | Javad Qasemi | Start des Dokuments. |
| 0.2 | 28.01.2023 | Javad Qasemi |  |
|  |  |  |  |

## Freigabe

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Freigabedatum** | **Freigebende Stelle(n)** | **Bemerkung** |
| 0.1 | 21.01.2023 | Javad Qasemi | Freigabe der Version 0.1 als Version 0.1 |

Inhaltsverzeichnis

[Änderungskontrolle 3](#_Toc125236568)

[Änderung 3](#_Toc125236569)

[Freigabe 3](#_Toc125236570)

[1 Erste Schritt 5](#_Toc125236571)

[1.1 Make directory. 5](#_Toc125236572)

[2 Verzeichnisse 15](#_Toc125236573)

[2.1 Tabelle 15](#_Toc125236574)

[3 Abbildung 15](#_Toc125236575)

MERN Admin Dashboard

# Erste Schritt

## Make directory.

|  |  |
| --- | --- |
|  |  |
|  | mkdir server |
|  | Directory: D:\My-Projects\Mydesign  Mode LastWriteTime Length Name  ---- ------------- ------ ----  d---- 1/21/2023 11:33 AM server |
|  | PS D:\My-Projects\Mydesign> cd server |
|  | PS D:\My-Projects\Mydesign\server> mpm -y |
| mpm: The term 'mpm' is not recognized as a name of a cmdlet, function, script file, or executable program.  Check the spelling of the name, or if a path was included, verify that the path is correct and try again.  PS D:\My-Projects\Mydesign\server> **where npm**  PS D:\My-Projects\Mydesign\server> **npm init -y**  npm: The term '**npm**' is not recognized as a name of a cmdlet, function, script file, or executable program.  Check the spelling of the name, or if a path was included, verify that the path is correct and try again. | |
| **When you have this issue do this:**   1. Open the Control Panel (Click the Start button, then click Control Panel) 2. Click User Accounts 3. Click Change **my Environment** variables 4. Select **PATH** and click the **Edit**... button. 5. At the end of the Variable value, **add**; **C:\Program Files\nodejs.** 6. Click Ok on the "Edit User Variable" window, then click Ok on the "Environment Variables" window. 7. Start a command prompt window (Start button, then type **cmd** into the search and hit enter) 8. At the prompt (C:\>) type **npm** and hit enter; you should now see some help text (Usage: **npm <command>** etc.) rather than "**npm** *is not recognized...*" | |
|  | PS D:\My-Projects\Mydesign\server> npm install |
| npm ERR! code ENOENT  npm ERR! syscall open  npm ERR! path D:\My-Projects\Mydesign\server/package.json.  npm ERR! errno -4058  npm ERR! enoent ENOENT: no such file or directory, open 'D:\My-Projects\Mydesign\server\package.json'  npm ERR! enoent This is related to npm not being able to find a file.  npm ERR! enoent  npm ERR! A complete log of this run can be found in:  npm ERR! C:\Users\javad\AppData\Local\npm-cache\\_logs\2023-01-21T10\_45\_51\_391Z-debug-0.log | |
| Just Download and Install Node.js from here <https://nodejs.org/en/>  If you run the downloaded file and install it, they will automatically configure for your system  You don't need any other configurations anymore, now you can use the **npm** command anywhere  If the Nodejs is successfully installed and still displays the message like this:  'npm' is not recognized as an internal or external command, operable program or batch file.  Follow the steps below for Windows users:   1. Go to **My Computer** Properties 2. Click **Advanced System Setting** from the Left bar of a window. 3. Now you have a System Properties window. Click **Advanced** 4. Then, Click **Environment Variable** button 5. Now you have Environment variable window: From System Variable, Select **Path** 6. Click **Edit** 7. At the end of the Variable value, add ;C:\Program Files\nodejs\   **Note**: If you have installed nodejs on other drives then please act accordingly.   1. Click **Ok** all the open dialogue box   **Very important Note**: **"Close your Command Prompt And Restart Again"** (It's very important because if you didn't restart your command prompt then changes will not be reflected.)  Now you can use the **npm** command anywhere | |
|  | PS D:\My-Projects\Mydesign\server> npm init -y |
|  | Wrote to D:\My-Projects\Mydesign\server\package.json:  {  "name": "server",  "version": "1.0.0",  "description": "",  "main": "index.js",  "scripts": {  "test": "echo \"Error: no test specified\" && exit 1"  },  "keywords": [],  "author": "",  "license": "ISC"  } |
|  | PS D:\My-Projects\Mydesign> **cd server**  PS D:\My-Projects\Mydesign\server> **npm i express body-parser cors dotenv helmet morgan mongoose nodemon**  added 201 packages, and audited 202 packages in 21s  15 packages are looking for funding  run `npm fund` for details  found 0 vulnerabilities  PS D:\My-Projects\Mydesign\server> |
| Müssen in der vorlage «node\_modules» ein neu «index.js» File erstellen. Dann in File «index.js» neue **imports** erstellen. | import express from "express";  import bodyParser from "body-parser";  import mongoose from "mongoose";  import cors from "cors";  import dotenv from "dotenv";  import helmet from "helmet";  import morgan from "morgan";  */\*Routes import configuration\*/*  import clientRoutes from "./routes/client.js";  import generalRoutes from "./routes/general.js";  import managementRoutes from "./routes/management.js";  import salesRoutes from "./routes/sales.js";  */\*configuration\*/*  dotenv.config();  const app = express();  app.use(express.json());  app.use(helmet());  app.use(helmet.crossOriginEmbedderPolicy({ policy: "cross-origin" }));  app.use(morgan("common"));  app.use(bodyParser.json());  app.use(bodyParser.urlencoded({ extended: false }));  app.use(cors());  */\*router configuration\*/*  app.use("/client", clientRoutes);  app.use("/general", generalRoutes);  app.use("management", managementRoutes);  app.use("/sales", salesRoutes); |
| Dann muss eine neue «routes» Vorlage in der Vorlage Server erstellen. Inder Vorlage «routes» müssen für jedes «Routes» ein neues File erstellen.  Z.B.:   * **Client.js** * **genera.js** * **management.js** * **sales.js** | |
| Dann in jedes File gebe der folgende Code: | import express from "express";  const routes = express.Router();  *export* default routes; |
| Dann müssen wir in der Vorlage «server» einige Files erstellen. Die folgenden Files:   * **controllers** * **data** * **models** | |
| Dann müssen wir die folgenden Files in der Vorlage «controllers» erstellen.  * **Client.js** * **genera.js** * **management.js** * **sales.js** | |
| **Dann müssen wir die Database mit Mongodb Database, mit den folgenden Schritten verbinden.**  [**https://www.mongodb.com**](https://www.mongodb.com)**/Singin**  **Wenn sie kein Konto in MongoDB haben, dann müssen Sie ein Konto erstellen.**  **Wenn Sie haben ein Konto erstellt, dann müssen Sie ein Database und ein User erstellen und den IP-Adresse konfigurieren.** | |
|  | |

# Database Connection

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
| Dann müssen wir ein neu «**.env**» File in der Vorlage **Serve**r erstellen. | |
| Müssen wir in File «.env» die MongoDB mit der folgenden Methode verbinden.  Es ist sehr, dass wir müssen das korrekte Password von Diabase User zu geben.  mongo\_url = mongodb+srv://admqasemi:<password>@cluster0.rfarale.mongodb.net/?retryWrites=true&w=majority  port = "5001 | |
| Dann müssen wir ein neu «**.gitingnore**»File erstellen damit nicht alle Daten in GitHub hochgeladen werden.  In der «**.gitignore**» File geben wir die folgenden Codes damit die Daten von diese beide Files nicht in GitHub hochgeladen werden.  /node\_modules  .env | |
| Dann müssen wir in «package.json» File den folgenden Code geben.  Nach den folgenden Codes:  "name": "server",    "version": "1.0.0",    "description": "",    "main": "index.js",  geben Sie den Folgenden Code:  "type": "commonjs", | |
| Wir müssen nach den script den folgenden Code geben:  "scripts": {      "test": "echo \"Error: no test specified\" && exit 1",  "start": "node index.js",      "dev": "nodemon index.js"  }, | |
| dann müssen wir das App mit «**npm run dev**» starten ob alles funktioniert gut. | |
| PS D:\My-Projects\Mydesign\server> **npm run dev**  > server@1.0.0 dev  > nodemon index.js  [nodemon] 2.0.20  [nodemon] to restart at any time, enter `rs`  [nodemon] watching path(s): \*.\*  [nodemon] watching extensions: js,mjs,json  [nodemon] starting `node index.js`  (node:27128) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.  (Use `node --trace-warnings ...` to show where the warning was created)  D:\My-Projects\Mydesign\server\index.js:1  import express from "express";  ^^^^^^  SyntaxError: Cannot use import statement outside a module  at internalCompileFunction (node:internal/vm:73:18)  at wrapSafe (node:internal/modules/cjs/loader:1159:20)  at Module.\_compile (node:internal/modules/cjs/loader:1203:27)  at Module.\_extensions..js (node:internal/modules/cjs/loader:1293:10)  at Module.load (node:internal/modules/cjs/loader:1096:32)  at Module.\_load (node:internal/modules/cjs/loader:935:12)  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run\_main:84:12)  at node:internal/main/run\_main\_module:23:47  Node.js v19.4.0  [nodemon] app crashed - waiting for file changes before starting... | |
|  | |
| Wenn das Problem ist gelöscht dann müssen wir den MongoDB in «index.js» File mit die folgenden Methoden konfigurieren.  const PORT = process.env.PORT || 3000;  mongoose.connect(process.env.MONGO\_url{      useNewUrlParser: true,      useUnifiedTopology: true,  });.then(() => {      app.listen(PORT, () => console.log('server Port: ${PORT}'))  }).catch((error) => console.log('${error} Sorry you are not connected')); | |
| Dann müssen wir neu File mit der folgenden Methode erstellen:  PS D:\My-Projects\Mydesign> **npx create-next-app client**  √ Would you like to use TypeScript with this project? ... No / Yes  √ Would you like to use ESLint with this project? ... No / Yes  Creating a new Next.js app in D:\My-Projects\Mydesign\client. | |
| Dann müssen wir eigene Packages in der Vorlage Client installieren:  npm i react-redux @reduxjs/toolkit react-datepicker react-router-dom@6 @mui/material @emotion/react @emotion/styled @mui/ic  ons-material @mui/x-datagrid @nivo/core @nivo/bar @nivo/geo @nivo/pie | |
| Dann müssen wir ein neu «**jsconfig.json**» File in der Vorlage «**client**» erstellen. | |
|  | |

# Verzeichnisse

## Tabelle

**No table of figures entries found.**

# Abbildung

**No table of figures entries found.**