

VueCli installieren

Begriffe Definition

Babel

Ein JavaScript Kompiler. Übersetzt Code in EcmaScript, welcher auch mit alten JavaScript Browsern kompatibel ist und vieles mehr

PostCss

Beim kompilieren werden Prefixe an Css Klassen / Ids hinzugefügt, damit in einer SinglePage Applikation nicht plötzlich ein und dieselbe Id für unterschiedliche Komponente genutzt werden

Lint

Ein Prozess wo der Source Code nach potentiellen Fehler durchsucht wird

package.json

Beschreibt die Abhängigkeiten (dependencies) inkl. Version zu anderen Bibliotheken

Jede App in NodeJs hat im Root Verzeichnis der App so eine Datei.

Beim Abrufen von einer App, werden die Abhängigkeiten automatisch mit herunter geladen

sass / less

Dateien werden von z.B. Webpack zu einer Css Datei interpretiert

TypeScript (.ts)

Dateien werden von z.B. Webpack zu einer Js Datei interpretiert
Dadurch sind z.B. Klassen und Vererbungen in JavaScript möglich

NodeJs

Ist eine JavaScript Serverumgebung

Npm

Node-package-manager verwaltet Node Applikationen (packages)

Vue.js

Vue.js ist ein clientseitiges JavaScript-Webframework zum

Erstellen von Single-Page-Webanwendungen

Vuex

Nutzt man mehrere Komponenten, die untereinander noch verschachtelt sind, dann müssen Events immer nach oben gereicht werden.

Mit Vuex wird ein Datenstore geschaffen, der alle Daten abrufen und bereit stellt. Dadurch erhält man eine Zustandsmaschine

Webpack

Module Bundler. Packt mehrere JavaScript, CSS usw. Dateien zu einer minified zusammen

VueCLI

Vue ist das JavaScript Framework und CLI steht für Command Line Interface

Installation VueCLI

1. NodeJS installieren x64 -> <https://nodejs.org/en/download/>
von nun an werden die Befehle in der NodeJS cmd eingegeben.

Node.js Command prompt:

```
# C:\Program Files\nodejs\nodevars.bat
```

2. Vue Cli installieren -> `npm i -g @vue/cli`

```
# Die Apps werden hier gespeichert: %UserProfile%/node_modules
```

3. `vue ui` -> startet den Service für Vue Cli Ui Interface
<http://localhost:8000/dashboard>

4. Vue global startbar machen

```
npm install -g @vue/cli-service-global
```

5. In PowerShell mkcert installieren

```
choco install mkcert
```

Damit erstellt man für die eigene Entwicklungsmaschine ein SSL Zertifikat. Zertifikate von CA können nicht mit localhost arbeiten und Self Signed werden von Applikationen nicht anerkannt

6. Über `cd [... Pfad zu meiner Applikation]` navigieren

```
7. mkcert -install
```

8. `ipconfig` -> Eigene IP rausfinden (IPv4-Adresse)

. . . . □

9. (Durch eigene IPv4 ersetzen) `mkcert localhost 127.0.0.1 192.168.178.47 ::1`

Ordnerstruktur Vue App

node_modules Ordner

Hier liegen alle Bibliotheken (Modules), die für das Vue Projekt benötigt werden

public/Index.html

Die index.html ist die einzige Html Seite, die an den Client übertragen wird. Der Inhalt wird in sie rein gerendert. Und zwar hier zwischen:

Sobald man buildet, werden dort auch 2 Skript Tags generiert, die den JavaScript Teil enthalten

src/assets (Engl. Teile, Zubehör)

Hier liegen die Bilddateien, Fonts etc.

src/components

Hier liegen eigene Vue Komponente

src/App.vue

Beispiel Komponente

src/main.js

Diese JavaScript Datei ist die Haupt Datei, die sagt, nehme die und die JavaScript Komponente und rendere sie in Vue hinein

.gitignore

Regeln für Git, welche Dateien in das Projekt kommen und bei Änderungen hochgeladen werden sollen

babel.config.js

Einstellung von Babel, z.B welche Version von EcmaScript genutzt werden soll

package.js

Konfigurationsdatei vom Vue Projekt.

- Name, Version
- "scripts" welcher NodeJs Service beim builden, deployen und linten genutzt werden soll
- "dependencies" welche Bibliotheken in welcher Version

genutzt werden sollen

- “devDependencies” welche Bibliotheken ausschließlich im Entwicklungsmodus genutzt werden sollen

Postcss.config.js

Konfigurationsdatei für PostCss

README.md

Dokumentationsdatei. Wird in GitHub z.B. als Doku angezeigt:

<https://guides.github.com/features/mastering-markdown/>

Tsconfig.json

Konfigurationsdatei für TypeScript

Tslint.json

Konfigurationsdatei für TypeScript Lint

dist Ordner

Mit dem Befehl “npm run build” wird die App deployed und in das Verzeichnis dist gepackt. Auf einen produktiven Server kommt also nur der Inhalt vom dist Ordner

ASP Core 3.1 inkl VueCLI mit mehreren VueApps verwenden

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.SpaServices;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.FileProviders;
```

```

using Microsoft.Extensions.Hosting;
using VueCliMiddleware;

namespace ASPMultipleVueSPA
{
public class Startup
{
// This method gets called by the runtime. Use this method to
add services to the container.
// For more information on how to configure your application,
visit https://go.microsoft.com/fwlink/?LinkID=398940
public void ConfigureServices(IServiceCollection services)
{
}

// This method gets called by the runtime. Use this method to
configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app,
IWebHostEnvironment env)
{
if (env.IsDevelopment())
{
app.UseDeveloperExceptionPage();
}

app.UseRouting();

app.UseEndpoints(endpoints =>
{
endpoints.MapGet("/", async context =>
{
await context.Response.WriteAsync("Hello World!");
});
});

/*
* // Wichtig: In vue.config.js publicPath eintragen! z.B.
module.exports = {
publicPath: '/en/'
}
*/

```

```
};
*/

endpoints.MapToVueCliProxy(
pattern: "en/{*path}",
options: new SpaOptions { SourcePath = "ClientApp" },
npmScript: (System.Diagnostics.Debugger.IsAttached) ? "serve"
: null,
port: 8080,
https: false,
runner: ScriptRunnerType.Npm,
regex: "Compiled successfully",
forceKill: true,
wsl: false
);

endpoints.MapToVueCliProxy(
pattern: "fr/{*path}", // Wichtig: In vue.config.js publicPath
eintragen: publicPath: '/fr/'
options: new SpaOptions { SourcePath = "ClientApp2" },
npmScript: (System.Diagnostics.Debugger.IsAttached) ? "serve"
: null,
port: 8081,
https: false,
runner: ScriptRunnerType.Npm,
regex: "Compiled successfully",
forceKill: true,
wsl: false
);

});

}
}
}
```