

IHK Abschluss-Projekt 2017

Jeremy Seipelt

5. Dezember 2017

Für den Beruf des Fachinformatiker der Anwendungsentwicklung

Agentur Website

Ein Full-Stack Web-Developer Projekt

SOURCE CODE

Auszubildner:

Jeremy Seipelt

Pichelsdorferstr. 72

13595 Berlin



Ausbildungsbetrieb:

zmb GmbH

Scharnhorstr. 112

11234 Berlin

Inhaltsverzeichnis

1	Services	3
1.1	DrupalService	3
2	Components	4
2.1	HeroSectionComponent	4
2.2	PageComponent	5
3	Server	8
3.1	Node Main	8
3.2	NGINX	9
4	Other	12
4.1	Gulp	12
4.2	package.json	14
4.3	Compiler	15

1 Services

1.1 DrupalService

firstline

```
import { Injectable, PLATFORM_ID, Inject } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs/Observable';
import { isPlatformServer, Location } from '@angular/common';

//import * as request from 'request';
//import { isPlatformServer } from '@angular/common';

@Injectable()
export class DrupalService {
  public baseApi: string;

  constructor(@Inject(PLATFORM_ID) private _platformId: object, private httpClient: HttpClient) {
    //TODO create way to store Endpoint more dynamic
    if (isPlatformServer(_platformId)) {
      this.baseApi = "https://zmb.agency/drupal/index.php"
    } else {
      console.log(location);
      this.baseApi = "//zmb.agency/drupal/index.php"
    }
  }

  getPage(page: string): Observable<any>{
    console.log(this.baseApi + '/export/pages/' + page);
    return this.httpClient.get(this.baseApi + '/export/pages/' + page);
  }

  getPages(): Observable<any>{
    return this.httpClient.get(this.baseApi + '/export/pages/all');
  }

  getParagraph(section: string): Observable<any>{
    return this.httpClient.get(this.baseApi + '/export/paragraphs/' + section);
  }

  getParagraphs(): Observable<any>{
    return this.httpClient.get(this.baseApi + '/export/paragraphs/all');
  }

  getHeadBar(): Observable<any>{
```

```
        return this.httpClient.get(this.baseApi + '/export/headbar');
    }

    getForm(string:string):Observable<any>{
        return this.httpClient.get(this.baseApi + '/webform_rest/elements/' + string + '?_form');
    }

    /**
     * [Because of the limitations of URL Parsing we use ":" as "/" for the cms set url]
     * @param {string} string [description]
     * @return {[type]} [description]
     */
    transformUrl2Api(string:string){
        //Remove first slash when applicable
        if(string[0] === "/"){
            string = string.slice(1,string.length);
        }

        return string.replace("/",":");
    }
}
```

2 Components

2.1 HeroSectionComponent

```
import {Component,Input,HostBinding} from '@angular/core';

@Component({
    selector: 'hero-image-section',
    template: `
<div class="hero-image-div" [ngStyle]="{'background-image': field_backgroundimage }">
    <h1>
        <span class="title" *ngIf="section.field_header">{{section.field_header[0].value}}</span>
        <br>
        <span class="subtitle" *ngIf="section.field_subheader">{{section.field_subheader[0].value}}</span>
    </h1>
</div>
    `,
    styleUrls:['./HeroSectionStyle.css'],
})
export class HeroSectionComponent
{
```

```
@Input() section: any;

headline: any;
subline: any;
body: any;
field_backgroundimage: string = "none";

/*
@HostBinding( 'style.background' )
field_backgroundimage: string = "url('http://52.29.161.188/sites/default/files/2017-10/Ad
*/

public css: any;

constructor() {}

ngOnInit()
{
    this.field_backgroundimage = 'url(' + this.getHeroImageUrl() + ')';
}

/**
 * retrieves Hero Image URL from the Section Metadata else ""
 * @return {string} [Hero IMAGE URL]
 */
public getHeroImageUrl(): string
{
    if (this.section.field_backgroundimage[0]) {
        return this.section.field_backgroundimage[0].url;
    }
    return "";
}

}
```

2.2 PageComponent

```
import {Component, ComponentFactoryResolver, ViewContainerRef, ComponentRef, ViewEncapsulation} from '@angular/core';
import {DrupalService} from '../services/DrupalService';
import { Router, NavigationEnd } from '@angular/router';
import { Meta, Title } from '@angular/platform-browser';
import {Observable} from 'rxjs/Observable';
import 'rxjs/add/operator/toPromise';
import {HeroSectionComponent} from '../sections/hero/HeroSectionComponent';
import {FormSectionComponent} from '../sections/form/FormSectionComponent';
```

```
import {RichTextSectionComponent} from "../sections/imprint/ImprintSectionComponent";

const paragraphDictionary = {
  "heroimage_paragraph" : HeroSectionComponent ,
  "form_paragraph" : FormSectionComponent ,
  "richtext_paragraph" : RichTextSectionComponent ,
  "err-section ":" PlaceholderComponent "
};

@Component({
  selector: 'page',
  styleUrls:[ './ PageComponentStyle.css '],
  template:'',
  encapsulation: ViewEncapsulation.None
})
export class PageComponent {
  public page:Observable<any>;
  public sections:any[];

  private component: ComponentRef<any>;

  constructor(private drupalService:DrupalService , private factoryResolver:ComponentFactoryResolver) {

  }

  ngOnInit(){

    this.router.events.subscribe((data)=>{
      if( data instanceof NavigationEnd){
        this.updatePage();
      }
    });

    this.updatePage();

  }

  updatePage() {

    if(this.router.url === "/"){
      this.page = this.drupalService.getPage("startseite");
    }else{
      console.log(this.router.url);
      this.page = this.drupalService.getPage(this.drupalService.transformUrl2Api(this.router.url));
    }
  }
}
```

```
this.page.subscribe((data:any) =>{
    console.log(data);
    if(data[0].field_paragraph){
        this.setSections(data[0].field_paragraph);
    }
    this.sections = this.sections.map(section => section.toPromise());

    if(data[0].field_description[0]){
        this.meta.updateTag({ charset:"utf8",content:data[0].field_description[0].value});
    }

    if(data[0].field_seo_titel[0]){
        this.title.setTitle(data[0].field_seo_titel[0].value as string);
    }else{
        this.title.setTitle(data[0].title[0].value as string);
    }

    Promise.all(this.sections).then(this.onSectionsRdy.bind(this));

});

}

/**
 * [Callback Function executed when all Section promises resolved]
 * @param {any} sections [Section]
 */
onSectionsRdy(sections:any){

    //Its Possible that old view container where loaded so we clear em here
    this.viewContainer.clear();

    sections.forEach((section:any) => {
        try{
            let type = this.getSectionType(section[0]);
            const factory = this.factoryResolver.resolveComponentFactory(paragraphDictionary.get(type));
            this.component = this.viewContainer.createComponent(factory);
            this.component.instance.section = section[0];
        }catch(error){
            console.error("Could not add Section component for " + section + "\n Error: " + error);
        }
    });
}
```

```
public setSections(sections:any[]){
    this.sections = sections.map((section) => {return this.drupalService.getParagraph(se
})

/**
 * Takes a section and returns it type
 * @param {any}    section
 * @return {string} [section Type]
 */
private getSectionType(section:any):string{

    if(section.type && section.type[0]){
        return section.type[0].target_id;
    }

}

}
```

3 Server

3.1 Node Main

```
import 'zone.js/dist/zone-node';
import 'rxjs';
import 'reflect-metadata';
import * as express from 'express';
import { ngExpressEngine } from '@nguniversal/express-engine';
import routes from './app/routing/routes';

//IMPORTANT import your app root module from a precompiled angular app aka factory
import {AppServerModuleNgFactory} from './app/app.server.ngfactory';

//setup express server
const server = express();

// Set the express-engine to bootstrap our root module to the application as html
server.engine('html', ngExpressEngine({
    bootstrap: AppServerModuleNgFactory // Give it a module to bootstrap(u should use ROOT Mod
})));

//function to render request to index
function ngApp(req:any, res:any) {
    res.render('./index', {req, res});
}
```



```
//set the view engine to the server engine
server.set('view engine', 'html');

//set the views directory to out
server.set('views', 'out');

//set the server to handle all request
server.get('**/*', ngApp);

//set server to use the out folder as static for handling static files
server.use(express.static('out'));

//set server port to x default 8000
server.listen('1337');
```

3.2 NGINX

```
server {
    listen *:80;
    server_name zmb.agency www.zmb.agency;
    return 301 https://zmb.agency;
}

server {
    listen *:442 ssl http2;
    ssl_certificate_key /etc/letsencrypt/live/zmb.agency/privkey.pem;
    ssl_certificate /etc/letsencrypt/live/zmb.agency/fullchain.pem;
    server_name www.zmb.agency;
    return 301 https://zmb.agency;
}

server {
    listen *:443 ssl http2;
    ssl_certificate_key /etc/letsencrypt/live/zmb.agency/privkey.pem;
    ssl_certificate /etc/letsencrypt/live/zmb.agency/fullchain.pem;
    server_name zmb.agency;
    root /data/www;
    index index.html index.php;
    error_log /var/log/nginx/debug.log debug;

    location = /favicon.ico {
        log_not_found off;
        access_log off;
    }

    location = /robots.txt {
        allow all;
    }
}
```

```
    log_not_found off;
    access_log off;
}

# Very rarely should these ever be accessed outside of your lan
location ~* \.(txt|log)$ {
    allow 192.168.0.0/16;
    deny all;
}

location ~ \..*/.*\.php$ {
    return 403;
}

location ~ ^/sites/*/private/ {
    return 403;
}

# Allow "Well-Known URIs" as per RFC 5785
location ~* ^/.well-known/ {
    allow all;
}

# Block access to "hidden" files and directories whose names begin with a
# period. This includes directories used by version control systems such
# as Subversion or Git to store control files.
location ~ (^|/)\. {
    return 403;
}

location /drupal/ {
    # try_files $uri @rewrite; # For Drupal <= 6
    index index.php;
    root /data/www/drupal;
    #      try_files $uri /index.php?$query_string; # For Drupal >= 7
    expires 2d;
}

location @rewrite {
    rewrite ^/(.*)$ /index.php?q=$1;
}

# Don't allow direct access to PHP files in the vendor directory.
location ~ /vendor/.*\.php$ {
    deny all;
    return 404;
}
```

```
location ~ /\.php(/|$) {
    include snippets/fastcgi-php.conf;
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
    # With php7.0-cgi alone:
    # fastcgi_pass 127.0.0.1:9000;
    # With php7.0-fpm:
    fastcgi_pass unix:/run/php/php7.0-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

# Fighting with Styles? This little gem is amazing.
# location ~ ^/sites/*/files/imagecache/ { # For Drupal <= 6
location ~ ^/sites/*/files/styles/ { # For Drupal >= 7
    try_files $uri @rewrite;
}

# Handle private files through Drupal. Private file's path can come
# with a language prefix.
location ~ ^(/[a-z\-\_])?/system/files/ { # For Drupal >= 7
    try_files $uri /index.php?$query_string;
}

location ~* /drupal/(.+\. (js|css|png|jpg|jpeg|gif|ico|svg))$) {
    try_files $uri @rewrite;
    expires 7d;
    log_not_found off;
}

location /out/
{
    expires 7d;
    alias /home/ubuntu/stack/angular/out/;
}

location /dev/out/
{
    expires 7d;
    alias /home/ubuntu/stack/dev/out/;
}

location /dev {
    proxy_pass http://localhost:1336/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
```

```
        proxy_cache_bypass $http_upgrade;
    }

    location / {
        proxy_pass http://localhost:1337/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

4 Other

4.1 Gulp

```
firstline
/*copy src into cache*/
var gulp    = require('gulp');
var ts      = require('gulp-typescript');
var ngc     = require('gulp-ngc');
var webpack = require('webpack');
var merge2  = require('merge2');
var sass    = require('gulp-sass');
var tar     = require('gulp-tar');
var gzip    = require('gulp-gzip');

var tsProject = ts.createProject('tsconfig.json', {"include": ["build/**/*.ts"]});

gulp.task('setup', function() {
    return merge2(
        gulp.src('./src/**/*.ts', {base: 'src'}),
        gulp.src('./src/out/**/*.ts', {base: 'src'}).pipe(gulp.dest('./dist')),
        gulp.src('./package.json').pipe(gulp.dest('./dist')),
        gulp.src('./package-lock.json').pipe(gulp.dest('./dist'))
    )
    .pipe(gulp.dest('./build'));
});

gulp.task('copyFonts', function() {
    return gulp.src('./fonts/**/*.ttf', {cwd: './node_modules/roboto-fontface/'}).pipe(gulp.dest('./'));
});

gulp.task('compile:sass', function() {
```

```
    return gulp.src('./src/**/*.scss',{base:'src'})
      .pipe(sass())
      .pipe(gulp.dest('./build'));
  })

gulp.task('compile:angular',[ 'setup','compile:sass'], function() {
  return ngc("tsconfig-aot.json");
});

gulp.task('compile:typescript', [ 'compile:angular'], function() {
  var tsResult = tsProject.src()
    .pipe(tsProject());

  return tsResult.js.pipe(gulp.dest('dist'));
});

gulp.task('compile:webpack', [ 'compile:angular'], function(callback) {
  webpack(require('./webpack.config.js'), (err, stats) => {
    if (err || stats.hasErrors()) {
      console.log(err);
      const info = stats.toJson();

      if (stats.hasErrors()) {
        info.errors.forEach(error => console.error(error));
      }

      if (stats.hasWarnings()) {
        info.warnings.forEach(warning => console.warn(warning));
      }
    }

    callback();
  });
});

gulp.task('compress',[ 'compile','setup','copyFonts'],function(){
  return gulp.src('./dist/**/*.*)
    .pipe(tar('dist.tar'))
    .pipe(gzip())
    .pipe(gulp.dest('./'))
})

gulp.task('compile',[ 'compile:angular','compile:webpack',"compile:typescript"]);
```

```
gulp.task('build', ['setup', 'compile', 'copyFonts', 'compress']);
```

4.2 package.json

```
{
  "name": "zmbssr",
  "version": "1.0.0",
  "description": "zmb SSR Basic Test",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Jeremy Seipelt",
  "license": "ISC",
  "dependencies": {
    "@angular/animations": "^4.4.6",
    "@angular/common": "^4.4.6",
    "@angular/core": "^4.4.6",
    "@angular/forms": "^4.4.6",
    "@angular/http": "^4.4.6",
    "@angular/platform-browser": "^4.4.6",
    "@angular/platform-browser-dynamic": "^4.4.6",
    "@angular/platform-server": "^4.4.6",
    "@angular/router": "^4.4.6",
    "@nguniversal/express-engine": "^1.0.0-beta.3",
    "@types/express": "^4.0.37",
    "@uirouter/angular": "^1.0.0-beta.7",
    "express": "^4.15.4",
    "preboot": "^5.1.7",
    "reflect-metadata": "^0.1.10",
    "request": "^2.81.0",
    "roboto-fontface": "^0.8.0",
    "rxjs": "^5.0.1",
    "zone.js": "^0.8.4"
  },
  "devDependencies": {
    "@angular/compiler": "^4.4.6",
    "@angular/compiler-cli": "^4.4.6",
    "@types/request": "^2.0.3",
    "gulp": "^3.9.1",
    "gulp-gzip": "^1.4.0",
    "gulp-ngc": "^0.3.2",
    "gulp-sass": "^3.1.0",
    "gulp-tar": "^2.0.0",
    "gulp-typescript": "^3.2.2",
    "gulp-webpack": "^1.5.0",
  }
}
```

```
    "merge2": "^1.1.0",  
    "ts-loader": "^2.3.4",  
    "typescript": "^2.4.2",  
    "webpack": "^3.5.5"  
  }  
}
```

4.3 Compiler

```
{  
  "compilerOptions": {  
    "baseUrl": "src/app",  
    "module": "ES2015",  
    "target": "ES5",  
    "moduleResolution": "node",  
    "removeComments": true,  
    "emitDecoratorMetadata": true,  
    "experimentalDecorators": true,  
    "allowSyntheticDefaultImports": true,  
    "sourceMap": true,  
    "noEmit": true,  
    "noEmitHelpers": true,  
    "strictNullChecks": false,  
    "outDir": "dist",  
    "lib": [  
      "es2015",  
      "dom"  
    ],  
    "typeRoots": [  
      "node_modules/@types"  
    ],  
    "noImplicitAny": true,  
    "suppressImplicitAnyIndexErrors": true  
  },  
  "exclude": [  
    "node_modules",  
    "dist",  
    "out"  
  ],  
  "files": [  
    "build/app/app.server.ts",  
    "build/app/app.ts"  
  ],  
  "angularCompilerOptions": {  
    "genDir": "./",  
    "trace": true,  
    "skipMetadataEmit": true,  
  },  
}
```

```
        "strictMetadataEmit": true
    },
    "compileOnSave": false ,
    "buildOnSave": false
}
```