11 Angular Universal(25)

Session_25_Universal(02-ng-universal-finished)
347 Prequisites for Angular Universal
348 Creating the Server Rendering
352 Deploy it to the server
(lazy loading script is inlcuded on the offline databse)

https://ide.c9.io/laczor/angular2

More info:
https://github.com/angular/angular-cli/wiki/stories-universal-rendering

https://malcoded.com/posts/angular-fundamentals-universal-server-side-rendering

347 Prequisites for Angular Universal

Universal means that the content is rendered on the server and the content is provided to the browser, for SEO optimalization.

Things to know:
- You have to setup, configure it using the upper link
- Ensure you are not manipulating the DOM at all! since at the server side there is no DOM!

348 Creating the Server Rendering

Follow the instructions:
https://github.com/angular/angular-cli/wiki/stories-universal-rendering

0. Ok
1. ok
2.Use the following code:
main.server.ts

```
import {enableProdMode} from '@angular/core';

export { AppServerModule } from './app/app.server.module';




enableProdMode();
```

3. Cut the **tsconfig.json** file from the src to the root folder, next to the src folder.
Parse the following code:

```
{
  "compileOnSave":false,   //have the same behvior then in the updated cli version
  "compilerOptions": {
    // "baseUrl": "",
    "declaration": false,
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "lib": [
      "es2017",
      "dom"
    ],
    "mapRoot": "./",
    "module": "es2015",
    "moduleResolution": "node",
    "outDir": "../dist/out-tsc",
    "sourceMap": true,
    "target": "es5",
    "typeRoots": [
      "../node_modules/@types"
    ]
  }
}
```

3.1 Create a **config.app.json** in the src folder

```
{
  "extends": "../tsconfig.json",
  "compilerOptions": {
    "outDir": "../out-tsc/app",
    "baseUrl": "./",
    "module": "es2015",
    "types": []
  },
```

```
  "exclude": [
    "test.ts",
    "**/*.spec.ts"
  ]
}
```

3.2 Create a **config.server.json** in the src folder

```
{
  "extends": "../tsconfig.json",
  "compilerOptions": {
    "outDir": "../out-tsc/app",
    "baseUrl": "./",
    "module": "commonjs",
    "types": []
  },
  "exclude": [
    "test.ts",
    "**/*.spec.ts"
  ],
  "angularCompilerOptions": {
    "entryModule": "app/app.server.module#AppServerModule"
  }
}
```

4. modify your **.angular.cli.json**

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "project": {
    "name": "n5-complete-guide"
  },
```

```
  "apps": [
    {
      "root": "src",
      "outDir": "dist",
      "assets": [
        "assets",
        "favicon.ico"
      ],
      "index": "index.html",
      "main": "main.ts",
      "polyfills": "polyfills.ts",
      "test": "test.ts",
      "tsconfig": "tsconfig.app.json",
      "testTsconfig": "tsconfig.spec.json",
      "prefix": "app",
      "styles": [
        "../node_modules/bootstrap/dist/css/bootstrap.min.css",
        "styles.css"
      ],
      "scripts": [],
      "environmentSource": "environments/environment.ts",
      "environments": {
        "dev": "environments/environment.ts",
        "prod": "environments/environment.prod.ts"
      }
    },
    {
      "name": "universal",
      "platform": "server",
      "root": "src",
      "outDir": "dist-server",
      "assets": [
```

```
      "assets",
      "favicon.ico"
    ],
    "index": "index.html",
    "main": "main.server.ts",
    "tsconfig": "tsconfig.server.json",
    "prefix": "app",
    "styles": [
      "../node_modules/bootstrap/dist/css/bootstrap.min.css",
      "styles.css"
    ],
    "scripts": [],
    "environmentSource": "environments/environment.ts",
    "environments": {
      "dev": "environments/environment.ts",
      "prod": "environments/environment.prod.ts"
    }
  }
],
"e2e": {
  "protractor": {
    "config": "./protractor.conf.js"
  }
},
"lint": [
  {
    "project": "src/tsconfig.app.json",
    "exclude": "**/node_modules/**"
  },
  {
    "project": "src/tsconfig.spec.json",
    "exclude": "**/node_modules/**"
```

```
    },
    {
      "project": "e2e/tsconfig.e2e.json",
      "exclude": "**/node_modules/**"
    }
  ],
  "test": {
    "karma": {
      "config": "./karma.conf.js"
    }
  },
  "defaults": {
    "styleExt": "css",
    "component": {}
  }
}
```

4.2 Bundle all the files

```
// This builds the client application in dist/browser/
$ ng build --prod
...
// This builds the server bundle in dist/server/, since we listed at the.angular.cli.json as a
 second parameter it can be referenced as [1]
$ ng build --prod --app 1 --output
```

4.3 Write your own build scripts, to have proper names
package.json

```
"build:ssr": "ng build --prod && ng build --prod --app 1 --output-hashing=none",
```

4.4 Creating both server + client side angular

```
npm run build:ssr
```

5. **Setup a Server**
INstall exrpess + rendering engine agnular templates to html +

```
npm install --save express                                          //using express f
ramework
npm install --save @nguniversal/express-engine                      //rendering angul
ar templates to html on the server
npm install --save @nguniversal/module-map-ngfactory-loader         //Allows to creat
e/render the different modules
```

Create a **server.js** in the main folder
then run **node server.js**

```
'use strict';                               //We are writing javascript here,


require('zone.js/dist/zone-node');          //angular uses it for changes detection
require('reflect-metadata');                //To be able to handle the metadata


const express = require('express');                                  //To handle exp
ress
const ngUniversal = require('@nguniversal/express-engine');          //rendering angular tem
plates to html on the server
const { provideModuleMap } = require('@nguniversal/module-map-ngfactory-loader');  //Allows to
 create/render the different modules


const { AppServerModuleNgFactory, LAZY_MODULE_MAP } = require('./dist-server/main.bundle');


//we are on the server side, getting requst and provide responses.
function angularRouter(req, res) {
  res.render('index', {req, res});                  //Render our index page, and pass the r
eq,res in an ojbect
}
```

```
const app = express();


//Setup our rendering engine setup, to use for the rendering.
app.engine('html', ngUniversal.ngExpressEngine({
  bootstrap: AppServerModuleNgFactory,
  providers: [
    provideModuleMap(LAZY_MODULE_MAP)
  ]
}));
//rendering html files as inputs and outputing html files
app.set('view engine', 'html');
//We want to render our views to a dist folder
app.set('views', 'dist');
//upon recieving this url we execute the function, which renders ourd index.html
app.get('/', angularRouter);
//we can use static files.
app.use(express.static(`${__dirname}/dist`));
//redirect everything, to the angular routers.
app.get('*', angularRouter);


app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

5. Include Lazy Loading

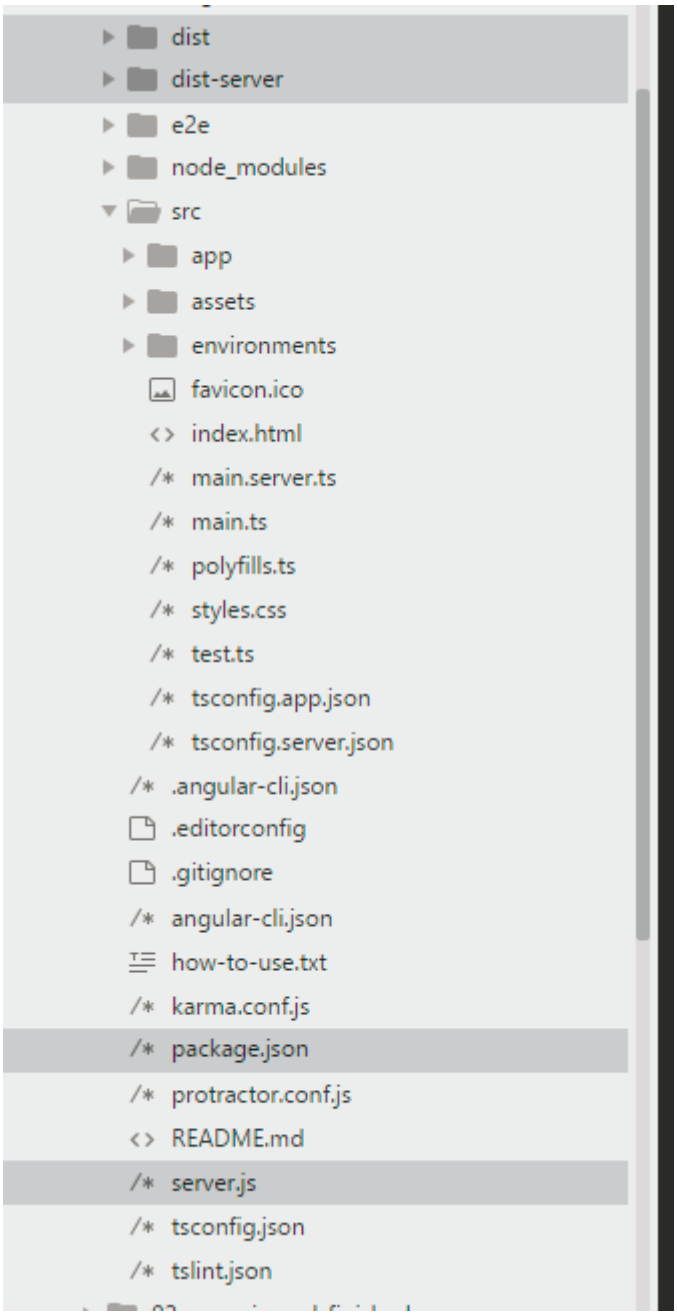
add the following code to the **main.server.ts**

```
import { enableProdMode } from '@angular/core';


import { AppModule } from './app.module';
```

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
//Installed separatly
import { ServerModule } from '@angular/platform-server';
import { ModuleMapLoaderModule } from '@nguniversal/module-map-ngfactory-loader';

import { AppComponent } from './app.component';

@NgModule({
   declarations: [],
   imports: [
       //Make sure the string matches
       BrowserModule.withServerTransition({
           appId: 'my-app-id'
       }),
       ServerModule,
       AppModule,
       ModuleMapLoaderModule // The new module
   ],
   providers: [],
   bootstrap: [AppComponent]
})
export class ServerAppModule {}

enableProdMode();
```

352 Deploy it to the server

The only thing you have to to is to put on the node.js server the following files, install the dependencies and run node **server.js**


dist
dist-server
e2e
node_modules
src
app
assets
environments
favicon.ico
index.html
main.server.ts
main.ts
polyfills.ts
styles.css
test.ts
tsconfig.app.json
tsconfig.server.json
.angular-cli.json
.editorconfig
.gitignore
angular-cli.json
how-to-use.txt
karma.conf.js
package.json
protractor.conf.js
README.md
server.js
tsconfig.json
tslint.json