Server Side Rendering Angular

# Overview

Typically angular applications is running only in browser. This is called CSR - Client side rendering. However Angular Universal runs angular application in server. This is Call SSR.

We have confusion about angular universal and SSR but these both meaning is same. SSR is a nickname of Angular Universal.

Angular Universal runs application on the server side , generating static applications pages that’s why you can not see any loader before like CSR. basically here first paint of HTML is quickly, In contrast to CSR.

# Advantages.

1. First benefits of SSR is for **SEO** you can see your HTML page in page source that can enable easily SEO.
2. On Mobile Device, You can see performance is increased.
3. Initial page load is faster.

# Disadvantages.

1. You can face that **TTFB** is slow here. But still we didn’t figure out any solution.
2. You need basic knowledge of **node Js** to run angular app through SSR.

# How to Implement ?

1. Please update angular application to **latest version**.
2. If possible please **remove third party** packages from application especially ‘ng2’ prefix.
3. You can use scaffolding for installing SSR using ng add **@nguniversal/express-engine --clientProject your-app-name** or you can add all related files manually.
4. You need to create main.server.ts , app.server.module.ts , server.ts , tsconfig.server.json and webpack.server.config.js files.
5. You also need to change ‘.angular.json.cli’ file. Add server platform are there. As below snippet.

|  |  |
| --- | --- |
| { |  |
|  | "platform": "server", |
|  | "root": "src", |
|  | "outDir": "dist-server", |
|  | "main": "server.main.ts", |
|  | "tsconfig": "server.tsconfig.app.json", |
|  | "testTsconfig": "tsconfig.spec.json", |
|  | "environmentSource": "environments/environment.ts", |
|  | "environments": { |
|  | "dev": "environments/environment.ts", |
|  | "prod": "environments/environment.prod.ts" |
|  | } |
|  | } |

1. You add your different environments also there otherwise your environment did not work.
2. Add build command in package json file.

"scripts": {

"build:browser": "ng build --prod --app 0",

"build:server": "ng build --prod --app 1 --output-hashing none",

"build": "npm run build:browser && npm run build:server",

"serve": "node server.js"

},

1. Run angular application using npm run build:ssr && npm run serve:ssr using this command. It will run your server.js file from dist folder that is generated by build command .

# Problems In SSR ?

1. You can face that **localstorage , document and window** are not working in ssr.

Solution: - You can use below snippet for this. Use these three inside platform-browser.

**import { Component, OnInit,PLATFORM\_ID } from '@angular/core';**

**import { isPlatformBrowser } from '@angular/common';**

**constructor(@Inject(PLATFORM\_ID) private platformId: Object) { }**

**ngOnInit() {**

**if (isPlatformBrowser(this.platformId)) {**

**localStorage.setItem('Key','Value');**

**// window and document also will work same.**

**}**

**}**

if (isPlatformBrowser(this.platformId)) {

**// You can write here anything that you don’t want to run on server.**

}

1. If You should implement 301 redirects then you can use server.js for SSR.

app.get('\*', (req, res) => {

if (req.headers.host === 'pynora.com') {

res.redirect(301, 'https://www.pynora.com' + req.url);

} else {

res.render('index', { req });

}

});

# Related links

<https://angular.io/guide/universal>

<https://medium.com/@MarkPieszak/angular-universal-server-side-rendering-deep-dive-dc442a6be7b7>

<https://itnext.io/server-side-rendering-ssr-in-angular-5-the-simplest-and-quickest-ssr-approach-34cf53224f32>