

# Angular Universal

## Module Introduction

- **Angular Universal** let's us **pre-render** Angular **applications** on their **servers**
- User's receive the **wholly-loaded** page upon entry
- Without Angular Universal, we'll see a page that relies on **JavaScript code** to present Angular content and capabilities
  - Poor for **slower networks**
  - **Search engines** only see what's provided by the **server**, not the actual page's **contents** that rely on a **JavaScript download**

## Adding Angular Universal

- Within the **Angular CLI**, call **ng add @nguniversal/express-engine --clientProject <file-name>**
  - **<file-name>** is the first field after **projects** in **angular.json**
- Generate **app.server.module.ts** with **npm install --save @nguniversal/module-map-ngfactory-loader**
  - In this file, ensure that you add **AppModule**, **ServerModule**, and **ModuleMapLoaderModule** to the **imports** array
- We can determine if we're **running** on the **server** by **injecting @Inject(PLATFORM\_ID) private platformId** and calling **isPlatformBrowser(this.platformId)**
- You can **build** the **app** for Angular Universal with **npm run build:ssr**
- You can **run** the **app** with **npm run serve:ssr**

## Adding Angular Universal with NestJS

- We can **emulate** this as well by using **NestJS** via **ng add @nestjs/ng-universal**
- Provides **universal rendering** and an entire **NestJS** application
  - Good for a full-stack application

## Deploying Universal Apps

- You **can't deploy** Angular Universal apps to static hosts
  - This is because hosts don't support **NodeJS**, which we need
  - No more **Firebase**