Thursday: Deployment

Deploying Angular Apps

After we create an application, we can deploy it to a remote server so that users can interact with it online. There are various ways of deploying applications and the one that has lots of advantages is Cloud Hosting. There are many cloud platforms which offer hosting services such as AWS(Amazon Web Services), Google Cloud Platform and Microsoft Azure. Most of these cloud hosting platforms offer payment plans for hosting but in our case, we will use Github which is free.

Let's, therefore, see how we deploy our angular application and publish it to gh-pages on Github. At this point, we all have Github accounts so let's do the first thing.

1. Create Github repo

If you had not created a Github repository prior to this, go ahead and create a repo in your Github account. After creating the repo, commit all your changes and push them to the repository you have created.

If you had created your repo prior to this, also commit and push all your changes to your Github repository.

2. Install angular-cli-ghpages

We have a tool at our disposal that we can use to easily deploy our angular app to gh-pages, the angular-cli-pages tool. We use this tool as a command on the angular CLI for the purpose of deployment. To install it, let us run this command on our terminal:

```
$ npm i angular-cli-ghpages --save-dev
```

This command has installed angular-cli-ghpages globally in our operating system. We, therefore, don't have to install it again in case we need to use it.

3. Run build

We need to build our angular app for use in production. This simply means configuring our app to be useable on a remote server. While we were creating our app, we were serving our application on our local development server **http://localhost:4200/** (http://localhost:4200/) . The configuration for the local server will not serve the application when it is deployed on the remote server, gh-pages, which is why we need to build our app. Let's run this command on our terminal to build the application:

```
$ ng build --prod --base-href "https://GithubUserName.github.io/GithubRepoName/"
```

Make sure you put in your Github username in GithubUserName and the repository name you created in GithubRepoName.

4. Deploy to gh-pages

It's now time to use angular-cli-ghpages. Let's run this command in our terminal:

```
$ npx angular-cli-ghpages --dir=dist/Project-name
```

Make sure you put the name of the project in the place of Project-name. You can find this in the *angular.json* file under *defaultProject* which is at the bottom of the file.

This command has created a gh-pages branch on Github for us. It has then read all the changes in our remote branch and pushed these changes to gh-pages while maintaining the build configuration that will ensure our application works while on the remote server. If we visit the URL we created earlier on our terminal, https://GithubUserName.github.io/GithubRepoName/,we can see our application running remotely, which means it has successfully been published on gh-pages.