Configuration d'un Projet Angular

Prérequis

1. Node.js et npm (Node Package Manager)

Angular nécessite Node.js et npm pour fonctionner. Si vous ne les avez pas encore installés, téléchargez et installez la version la plus récente de Node.js depuis `https://nodejs.org`.

Vous pouvez vérifier l'installation en exécutant les commandes suivantes dans votre terminal :

```
node -v
npm -v
```

2. Angular CLI

Angular CLI (Command Line Interface) est un outil de ligne de commande qui simplifie le processus de création, de gestion et de test des applications Angular.

Pour installer Angular CLI globalement, exécutez la commande suivante :

```
npm install -g @angular/cli
```

Vérifiez l'installation de Angular CLI :

```
ng version
```

Étape 1 : Créer un Nouveau Projet Angular

1. Créer le projet

Dans le terminal, allez dans le répertoire où vous souhaitez créer votre projet Angular. Par exemple :

```
cd /chemin/vers/dossier
```

Utilisez la commande `ng new` pour créer un nouveau projet Angular :

```
ng new nom-de-votre-projet
```

Cette commande va vous poser quelques questions :

- Voulez-vous utiliser Angular Routing ? Si vous avez besoin de la gestion des routes, répondez `Yes`.
- Choisir un format de feuille de style : CSS, SCSS, Sass, Less, ou Stylus. Choisissez le format que vous préférez (CSS par défaut).

Après avoir répondu aux questions, Angular CLI va créer le projet dans un dossier avec le nom que vous avez spécifié.

2. Naviguer dans le Dossier du Projet

Une fois le projet créé, allez dans le répertoire du projet :

```
cd nom-de-votre-projet
```

3. Lancer le Serveur de Développement

Pour vérifier que tout fonctionne bien, lancez le serveur de développement avec la commande suivante :

```
ng serve
```

Cette commande va :

- Compiler le projet.
- Lancer un serveur de développement sur `http://localhost:4200`.

Vous pouvez maintenant ouvrir votre navigateur et accéder à l'application Angular en naviguant vers `http://localhost:4200`. Si tout fonctionne correctement, vous verrez la page d'accueil Angular par défaut.

Étape 2 : Structure du Projet

Un projet Angular généré par Angular CLI a la structure suivante :

```
nom-de-votre-projet/
 e2e/                          # Tests end-to-end
 node_modules/                 # Dépendances du projet
 src/                          # Code source de l'application
    app/                      # Composants, services, etc.
    assets/                   # Fichiers statiques (images, etc.)
    environments/             # Fichiers de configuration pour différents environnements
    index.html                # Point d'entrée de l'application
    main.ts                   # Point d'entrée de l'application (fichier TypeScript)
    styles.css                # Feuille de style globale
 angular.json                  # Fichier de configuration du projet Angular
 package.json                  # Dépendances et scripts npm
 tsconfig.json                 # Configuration TypeScript
```

Étape 3 : Ajouter un Composant

1. Pour créer un nouveau composant, utilisez la commande suivante :

```
ng generate component nom-du-composant
```

Ou la forme abrégée :

```
ng g c nom-du-composant
```

Cette commande va générer les fichiers nécessaires dans le dossier `src/app/nom-du-composant` :

- `nom-du-composant.component.ts` (le fichier TypeScript du composant)
- `nom-du-composant.component.html` (le template HTML du composant)
- `nom-du-composant.component.css` (les styles du composant)
- `nom-du-composant.component.spec.ts` (le fichier de test unitaire)

Étape 4 : Ajouter un Service

Les services sont utilisés pour fournir des données et des fonctionnalités dans l'application Angular.

1. Pour générer un service, utilisez la commande :

```
ng generate service nom-du-service
```

Ou la forme abrégée :

```
ng g s nom-du-service
```

Cela va créer un fichier de service dans `src/app/nom-du-service.service.ts`.

Étape 5 : Ajouter des Routes

Pour configurer des routes dans Angular, vous devez : 1. Créer un fichier `app-routing.module.ts` si ce n'est pas déjà fait. 2. Définir les routes et importer le `RouterModule`.

Exemple :

```
1
2  import { NgModule } from '@angular/core';
3  import { RouterModule, Routes } from '@angular/router';
4  import { HomeComponent } from './home/home.component';
5  import { AboutComponent } from './about/about.component';
6
7  const routes: Routes = [
8    { path: '', component: HomeComponent },
9    { path: 'about', component: AboutComponent }
```

```
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
\end{verbatim}

\section*{ tape  6 : D ployer l'Application}
Une fois votre application pr te , vous pouvez la construire pour
    la production en ex cutant la commande suivante :
\begin{verbatim}
ng build --prod
```

Listing 1: Using a service in a component

Cette commande va créer une version optimisée de l'application dans le dossier `dist/`.

Vous pouvez ensuite déployer les fichiers de ce dossier sur votre serveur de production ou une plateforme cloud comme `Firebase`, `AWS`, ou `Netlify`.