

# Cours 1 : Angular JS

## Angular :

Angular JS est un framework Javascript coté client basé sur TypeScript, développé par Google. Il est principalement utilisé pour la création de sites one-page qui ne nécessitent pas de rechargement de page. Angular est basé sur une architecture du type MVC, ainsi les différents aspects du site sont séparées pour mieux êtres gérées.

Les avantages que ce framework offre sont nombreux :

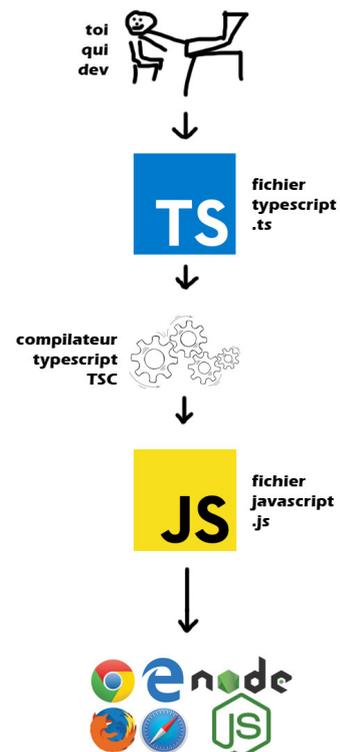
- La testabilité du code est améliorée
- Le code peut être facilement réutilisé dans des projets différents
- Angular JS fonctionne côté client, néanmoins, il apporte des services initialement trouvées côté serveur (contrôleurs de vue)
- Les applications web développées avec Angular sont très légères côté serveur
- Angular propose des outils de routing et de gestion du http
- Il est open source et développé par Google, ainsi le support est très présent
- Le Typescript est un langage stable et rapide

## Typescript :

Le Typescript est un sur-ensemble de JavaScript ce qui signifie que tout programme JS est lui-même un programme TS valide. En réalité, TypeScript **transcompile** du code en JS. On écrit un programme en TS qui va être compilé par un TSC puis générer en sortie un fichier .js pour les navigateurs.

L'intérêt d'utiliser TS est qu'il est beaucoup plus sécurisé que JS et s'appuie sur des méthodes de programmation objet.

L'application créée est donc plus renforcée



## Préparation de l'environnement de développement Angular :

1. **Installation de NODE.JS :** <https://nodejs.org/en/download/>
2. **Installation de NPM :** Ouvrir un terminal de commande et taper (*sudo pour Mac/Linux*) :

```
sudo npm install -g npm@latest
```

3. **Installation du CLI Angular :** Taper (*sudo pour Mac/Linux*) :

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

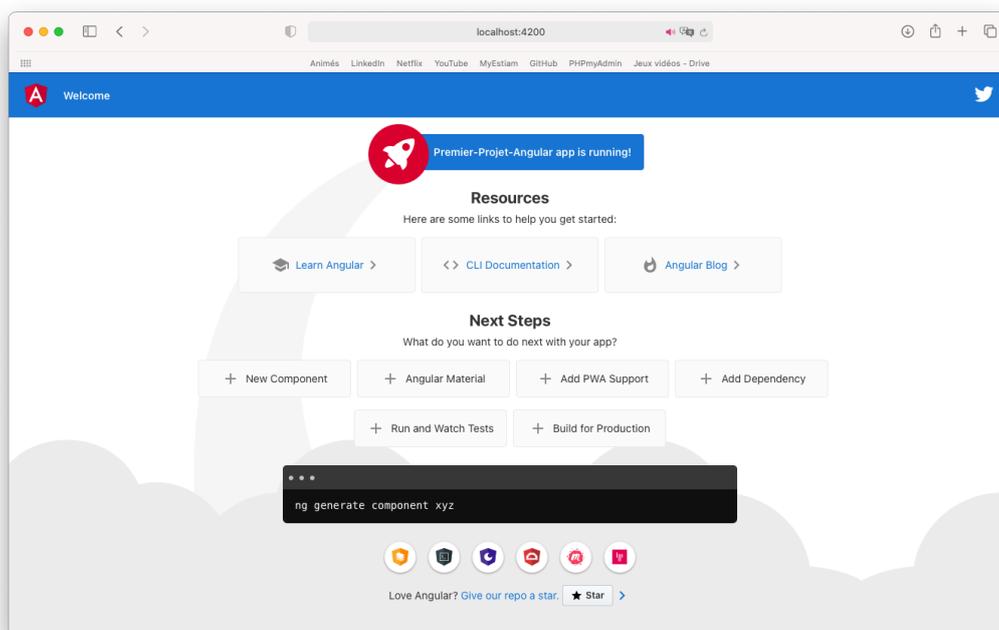
4. **Création d'un projet :** Aller dans le dossier souhaité pour le projet puis taper :

```
ng new premier-projet
```

Après l'exécution de la commande on voit apparaître 2 questions, taper **y** aux 2, puis choisir son langage de style préféré (CSS, Less, SASS, ...).

5. **Tester et ouvrir le projet :** Aller dans le dossier du projet « premier-projet » fraîchement créé :

`cd premier-projet` puis taper `ng serve --open` pour ouvrir lancer le serveur du projet, une fenêtre dans le navigateur devrait apparaître, notifiant que l'application est bien fonctionnelle sur le port 4200 normalement :



L'environnement de développement est maintenant prêt.

**6. Configuration du projet :** Aller dans le dossier du projet « premier-projet » fraîchement créé :

`cd premier-projet` puis taper `npm install bootstrap@4.6.0 --save` pour télécharger Bootstrap et ainsi l'utiliser pour styliser notre page.

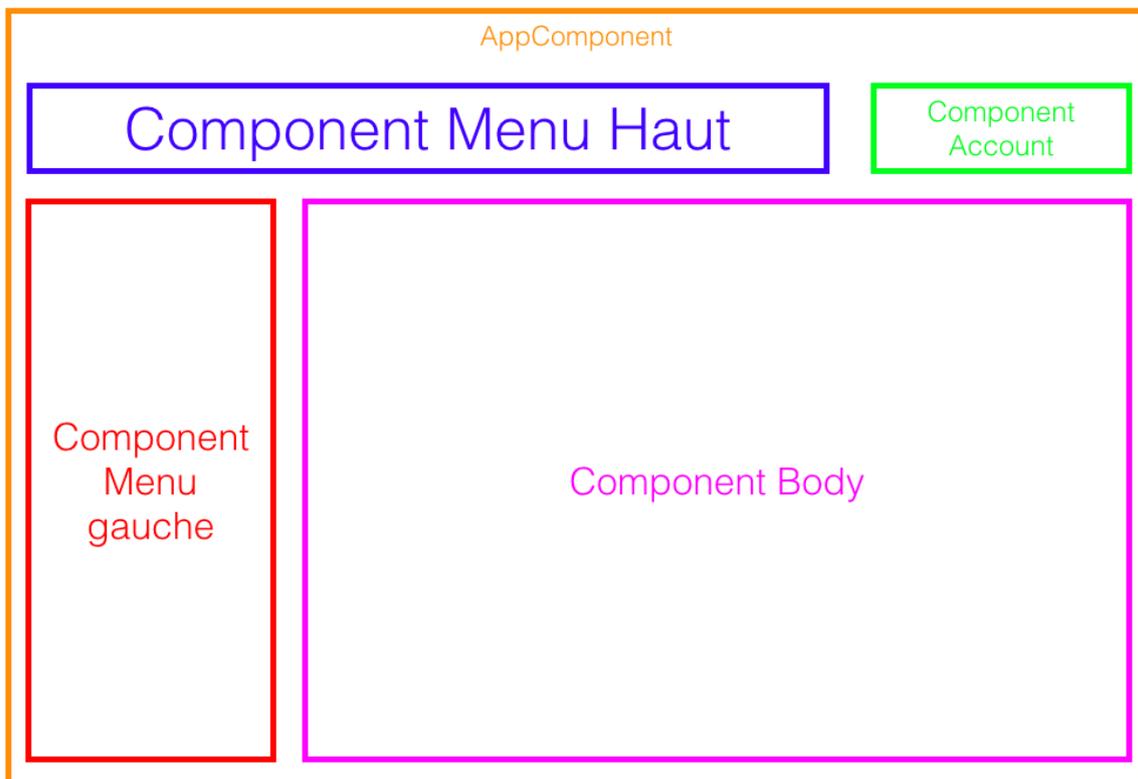
Ouvrir ensuite dans notre éditeur de code préféré le fichier « angular.json » et modifiez le chemin du fichier de style (pour moi c'est à la ligne 30) :

```
30 | "styles": [  
31 |   "src/styles.css"  
32 | ],  
par  
30 | "styles": [  
31 |   "node_modules/bootstrap/dist/css/bootstrap.css"  
32 | ],
```

On relance ainsi le serveur avec la commande `ng serve`.

### **Fonctionnement/composition d'une application Angular :**

Une application codée avec Angular est en réalité une association de plusieurs **components**.



Dans la lourde arborescence de notre projet, on va se concentrer sur le dossier racine « src », c'est ici que vont se trouver les fichiers que nous allons modifier.

En ouvrant le fichier « index.html », on peut apercevoir la balise « <app-root><app-root/> ». C'est ici que va se trouver le contenu dans notre page qui va être importé à l'aide des fichiers contenu dans le dossier « app » dans « src ». On va y trouver les fichiers « app.component. » css/html/ts qui vont contenir les éléments de notre page web.

Dans le fichier « app.component.ts » par exemple, nous allons trouver à la ligne 3 un objet **@Component** qui va contenir le **selector** qui correspond au nom de la balise que l'on attribuera pour contenir les composants de la page web.

Le **templateUrl** qui contiendra le chemin du code HTML ainsi qu'un array **styleUrls** qui contiendra le/les chemins vers les feuilles de style.

```
Premier-Projet-Angular > src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'Premier-Projet-Angular';
10 }
```

## **Sources :**

### **Définition :**

<https://webojob.ch/technologies/javascript/angular>

<https://fr.wikipedia.org/wiki/AngularJS>

<https://openclassrooms.com/fr/courses/4668271-developpez-des-applications-web-avec-angular>

<https://www.jesuisundev.com/comprendre-typescript-en-5-minutes/>