



# Angular



Framework Front-End JavaScript

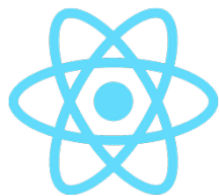


# L'essentiel

- Angular sert à construire des **applications Web** pour mobiles et ordinateur de bureau
- Angular est un « cadriceiel » (*framework*) utilisant une librairie basée sur **TypeScript**
- Angular s'utilise **côté client** sous licence open source et est soutenu essentiellement par Google

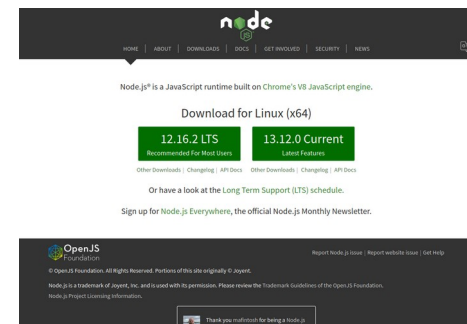
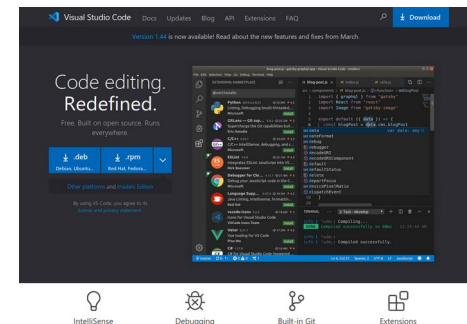
# Comparaison

- Angular fait partie des **trois frameworks** les plus utilisés actuellement
- Angular se situe à la deuxième place derrière **React** et devant **Vue**
- Angular est un framework **applicatif**, **jQuery** est une librairie plus axée sur le design



# Cadre de travail

- Le site d'**Angular** pour la documentation officielle : <https://angular.io/>
- L'environnement de développement (IDE) à utiliser : **Visual Studio Code**  
<https://code.visualstudio.com/>
- L'outil JavaScript à utiliser : **NodeJS**  
<https://nodejs.org/en/>

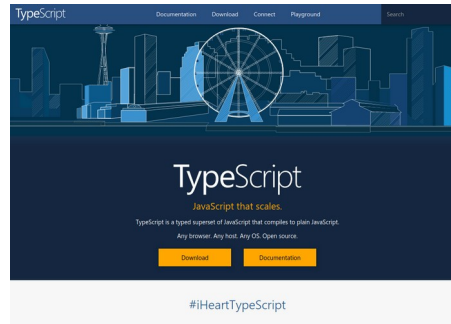




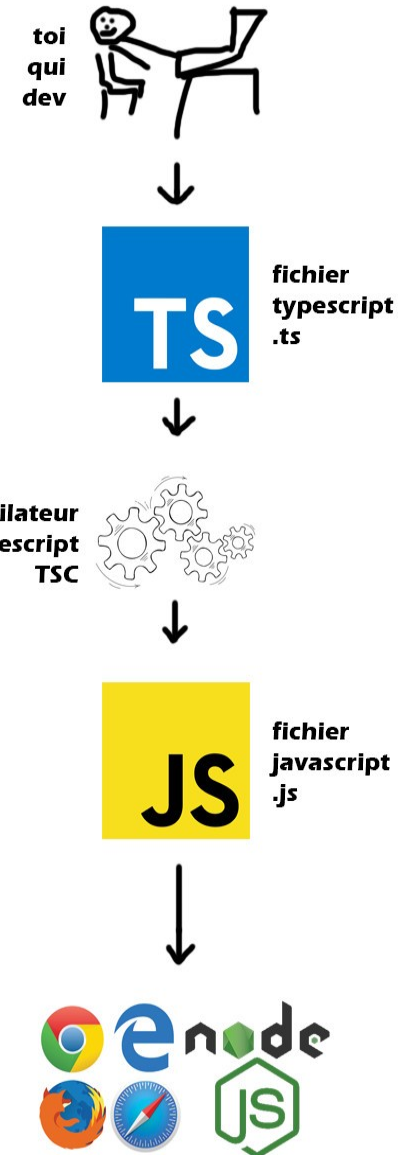
# Fonctionnalités

- Angular gère l'**interface** de l'application Web et les **actions** de l'utilisateur
- Angular gère les échanges des **données** avec des serveurs
- Angular utilise des **services Web** externes (messagerie, API...)

# TypeScript

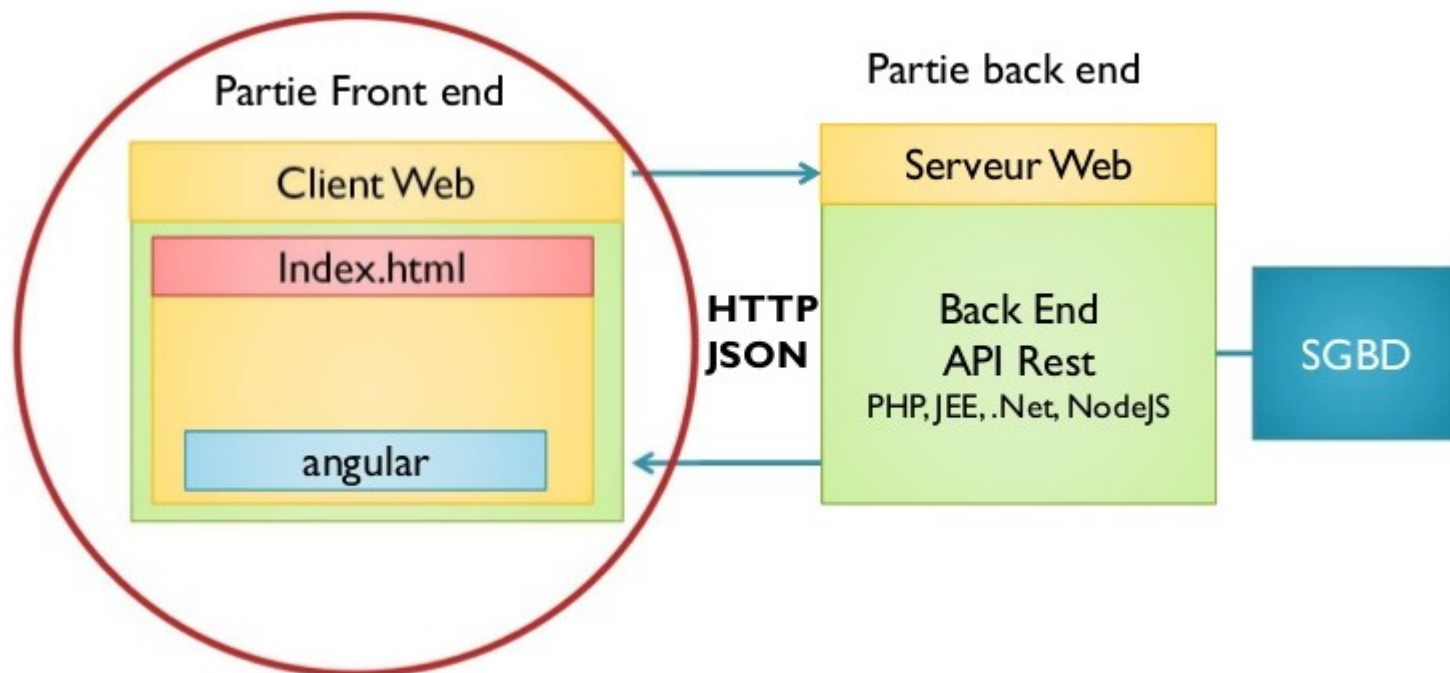


- TypeScript est une **évolution** (ou surcouche) de JavaScript apporté par Microsoft :
- <https://www.typescriptlang.org/>
- TypeScript introduit le **typage** des données et une **approche objet** plus classique
  - TypeScript se compile (ou « transpile ») en équivalent JavaScript



# Principe de la SPA

- SPA signifie **Single Page Application** avec une seule page : `index.html`
- La communication avec un serveur se gère par **AJAX** en appels asynchrones



# Création d'un projet (Windows)

- Pré-requis : **NodeJS** et **Visual Studio Code** installés



- Avec une console Windows PowerShell (admin) :

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

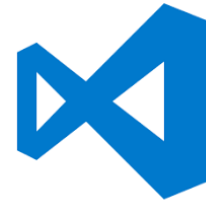
- Ensuite avec le « Node.js command prompt » :

```
mkdir angular  
cd angular  
ng new nom_du_projet --style=scss --routing=false
```



# Création d'un projet (Linux)

- Pré-requis : **NodeJS** et **Visual Studio Code** installés



- Utilisation d'une console utilisateur :

```
sudo npm install -g @angular/cli  
mkdir angular  
cd angular  
ng new nom_du_projet --style=scss --routing=false
```

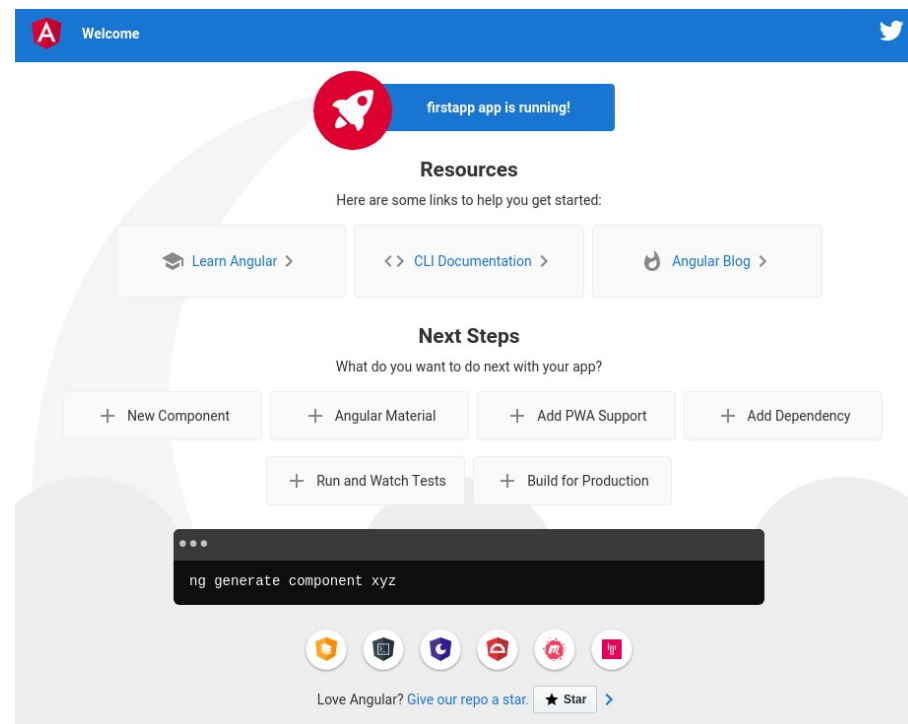
# Explications et lancement

- L'installation du client apporte la **commande** `ng`, commande principale d'Angular
- L'option **style** précise le préprocesseur pour la factorisation des feuilles de styles afin d'éviter les duplications ; **routing** ajoute un fichier de configuration pour les routes de l'application
- Le lancement du projet (ne pas fermer la fenêtre) se fait par :

```
cd nom_du_projet  
ng serve
```

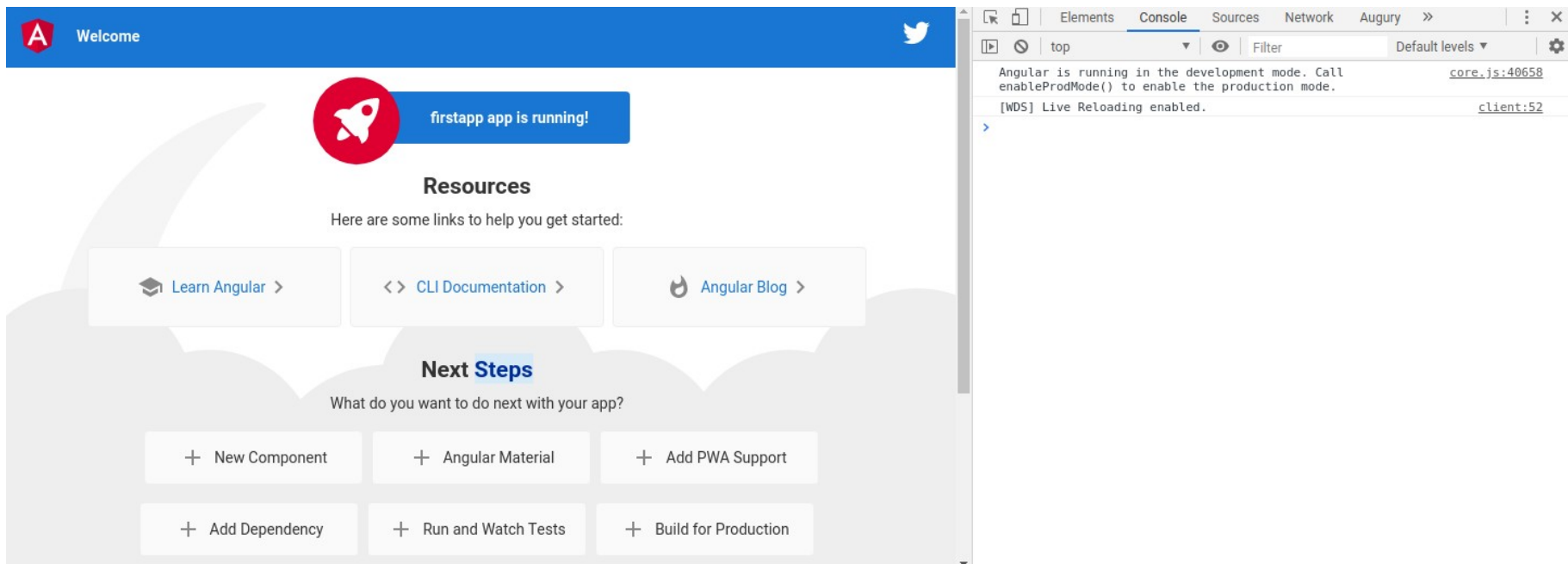
# Résultat du lancement

- Le lancement se fait par l'url `localhost:4200`, en priorité dans un navigateur de type **Chrome** :

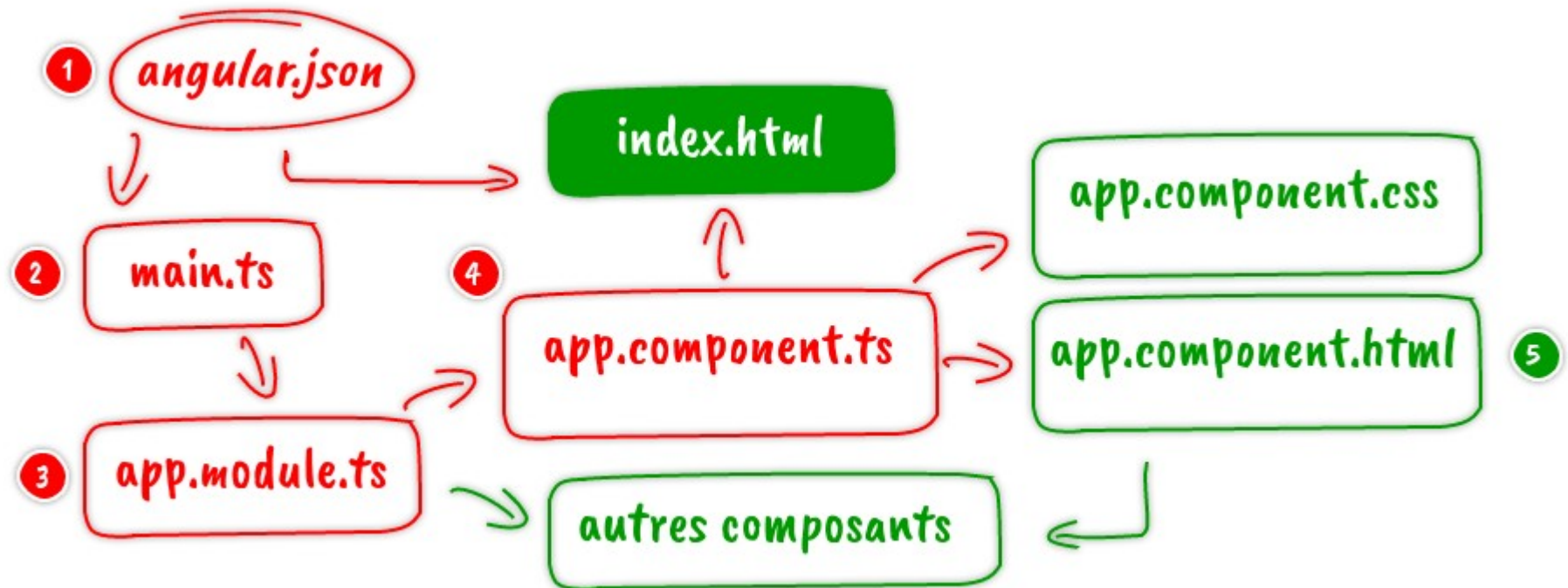


# Aide au développement

- Une **extension** du navigateur apporte des outils pour le développement d'un projet Angular : **Augury**



# Structure d'un projet



1. `angular.json` définit la source HTML et TS
2. `main.ts` importe le module principal `app.module.ts`
3. `app.module.ts` lance l'application avec le composant principal et +
4. `app.component.ts` s'intègre dans le **DOM** (balise `<app-root>`) de `index.html`
5. `app.component.html` utilise d'autres composants

# Notion de composant

- Un composant est une **classe** permettant de générer une **vue** constituée d'un « fragment » HTML
- Une application Angular se compose d'un **arbre de composants**
- Un composant est **réutilisable** au sein d'une **hiérarchie**

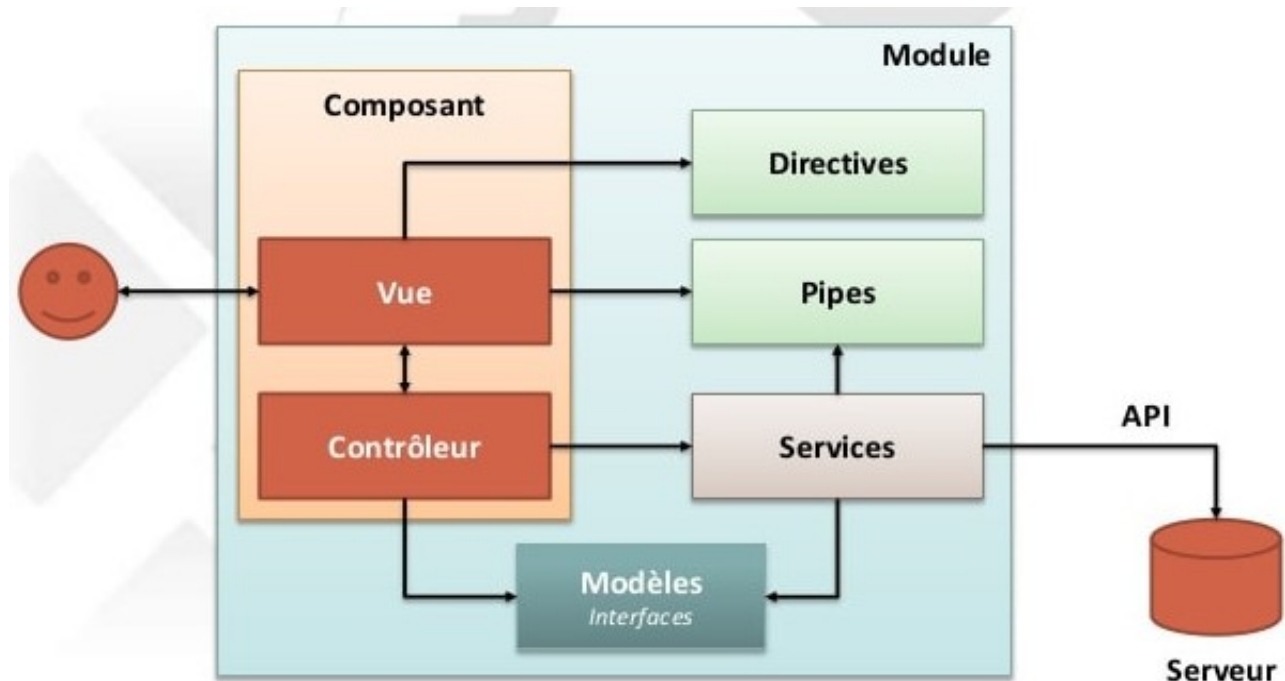


Schéma d'une structure d'une application Angular