Tutoriel Angular

Vous n’êtes probablement pas tombé sur ce site par hasard. Vous êtes ici pour apprendre Angular, alors laissez-moi vous donner un coup de main sur ce «framework». Vous allez voir, c’est sexy.

Ce qui est présenté dans ce tutoriel :

# Débuter avec Angular

Tout d’abord, comme mentionné plus haut, Angular est un framework, c’est-à-dire, une architecture de composants Web ayant pour but de programmer une application complète(fullstack) sur le World Wide Web. Il est écrit en HTML, en CSS, en Typescript, en fichier JSON et j’en passe parce que Angular regroupe tous les éléments nécessaires pour créer votre application.

Enfin, il y a des prérequis :

Premièrement, il est plus facile de s’organiser dans un éditeur de code. Pour ce tutoriel, j’utiliserai Visual Studio Code.

Deuxièmement , il lui faut **Node.js**, un programme qui s’occupe de l’exécution de d’autres programmes dont certaines tâches accomplies par Angular CLI.

Vous pouvez utiliser ce lien pour l’installer : https://nodejs.org/en/

Troisièmement, le client **Node Package Manager(npm)** le gestionnaire de paquets de Node.js, il sert à l’interaction avec les dossiers en ligne de commande. Par défaut, il est inclus avec Node.js.

Quatrièmement , il est essentiel d’installer, à partir du système de «packages» d’Angular, **son interface en ligne de commande(CLI)**. Il sera pratique pour créer des applications et des projets.

Il suffit d’entrer cette commande dans votre invite de commandes pour l’installer globalement sur votre ordinateur :

npm install -g @angular/cli@10.0.0

D’ailleurs, il est optionnel dans l’utilisation d’Angular, mais pour ce tutoriel: vous pouvez installer JSON-SERVER, qui servira de service Web pour simuler l’utilisation d’une base de données. Voici la commande pour l’installer: npm i g json-server.

# Utilisation d’Angular

Utilisation d’Angular CLI

Après son installation, vous pouvez commencer à utiliser votre nouvelle interface en utilisant la syntaxe d’Angular dans cette image.

Créer un projet avec routage

ng n nom\_du\_projet –routing

La création du projet initialise tout ce qu’on a besoin pour configurer Angular avec sa librairie extensive et ses fichiers de configuration JSON.

Les fichiers de meta-données

**Package.json** est un fichier donnant l’information à npm pour gérer les dépendances du projet, les appels des commandes npm dans le tableau «scripts» du fichier. Remplacez dans ce fichier, la ligne ng serve par ng serve -o –port 3333 suite à ‘’start’’ pour ouvrir la page Web lorsque l’application démarre, sur le port local 3333 de votre ordinateur. L’URL de la page votre Web au démarrage sera comme ceci : http://localhost:3333.

**Tsconfig.json** est un fichier de configuration du compilateur TypeScript, un des langages principaux pour développer une application angulaire. Vous n’avez rien à modifier dans ce fichier, mais il pourrait vous être utile dans le futur. Dans Angular, ce fichier est diviser en plusieurs parties ts.config.base.json, tsconfig.app.json, e2e/tsconfig.json et tsconfig.json.

**Angular.json** permet au projet d’utiliser des outils provenant des librairies importées dans le dossier **node\_modules**. Pour utiliser les libraires comme Bootstrap, Jquery et FontAwesome, il faut les installer et les sauvegarder dans le projet. Voici comment faire :

Dans ce fichier, ajoutez ces lignes :

"styles": [ "node\_modules/bootstrap/dist/css/bootstrap.min.css", "node\_modules/@fortawesome/fontawesome-free/css/all.css", "src/styles.css" ], "scripts": [ "node\_modules/bootstrap/dist/js/bootstrap.min.js" ]

Installer BootStrap localement : npm i boostrap –save

Installer Jquery localement : npm i jquery –save

Installer Font-Awesome localement : npm i fortawesome/fontawesome-free –save

**App.module.ts** est une fichier typescript qui est marqué par un décorateur @NgModule qui contient le tableau declarations de Composant et dans le tableau imports de Modules. Les modules devront être importés. Un module est une classe qui donne la possibilité de contrôler certains aspects additionnels d’un composant, comme l’utilisation des services Web avec HttpClientModule. Ajouter les modules HttpClientModule et FormsModule séparés de virgules.

**App-routing.module.ts** est le module ajouté lors de la création du projet Angular en précisant –routing. Il sert à définir les adresses Web complètes du composant dans le tableau de Routes nommé routes.

# Single Page Application(SPA)

Un des avantages de l’utilisation d’Angular est de rendre un site Web avec plusieurs pages et avec beaucoup d’envois d’adresses Web en une seule page, qui navigue entre composants.

# Les composants

Les composants

Leur utilité est de regrouper tout ce qui compose une page Web, ou une partie d’une page Web, en un seul répertoire. Le composant est divisé en parties : Une «back-end» dans un fichier TypeScript où les instructions/codes déterminent l’utilisation du composant. Les classes sont marquées comme composant Angular à l’aide d’un décorateur @Component dans lequel se trouve le selector(identifiant du composant), un templateUrl qui pointe vers le fichier HTML de ce composant ou un template qui est le code HTML inséré dans le même fichier. Puis, il y a un styleUrls ou style qui s’utilise pareillement.

Les composants principaux sont définies dans le tableau bootstrap dans le fichier app.module. Par défaut, le composant principal est AppComponent qui se trouve dans le fichier app.component.ts. L’identifiant de ce composant est app-root et c’est le seul composant qui est appelé sur la page index.html de l’application. Tous les autres composants sont appelés dans le composant principal.

Les classes

La syntaxe des classes sera présentée dans le tutoriel TypeScript ci-joint.

Les directives

Il y a des directives structurelles incrustées dans le «framework», comme \*ngSwitch et le \*ngFor qui permettront d’ajouter de la logique dans les balises HTML pour éviter la répétition de balise ou pour vérifier une condition. Voir image ci-dessous. Puis, il y a l’interpolation qui sert au contrôle des attributs des composants {{nom\_attribut}}.

# Les services

Un service sert à implémenter des méthodes qui , elles, servent à former un lien entre le serveur HTML et le site Web. Pour le moment, concentrons-nous sur le C.R.U.D

Pour créer un service Angular, on commence par le générer avec la commande : ng generate service [–skipTests]. Puis, il faut importer dans notre service le HttpClientModule qui nous servira à faire le lien entre notre service et le serveur Web à l’aide du protocole http.

CREATE

READ

UPDATE

DELETE

Toutes ces méthodes seront assignées à d’autres méthodes pour être déclenchées à la suite d’une événement.

Intro crud

L’utilisation des services requiert l’importation du HttpClientModule dans le fichier app.module.ts, dans le tableau imports de la classe AppModule. On utilise le protocole http pour créer, lire, modifier et supprimer des éléments d’une base de données simulées par un fichier Json.