

FICHE / ANGULAR JS

Objectif :

- Comprendre le rôle du *framework* Angular.JS
- Mettre en œuvre un projet utilisant ce *framework*
- Savoir coder et lire Angular.JS

Angular.JS est un *framework* créée par Google, qui utilise JavaScript.

L'intérêt réside dans le fonctionnement avec NodeJS qui permet de faciliter l'affichage de pages web dynamiques.

Angular.JS a un concept propre à lui (= les directives)

Pour l'installation de Angular.JS, il faut suivre des étapes, que ce soit une installation autonome ou avec Node.JS

Utilisation de Angular.JS :

Angular.JS utilise un *bootstrap* au chargement, pour lire les différents fichiers et préparer la structure.

- >> Intégrer un script Angular.js dans une page web, celle-ci s'exécute et prend la main.
- >> Utiliser les balises propres à Angular.js pour exécuter du code JavaScript.
- >> Utilisation du côté client.

La création d'une application :

Nécessite la création d'un module (*ng-app*) puis permet d'utiliser des directives pour générer les contrôleurs et les vues.

* Mécanisme d'association d'une vue = une **route**

Il existe plusieurs vues associées à l'URL :

✚ **Angular.JS ne nécessite pas Node.JS.**

Il peut interagir avec d'autres serveurs et s'exécute côté client.

✚ **Angular.JS contient une version de jQuery**

(Préférable d'utiliser les concepts d'Angular.JS plutôt que d'intégrer de l'Angular.JS dans une application jQuery)

✚ **Angular.JS applique le modèle MVC** (Modèle – Vue – Contrôleur).

Il est normal d'avoir plusieurs fichiers et devoir lier (bind) les éléments entre eux.

✚ **Angular.JS propose une variable globale \$scope.**

✚ Angular.JS **utilise des instructions commençant ar ng-**.

Il est également possible (grâce aux directives) de créer ses propres préfixes, comme my-

✚ Angular.JS est **un framework de type SPA** (Single Page Application).

Il inclut le concept de contrôleur, de vue, de route, etc

Commande Angulars.Js :

>> Facilite le travail du développeur.

>> Apporte un découplage entre le code HTML et le code JS accompagné d'une excellente interactivité

Comprendre Angular.Js à l'aide d'un exemple :

Après avoir copier le code ci-dessous :

```

HelloWorld.html  X
1  <!DOCTYPE html>
2
3  <html>
4  <head>
5    <meta charset="UTF-8">
6    <script
7      src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></script>
8    </head>
9    <body>
10   <div ng-app>
11     Saisir votre prénom : <input type="text" ng-model="name">
12     <br>
13     Bonjour {{name}} !
14   </div>
15 </body>
16 </html>

```

On obtient ceci :

Saisir votre prénom :

 Bonjour !

Saisir votre prénom :

 Bonjour Yanissa !

Le système est dynamique : quand on tape dans le champ de texte, le résultat s'affiche dans la variable {{ name }}

C'est possible grâce au mécanisme de binding (lien).

ng-model = « name » → {{ name }}

Angular.Js permet d'étendre les capacités du code JavaScript

Utilisation NPM pour Angular.Js

Installation en ligne de commande de Angular.Js

En suivant les étapes, on obtient ceci :

```

Windows PowerShell

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (myfirstapp) apptest
version: (1.0.0)
description: programme 1
entry point: (index.js)
test command:
git repository:
keywords:
author: ygt
license: (ISC)
About to write to C:\Users\59069\Desktop\MyFirstApp\package.json:

{
  "name": "apptest",
  "version": "1.0.0",
  "description": "programme 1",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "ygt",
  "license": "ISC"
}

Is this OK? (yes)
PS C:\Users\59069\Desktop\MyFirstApp>
PS C:\Users\59069\Desktop\MyFirstApp>
69\Desktop\MyFirstApp> npm install angular
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN apptest@1.0.0 No repository field.
+ angular@1.8.0
added 1 package from 1 contributor and audited 1 package in 1.216s
found 0 vulnerabilities
  
```

Cette commande pose quelques questions :

Encadré **en rouge** ce sont les champs importants à compléter ;

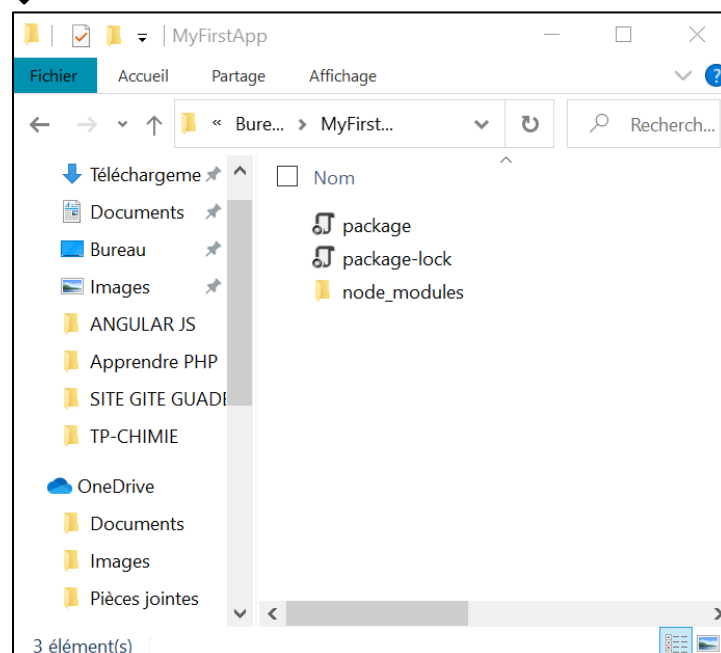
Puis on tape la commande suivante :

npm install angular



On obtient la partie encadrée **en jaune**

Toute les démarches permettent d'obtenir un dossier de travail avec notre environnement



Déclaration d'une application

Le fichier HTML appelle un *framework* JavaScript puis déclare une application Angular.Js. Une application doit contenir des modules. Ils seront donc créés par un script JavaScript (à l'intérieur de l'HTML ou un fichier à part)

Exemples

**** Exemple 1: Application Hello World!**

Code HTML:

```
<!DOCTYPE html>

<html lang="fr">
<head>
  <meta charset="utf-8" />
  <title>My First Angular</title>
  <script src="./node_modules/angular/angular.js"></script>
  <script src="Exemple_World.js"></script>
</head>
<body ng-app="ExempleApp1">
  <h1 ng-controller="ControleurNumero1"> Hello {{maVariable}} !</h1>
</body>
</html>
```

Code pour le fichier Js

```
var MyApp = angular.module("ExempleApp1", []);
MyApp.controller("ControleurNumero1", function($scope) {
  $scope.maVariable = "World";
});
```

- **A la 1^{ère} ligne de ce code** : déclare une application Angular.js
- **2^{ème} ligne** : un contrôleur « ControleurNumero1 » et associe le comportement de fonction \$scope est une variable globale
- **3^{ème} ligne** : affecte « World » à la variable *maVariable*

Problème rencontré :

Lors de l'affichage, au lieu d'obtenir « Hello World »
J'obtiens ceci :

Hello {{maVariable}} !

Solution : Trouver le bon chemin pour que le module fonctionne
Vérifier que le code n'a pas d'erreur ou qu'il ne manque pas un code...

Problème au niveau code du fichier JavaScript résolu :

A l'affichage on a bien :

Hello World !

** Exemple 2 : Le binding !

Code HTML:

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>My First Angular</title>
  <script src="./node_modules/angular/angular.js"></script>
  <script src="exemple_World.js"></script>
</head>
<body ng-app="ExempleApp1">
  <div>
    <h1>Hello {{maVariable}} !</h1>
    <p> saisissez un prénom </p>
    <input type="text" ng-model="maVariable"><br>
  </div>
  <div>
    <hr />
    <h1>Bye {{ "Bye "+maVariable+" !" }}</h1>
  </div>
</body>
</html>
```

À l'affichage :

↓

Hello !

saisissez un prénom

Bye Bye !

Puis lorsque l'on remplit la saisie, on obtient :

Hello Lolo !

saisissez un prénom

Bye Bye Lolo !

Possibilité d'effectuer des opérations :

```
</div>
<div>
  <h2>Calcul : {{(maVariable*1)+2000}}</h2>
</div>
<div>
  <hr />
  <h1>Bye {{ "Bye "+maVariable+" !" }}</h1>
```

Encadré en rouge : Ce code permet d'effectuer une opération. Pour cela, on effectue une multiplication par 1 pour forcer Angular.js à traiter notre variable comme un nombre.

À l'affichage :

↓

Hello !

saisissez un prénom

Calcul : 2000

Bye Bye !

En saisissant un chiffre dans le champ, on obtient :

Hello 20 !

saisissez un prénom

Calcul : 2020

** Exemple 3 : Les listes !

Code HTML:

↓

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Gestion de listes</title>
  <script src="./node_modules/angular/angular.js"></script>
  <script src="Exemple_Liste.js"></script>
</head>
<body ng-app="ExempleApp2" ng-controller="ControleurListe">
  <h1>Voici une liste !</h1>
  <ul>
    <li ng-repeat="ville in listeVilles">{{ville}}</li>
  </ul>
</body>
```

Code JavaScript:

↓

```
var MyApp = angular.module("ExempleApp2", []);
MyApp.controller("ControleurListe", function ($scope) {
  $scope.listeVilles = ["Grenoble", "Lyon", "Paris", "Bordeaux", "Marseille", "Strasbourg"];
});
```

La variable \$scope représente les données de l'application.

À l'affichage :

↓

Voici une liste !

- Grenoble
- Lyon
- Paris
- Bordeaux
- Marseille
- Strasbourg

** Exemple 4 : Les filtres !

Code HTML:



```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Gestion de listes triées</title>
  <script src="./node_modules/angular/angular.js"></script>
</head>
<body ng-app="">
  <p>Une autre liste (avec un filtre et un tri) :</p>
  <div ng-init="personnes=[{nom:'DUPONT', ville:'Paris', age:56},
{nom:'DUPONT', ville:'Lyon', age:45},
{nom:'DAMPIERRE', ville:'Grenoble', age:26},
{nom:'DESTIN', ville:'Grenoble', age:36},
{nom:'WINZALDISKI', ville:'Paris', age:44},
{nom:'DAMPIERRE', ville:'Paris', age:23}
]">
    <ul>
      <li ng-repeat="per in personnes | filter:txtCherche |
orderBy:'ville'">
        {{per.nom}} ({{per.age}}) - {{per.ville}}
      </li>
    </ul>
    Recherche : <input type="text" ng-model="txtCherche" />
  </div>
</body>
</html>
```

À l'affichage :



Une autre liste (avec un filtre et un tri) :

- DAMPIERRE (26) - Grenoble
- DESTIN (36) - Grenoble
- DUPONT (45) - Lyon
- DUPONT (56) - Paris
- WINZALDISKI (44) - Paris
- DAMPIERRE (23) - Paris

Recherche :

Une autre liste (avec un filtre et un tri) :

- DAMPIERRE (26) - Grenoble
- DAMPIERRE (23) - Paris

Recherche :

** Exemple 5 : Affichage masqué ou forcé !

Code HTML:



```
</head>
<body ng-app="">
  <p>Une autre liste (avec un filtre et un tri) :</p>
  <div ng-init="personnes=[
    {nom:'DUPONT', ville:'Paris', cp:75000, age:56},
    {nom:'DUPONT', ville:'Lyon', cp:69000, age:45},
    {nom:'DAMPIERRE', ville:'Grenoble', cp:38000, age:26},
    {nom:'DESTIN', ville:'Grenoble', cp:38000, age:36},
    {nom:'WINZALDISKI', ville:'Paris', cp:75000, age:44},
    {nom:'DAMPIERRE', ville:'Paris', cp:75000, age:23}
  ]">
    <ul>
      <li ng-repeat="per in personnes | filter:txtCherche |
orderBy:'ville'">
        {{per.nom}} ({{per.age}})
        <span ng-hide="masqueVille"> dans {{per.ville}} </span>
        <span ng-show="masqueVille">- {{per.cp}} </span>
      </li>
    </ul>
    Recherche : <input type="text" ng-model="txtCherche.ville" />
    <input type="checkbox" ng-model="masqueVille"> Masquer la vill
  </div>
</body>
</html>
```

À l'affichage :



Une autre liste (avec un filtre et un tri) :

- DAMPIERRE (26) dans Grenoble
- DESTIN (36) dans Grenoble
- DUPONT (45) dans Lyon
- DUPONT (56) dans Paris
- WINZALDISKI (44) dans Paris
- DAMPIERRE (23) dans Paris

Recherche :

☐ Masquer la ville

J'ai transformé la balise `` en une balise de tableau `<td>` :
`<td ng-hide= »masqueVille »>` pour la colonne Ville

Encadré **en jaune**



```
<li ng-repeat="per in personnes | filter:txtCherche |
ville'">
  {{per.nom}} ({{per.age}})
  <td ng-hide=" ! masqueVille"> dans {{per.ville}} </td>
  <span ng-show="masqueVille">- {{per.cp}} </span>
</li>
```

À l'affichage :



Une autre liste (avec un filtre et un tri) :

- DAMPIERRE (26) dans Grenoble
- DESTIN (36) dans Grenoble
- DUPONT (45) dans Lyon
- DUPONT (56) dans Paris
- WINZALDISKI (44) dans Paris
- DAMPIERRE (23) dans Paris

Recherche :

☐ Masquer la ville

Une autre liste (avec un filtre et un tri) :

- DAMPIERRE (26) dans Grenoble - 38000
- DESTIN (36) dans Grenoble - 38000
- DUPONT (45) dans Lyon - 69000
- DUPONT (56) dans Paris - 75000
- WINZALDISKI (44) dans Paris - 75000
- DAMPIERRE (23) dans Paris - 75000

Recherche :

☒ Masquer la ville

Petite Pause : les erreurs possibles

FICHE n°2 / ANGULAR JS

Routage Angular.Js

L'usage des ancres (# dans une URL) permet de traiter le lien URL, sans envoyer de requête vers le serveur.

1- Explications :

Le routage = option intéressante pour les *frameworks*

L'avantage de mettre en place des routes, est d'afficher une URL explicite, plutôt qu'une URL complexe :

- URL complexe : <http://monsite/index.php?id=17&profil=aR3ft&mode=vt>
- URL explicite : <http://monsite/user/adm/edit>

Il faut établir un tableau de routage comme cet exemple :

URL	Page réelle
/consultation	/web/patient/consultation.html
/operation	/web/chu/service/operer.html
/accueil	/index.html

**** ng-route** : permet la création des routes grâce au composant **\$routeProvider** (fournisseur)

Ce module permet de lire les URL grâce au composant **\$route** (= **service**)

**** ng-view** : insérer le *template* de la page.

Installation ngRoute :

J'ai créé un nouveau dossier nommé : MyFirstAngularJs

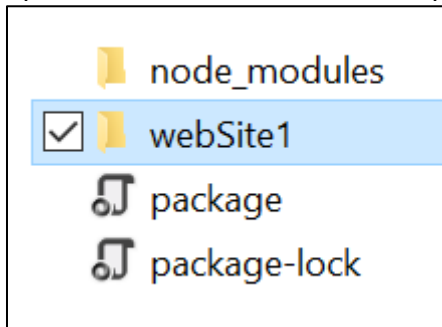
Puis j'ai un fichier PowerShell afin de saisir la commande suivante (encadré **en jaune**)

```
Windows PowerShell
PS C:\Users\59069\Desktop\MyFirstAngularJs> npm install --save angular-route
npm WARN ENOENT: no such file or directory, open 'C:\Users\59069\Desktop\MyFirstAngularJs\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN ENOENT: no such file or directory, open 'C:\Users\59069\Desktop\MyFirstAngularJs\package.json'
npm WARN MyFirstAngularJs No description
npm WARN MyFirstAngularJs No repository field.
npm WARN MyFirstAngularJs No README data
npm WARN MyFirstAngularJs No license field.

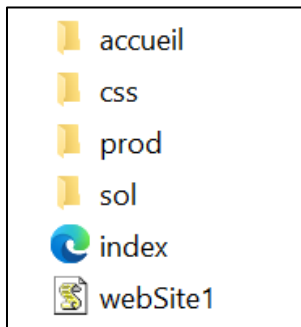
+ angular-route@1.8.0
added 1 package from 1 contributor and audited 1 package in 1.583s
found 0 vulnerabilities
```

** Exemple de route

Après avoir créer l'arborescence pour cette application :



Dans le dossier webSite1 :



Voilà le contenu des différents fichiers que l'on a :

** Fichier **index.html** :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Accueil du site</title>
  <script src="./node_modules/angular/angular.js"></script>
  <script src="./node_modules/angular-route/angular-route.js"></script>
  <script src="webSite1.js"></script>
  <link rel="stylesheet" href="css/monStyle.css">
</head>
<body ng-app="webSite1">
  <header>
    <h1>Mon premier site Angular.Js</h1>
  </header>
  <div class="menu">
    <a href="#!main">Accueil</a>
    <a href="#!produits">Produits</a>
    <a href="#!solutions">Solutions</a>
    <a href="#!tarifs">Tarifs</a>
  </div>
  <ng-view>ici...</ng-view>
  <footer class="menu">
    Mention légale - (c)2018
    <a href="mailto:chb.baudet@gmail.com">CBB</a>
  </footer>
</body>
</html>
```

** Fichier **webSite1.js** :

```
var app = angular.module("website1", ["ngRoute"]);
app.config(['$routeProvider', function ($routeProvider) {
  $routeProvider
    .when("/main", {
      templateUrl: "accueil/main.html"
    })
    .when("/produits", {
      templateUrl: "prod/produits.html"
    })
    .when("/solutions", {
      templateUrl: "sol/solutions.html"
    })
    .otherwise({
      redirectTo: "/main"
    });
}]);
```

** Fichier **monStyle.css** :

```
.menu {
  color: white;
  background: orange;
}

.menu a:link{
  color:white;
  text-decoration: none;
}
.menu a:visited{
  color:white;
  text-decoration: none;
}
.menu a:hover{
  color:black;
  background:yellow;
}
.produits{
  background:DeepSkyBlue;
}
.solutions{
  background:Chartreuse;
}
.main{
  background:Silver;
}
```

** Fichier **main.html** dans le dossier Accueil :

```
<section class="main">
  <h1>Accueil</h1>
  <p>Bienvenue sur notre site anonyme.</p>
  <p>Ici vous ne trouverez aucune indication sur nous, nos produits ou nos solutions</p>
  <br />
  <b>C'est moi qui l'ai pas fait !</b>
</section>
```

** Fichier **produits.html** dans le dossier Prod :

```
<section class="produits">
  <h1>Produits</h1>
  <p>Nos produits sont d'origine 100% naturelle.
  L'herbe des pâturages, l'eau des montagnes filtrée pendant des
  milliers d'années. <b>Bref, le top de la nature</b></p>
</section>
```

** Fichier **solutions.html** dans le dossier Sol :

```
<section class="solutions">
  <h1>Solutions</h1>
  <p>Nos solutions ont déjà permis de nombreuses améliorations
  chez nos -encore plus nombreux- clients.
  <br />
  <b>Pour garder la vie privée (RGPD), il n'y a pas de témoignages ici !</b></p>
</section>
```

À l'affichage du fichier index.html:

↓

Mon premier site Angular.Js

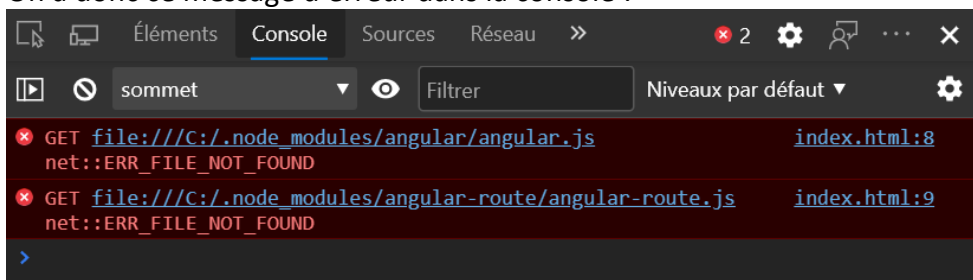
[Accueil](#) [Produits](#) [Solutions](#) [Tarifs](#)

ici...

Mention légale - (c)2018 [CBB](#)

Le navigateur lutte contre les insertions de codes ou de pages d'autres sites :

On a donc ce message d'erreur dans la console :



Problème rencontré :

Dans l'énoncé le CSS est sensé affiché ceci :

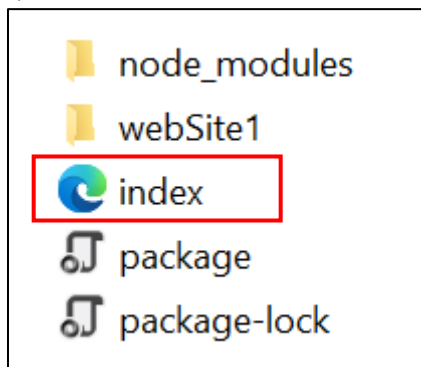


Alors que le mien, le CSS n'a pas l'air de fonctionné
Après plusieurs vérifications du code, je ne vois aucune erreur

Solutions trouvées : Résolu

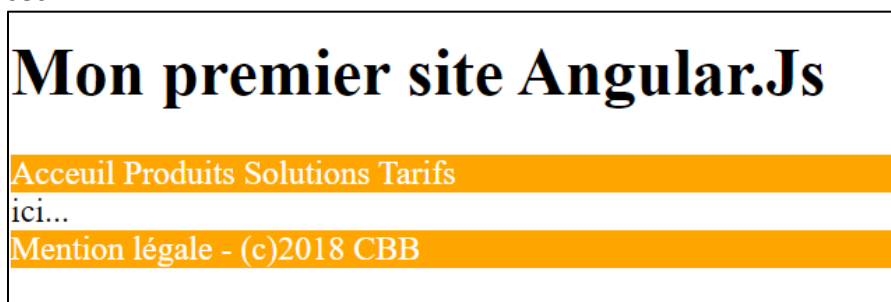
Quand j'ouvrais le fichier index.html dans le dossier webSite1, une copie était créée à l'extérieur de ce dossier (voir screen ci-dessous)

↓

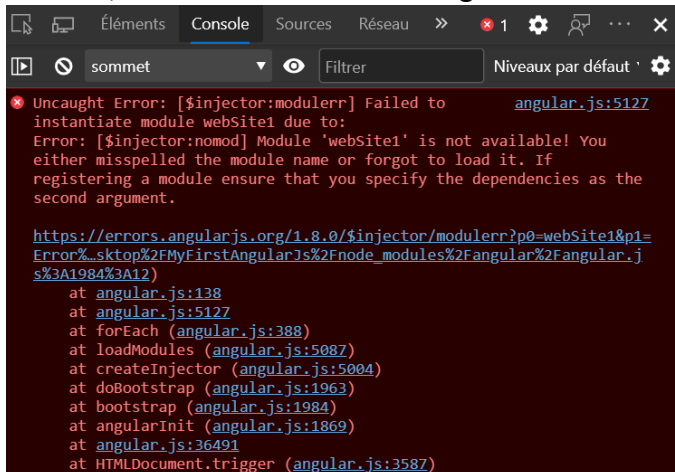


Toutes les modifications que j'ai apporté au script ont été réalisé dans le mauvais fichier index (celui encadré en rouge) et non celui dans le dossier.

Après suppression de ce fichier et réaliser les modifications nécessaires j'obtiens a l'affichage ceci :



Puis dans la console, j'ai remarqué que l'erreur, que j'avais précédemment, n'était pas celle attendu, donc voilà « le bon message d'erreur » à avoir :

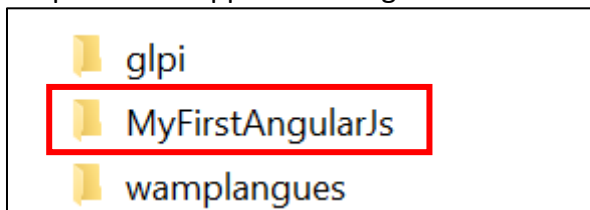


Par rapport au problème précédent (l'erreur dans la console) :

La solution est lancer le fichier dans un server Web

Personnellement j'ai choisi : WAMP Server

J'ai placé mon application Angular.Js dans le dossier /www de wamp :



Dans WampServer, tapez dans le navigateur : « localhost/MyFirstAngularJs/ »



Puis cliquer sur site web, on obtient ceci :

