

Bootstrap avec Angular 15

Guide Complet

07/12/2022

Bootstrap est le plus connu et le plus utilisé des **Frameworks CSS**.

Nous allons intégrer **Bootstrap version 5.2.3** dans notre Application Web.

Nous utiliserons pour cela le framework javascript **Angular version 15.0.2**



Ce que nous allons faire

- **Qu'est-ce que Bootstrap ?**
Nous allons parler de bootstrap et de son utilité dans les sites Web.
- **Création de notre projet Angular**
Nous utiliserons un projet existant contenant les fonctionnalités essentielles.
Le projet a été généré avec Angular CLI et utilise le Routing et le Lazy Loading.
- **Prototype Bootstrap**
Pourquoi et comment créer un prototype Bootstrap ?
- **Version 4 versus Version 5**
Quelle version de bootstrap devriez-vous choisir ?
- **Intégration à notre projet angular**
Comment intégrer Bootstrap en utilisant Angular CLI ?
- **Effectuer les Tests**
Nous testerons notre application via les tests unitaires et end-to-end intégrés dans Angular.
- **Code source**
Pour les plus pressés d'entre vous, le code complet du projet.
<https://github.com/ganatan/angular-react-bootstrap>

Qu'est-ce que Bootstrap?

Les pages d'un **site web** sont écrites en utilisant **3 langages informatiques (computer language en anglais)**

- Le langage **HTML**
HyperText Markup Language (Langage de Balisage HyperText)
il permet de créer et de représenter le contenu d'une page web et sa structure
- Le langage **CSS**
Cascading Style Sheets (Feuilles de style en cascade)
il est utilisé pour décrire la présentation d'une page
- Le langage **Javascript**
Il permet de créer des fonctionnalités interactives dans la page.

De nombreux outils permettent de simplifier la vie d'un développeur.
Notamment les **Frameworks CSS** qui constituent en quelque sorte des boîtes à outils graphiques.

Les **Framework CSS** sont extrêmement nombreux, parmi ceux-ci nous pourrions citer

- Foundation
- Materialize CSS
- Bootstrap
- Pure

Bootstrap est l'un des framework CSS les plus connus et les plus utilisés.

Bootstrap est sous licence **open source** depuis 2011.
Son **utilisation** est donc **gratuite**.

La version actuelle est la **version 5.2.3**

Création du projet

Pour pouvoir continuer ce tutoriel nous devons bien évidemment disposer de certains éléments

- **Node.js** : La plateforme javascript
- **Git** : Le logiciel de gestion de versions.
- **Angular CLI** : L'outil fourni par Angular.
- **Visual Studio code** : Un éditeur de code.

Vous pouvez consulter le tutoriel suivant qui vous explique en détails comment faire

- <https://www.ganatan.com/tutorials/demarrer-avec-angular>

Nous allons utiliser un projet existant dont les caractéristiques sont

- Généré avec Angular CLI
- Routing
- Lazy loading

Vous pouvez installer ce projet sur votre poste de travail en utilisant les commandes suivantes.

```
# Créez un répertoire demo (le nom est ici arbitraire)
mkdir demo

# Allez dans ce répertoire
cd demo

# Récupérez le code source sur votre poste de travail
git clone https://github.com/ganatan/angular-lazy-loading.git

# Allez dans le répertoire qui a été créé
cd angular-lazy-loading

# Exécutez l'installation des dépendances (ou librairies)
npm install

# Exécutez le programme
npm run start

# Vérifiez son fonctionnement en lançant dans votre navigateur la commande
http://localhost:4200/
```

Pourquoi Créer un prototype Bootstrap ?

En simplifiant il y a 3 types de développeur Web

- **Frontend**
- **Backend**
- **Fullstack**

On pourrait rajouter un 4ème métier

- **Web Designer**

Toujours en simplifiant, les **3 premiers** s'occupent de la partie **logique** et le **quatrième** de la partie **graphique**.

C'est pour cela que nous allons créer tout d'abord un prototype graphique. Ce prototype servira de base à la phase de conception de votre site web.

Quelques raisons de ce choix.

- Il est plus difficile et coûteux de faire des changements dans le code plutôt que dans un prototype.
- Si vous êtes amené à travailler en équipe, il est plus facile de savoir qui fait quoi.
- Il existe de nombreux outils de prototypage.
- Il facilite l'évolution et l'amélioration de l'interface.
- Il est plus rapide et facile de présenter un prototype à un éventuel client.

Donc.

*Pour se faire, il n'est pas utile de connaître Angular.
Seule la connaissance de HTML, CSS et d'un peu de javascript sera nécessaire.*

Il sera même possible de **sous-traiter** à un partenaire Web Designer ou graphiste.

Ce sont certains des éléments de ce prototype que nous intégrerons à notre application Angular.

L'architecture de ce prototype et le choix des noms est bien sûr arbitraire. Vous pouvez l'adapter à votre convenance.

Rapide résumé du prototypage ?

La **conception d'un site web** passera par plusieurs étapes.

L'une d'entre elles est le **prototypage**.

Le prototypage consiste à créer un **exemplaire** non définitif de ce que pourra être le **produit final**.

Il permet de concevoir une Interface Utilisateur ou **UI** (User Interface)

Il existe 2 étapes de prototypage.

- **Prototypage horizontal**
Créer une maquette statique.
Dessiner un schéma (wireframe)
Définir les zones et composants
- **Prototypage vertical**
Intégrer les fonctionnalités

Nous allons dans le cadre de ce tutoriel créer simplement une **maquette statique**.

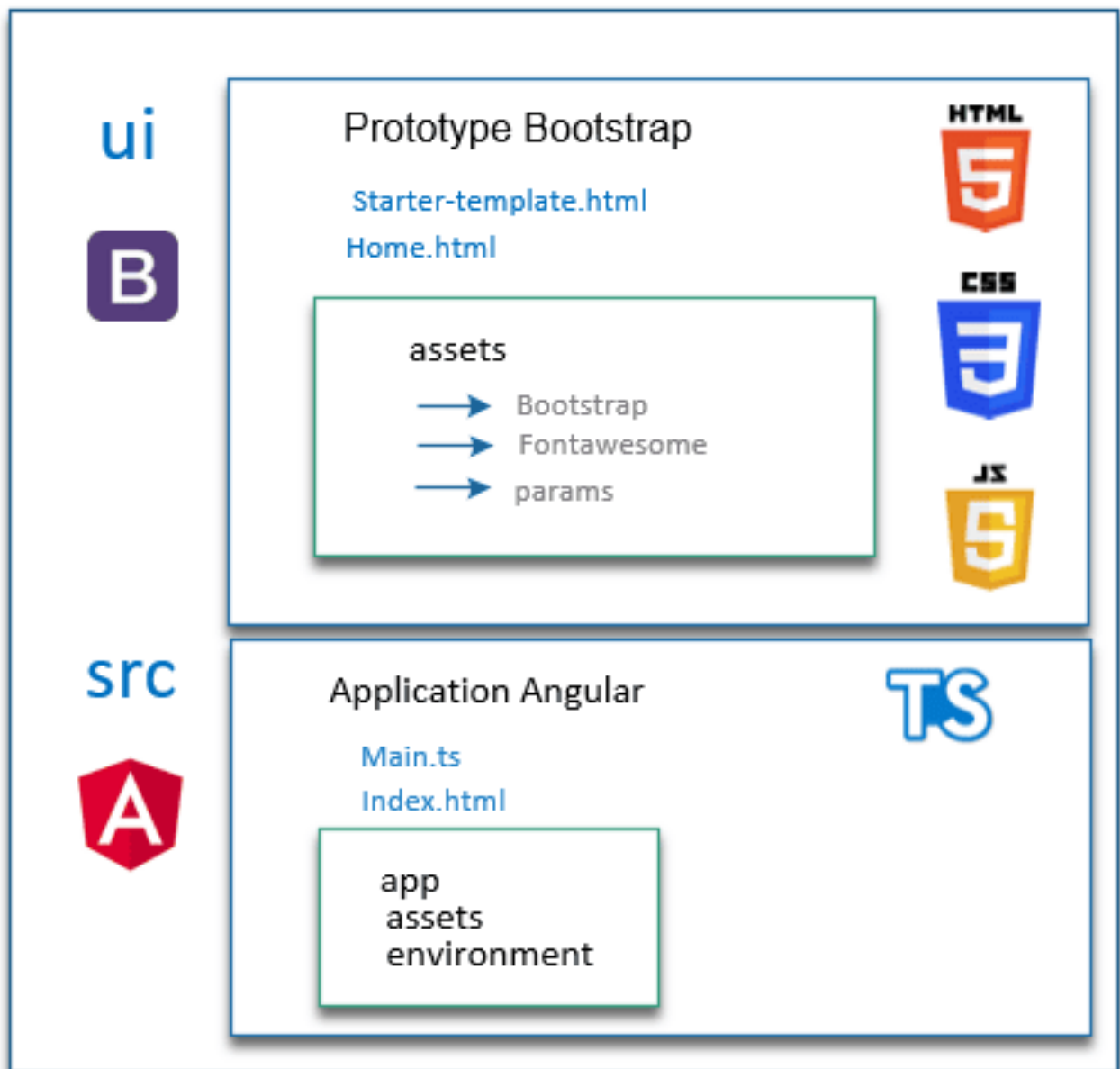
Comment créer un prototype Bootstrap?

Histoire de comprendre ce que l'on fait, nous n'utiliserons aucun outil de prototypage.

Nous concevrons ce prototype **manuellement** et à partir de rien (**from scratch**).

Le répertoire qui contiendra ce prototype sera **UI** ou User Interface (pour Interface Utilisateur).

Une image va nous permettre d'avoir une vue d'ensemble de notre prototype au sein de notre application Angular.



Passons à la pratique.

Nous allons chercher sur le web tous les éléments nécessaires que nous placerons dans l'arborescence de notre projet Angular.

Nous obtiendrons au final l'arborescence suivante.

```
-- node_modules/      (contient les librairies node)
-- src/                (contient le code source de notre application angular)
-- ui/                 (contient notre prototype bootstrap)
  |-- assets
    |-- bootstrap
      |-- css
      |-- js
    |-- fontawesome
      |-- css
      |-- js
      |-- webfonts
    |-- params
      |-- css
      |-- images
      |-- js
package.json
```

Version 4 vs Version 5

Bootstrap 4 c'est bien sympa.
Mais la version 5 est sortie.

Il s'agit de la **version 5.2.3**

Il est intéressant de voir que cette version apporte quelques modifications majeures.

- La suppression de JQuery.
Et par conséquent l'utilisation de vanilla javascript (autrement dit du javascript rien que du javascript)
- L'abandon de la compatibilité avec IE 10 et 11 (Internet Explorer)
C'était pas trop tôt

Le projet final utilisera la version 5, j'ai testé, ça marche, alors pourquoi pas !

Au diable la prudence.

On aura le temps d'être prudent quand on sera mort.

Les éléments du prototype

Donc commençons !

Dans le répertoire **ui** créez un répertoire **assets**.

Dans ce répertoire **assets** créez les 3 répertoires suivants

- **bootstrap**
- **fontawesome**
- **params**

Récupérons maintenant les différents éléments utiles à notre prototype.

Les éléments de bootstrap

- Allez sur le site de bootstrap (partie donwload)
<https://v5.getbootstrap.com/docs/5.2/getting-started/download/>
- Téléchargez les fichiers compilés CSS et JS (**compiled CSS et JS / download**)
- Le fichier qui nous intéresse est **bootstrap-5.2.3-dist.zip**
- Unzippez le fichier bootstrap-5.2.3-dist.zip
- Copiez les répertoires **CSS** et **JS** dans le répertoire **ui / assets / bootstrap**

Les éléments de fontawesome

- Allez sur le site de fontawesome (partie download)
<https://fontawesome.com/download>
- **Download font Awesome free for the web**
Téléchargez les fichiers compilés CSS et JS
- Le fichier qui nous intéresse est **fontawesome-free-6.2.1-web.zip**
- Unzippez le fichier fontawesome-free-6.2.1-web.zip
- Copiez les répertoires **CSS**, **JS** et **webfonts** dans le répertoire **ui / assets / fontawesome**

Remarque

Tous les fichiers installés ne seront pas utilisés.

Nous pouvons dans ce cas faire le ménage et ne laisser que l'essentiel.

*Pour fontawesome nous ne garderons que les fichiers **all.css** , **all.min.css** et **all.js** et **all.min.js***

Vous retrouverez le résultat final sur le dépôt du code source indiqué en fin de tutoriel.

Dans le code source vous retrouverez le répertoire

- **ui** (code bootstrap 5)

Notre première page Bootstrap

Nous allons créer notre première page en utilisant un exemple proposé par bootstrap.

Nous utiliserons Starter template sur la page des exemples

- Version 5
<https://v5.getbootstrap.com/docs/5.2/examples/>

Arbitrairement je vous donne le résultat final d'une page exemple.
Les différentes parties suivent.

Style Body

```
<style>
  body {
    padding-top: 3.5rem;
    font-family: "Roboto", sans-serif;
  }
</style>
```

Style Header

```
<style>
.navbar.navbar-dark .navbar-nav .nav-item .nav-link {
  color: white;
  font-weight: 500;
  border-top: 1px solid #09238d;
  border-bottom: 1px solid #09238d;
}

.navbar.navbar-dark .navbar-nav .nav-item .nav-link:hover {
  color: yellow;
  border-top: 1px solid yellow;
  border-bottom: 1px solid yellow;
}

.nga-navbar {
  -webkit-box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 11px
10px 0 rgba(0, 0, 0, 0.12);
  box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 11px 10px 0
rgba(0, 0, 0, 0.12);
  background-color: #09238d;
}

.nga-logo {
  font-weight: 700;
}

.nga-logo:hover {
  color: rgba(255, 255, 255, 0.75);
}

.nga-btn-navbar {
  color: #fff;
  background-color: #1976d2;
  border-color: #0d6efd;
}

.nga-btn-navbar:hover {
  color: white;
  background-color: #0b5ed7;
  border-color: #0a58ca;
}
</style>
```

Style Footer

```
<style>
.nga-footer {
  background-color: #212121;
  color: white;
}

.nga-footer a {
  color: white;
  text-decoration: none
}

.nga-footer a:hover,
.nga-footer a:focus {
  color: yellow;
  text-decoration: underline;
}

.nga-footer .hint {
  background-color: #1976d2;
}

.nga-footer .hint:hover {
  opacity: 0.8;
}
</style>
```

Style Home

```
<style>
.nga-card {
  display: block;
  background-color: rgba(255, 255, 255, .8);
  box-shadow: 0 1px 3px rgba(0, 0, 0, .12), 0 1px 2px rgba(0, 0, 0, .24);
  border-radius: 2px;
  transition: all .2s ease-in-out;
  cursor: pointer;
}

.nga-card:hover {
  box-shadow: 0 10px 20px rgba(0, 0, 0, .19), 0 6px 6px rgba(0, 0, 0, .23);
}
</style>
```

Html Header

```
<header class="navbar navbar-expand-md navbar-dark fixed-top nga-navbar">
  <nav class="container" aria-label="Main navigation">
    <a href="" class="navbar-brand" alt="Accueil" aria-label="Ganatan">
      
      <span class="nga-logo mx-1">ganatan</span>
    </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse"
      aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav mx-auto">
        <li class="nav-item">
          <a class="nav-link" aria-current="page" href="">
            <i class="fas fa-home me-1"></i>Home</a>
          </li>
        </ul>
        <ul class="navbar-nav me-auto">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="">
              <i class="far fa-question-circle me-1"></i>About</a>
            </li>
          <li class="nav-item">
            <a class="nav-link" aria-current="page" href="">
              <i class="fas fa-envelope me-1"></i>Contact</a>
            </li>
          </ul>
          <form class="d-flex">
            <button type="button" class="btn btn-sm nga-btn-navbar me-2"><i class="fas fa-user-
plus me-2"></i>Sign
              up</button>
            <button type="button" class="btn btn-sm btn-outline-light me-2"><i
              class="fas fa-sign-in-alt me-2"></i>Login</button>
          </form>
        </div>
      </nav>
    </header>
```

Style Main

```
<main>

<div class="container py-5">
  <div class="row">
    <div class="col-12 col-sm-12 col-md-12 col-lg-3 col-xl-3 text-center mb-2">
      <h1 class="h5">
        <i class="fas fa-laptop fa-lg me-2 text-primary"></i>
        angular-starter
        <i class="fas fa-mobile-alt fa-lg ms-2 text-primary"></i>
      </h1>
    </div>
    <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-danger mb-2">
      <h2 class="h5">
        Angular 14.1.1<i class="fab fa-angular fa-lg ms-2"></i>
      </h2>
    </div>
    <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-primary mb-2">
      <h2 class="h5">
        Bootstrap 5.2.0<i class="fab fa-bootstrap fa-lg ms-2"></i>
      </h2>
    </div>
    <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-success mb-2">
      <h2 class="h5">
        Font Awesome 6.1.2<i class="fab fa-font-awesome-flag fa-lg ms-2"></i>
      </h2>
    </div>
  </div>
  <hr>
  <div class="row mb-2">
    <div class="col-md-12 text-center mb-4">
      <h3 class="h5">Features<i class="fas fa-list ms-2"></i></h3>
    </div>
  </div>
</div>

</main>
```

Intégration de bootstrap dans Angular

Passons maintenant à la partie **logique**.

Nous allons faire fonctionner cette page **home** dans **notre projet Angular** de base.

Tout d'abord il nous faut rajouter les librairies nécessaires.

On utilise pour cela **npm** (node package manager) le gestionnaire de dépendances de **Nodes.js**

Pour bootstrap on suivra les conseils sur le site officiel.

<https://v5.getbootstrap.com/docs/5.0/getting-started/download/>

```
# Rajout des dépendances dans package.json
npm install --save bootstrap@next
npm install --save @fortawesome/fontawesome-free
```

Comme nous l'avons vu dans le tutoriel [Démarrer avec angular](#) nous modifierons les descripteurs de version de dépendances.

Concernant les dépendances et leur version la documentation npm est la suivante

<https://docs.npmjs.com/files/package.json#dependencies>

Ce qui nous donnera le résultat suivant.

package.json

```
"@fortawesome/fontawesome-free": "6.1.2",  
"bootstrap": "5.2.0",  
"rxjs": "7.4.0",  
"tslib": "2.3.1",  
"zone.js": "0.11.4"
```

Mise à jour

Réutilisons donc les éléments de notre prototype.

Copiez tout le répertoire **params** de notre prototype dans notre application angular au niveau de **src/assets**

Nous allons modifier le fichier **angular.json** afin d'appeler les fichiers nécessaires au fonctionnement de nos pages html.

Fichiers de mise en forme css

- **index.css** (spécifique à notre projet via params)
- **all.min.css** (spécifique à fontawesome via node_modules)
- **bootstrap.min.css** (spécifique à bootstrap via node_modules)

Fichiers des scripts javascript

- **jquery.min.js** (spécifique à jquery via node_modules)
- **bootstrap.min.js**(spécifique à bootstrap via node_modules)

Remarque

Dans notre cas nous utiliserons le fichier **bootstrap.bundle.min.js**

Nous pourrons alors utiliser les composants Bootstrap de type **Toasts** ,**Tooltips** ou **Popovers**

La documentation est ici

<https://v5.getbootstrap.com/docs/5.2/components/tooltips/>

<https://v5.getbootstrap.com/docs/5.2/components/popovers/>

<https://v5.getbootstrap.com/docs/5.2/components/toasts/>

- **index.js**

Pour rajouter le code spécifique à notre projet ici

Remarque : Dans le cas de la version 5

Je préfère le rajouter sur app.component en fin de chargement de page et éviter les problèmes

Cela évite d'utiliser des fonctions du type document.ready par exemple

angular.json

```
"build": {
  "builder": "@angular-devkit/build-angular:browser",
  "options": {
    "outputPath": "dist/angular-starter",
    "index": "src/index.html",
    "main": "src/main.ts",
    "polyfills": "src/polyfills.ts",
    "tsConfig": "tsconfig.app.json",
    "assets": [
      "src/favicon.ico",
      "src/assets"
    ],
    "styles": [
      "node_modules/@fortawesome/fontawesome-free/css/all.min.css",
      "node_modules/bootstrap/dist/css/bootstrap.min.css",
      "src/assets/params/css/fonts.googleapis.min.css",
      "src/styles.css"
    ],
    "scripts": [
      "node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
    ]
  }
},
```

src/assets/params/css/index.css

```
body {  
  padding-top: 3.5rem;  
  font-family: "Roboto", sans-serif;  
}
```

Nous allons modifier les fichiers suivants qui contiendront la nouvelle interface

- **app.component.html**
- **app.component.css**
- **home.component.html**
- **home.component.ts**
- **home.component.css**
- **environment.prod.ts**
- **environment.ts**
- **app.component.spec.ts**
- **about.css**
- **signin.css**
- **contact.css**

Nous rajouterons les images utilisées dans le répertoire **assets/params/images/logo**

- **ganatan.png**

src/app/app.component.html (Version 4)

```
<header>  
  <nav class="navbar navbar-expand-md fixed-top navbar-dark " style="background-color:  
#212121;">  
    <a class="navbar-brand" routerLink="/">  
      <span class="mr-1" style="color:white">ganatan</span>  
      
  </a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarsExampleDefault"
  aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarsExampleDefault">
    <ul class="navbar-nav mx-auto">
      <li class="nav-item active">
        <a class="nav-link" routerLink="/">
          <i class="fas fa-home mr-1"></i>home
        </a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" routerLink="/about">
          <i class="far fa-question-circle mr-1"></i>About
        </a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" routerLink="/contact">
          <i class="fas fa-envelope mr-1"></i>Contact
        </a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" routerLink="/signin">
          <i class="fas fa-user mr-1"></i>Sign in
        </a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" href="https://github.com/ganatan">
          <i class="fab fa-github mr-1"></i>Github
        </a>
      </li>
    </ul>
  </div>
</nav>
</header>

<main>
  <div class="container-fluid">
    <router-outlet></router-outlet>
  </div>
</main>

<div class="footer">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <p class="text-center text-white">&copy; 2020 - www.ganatan.com</p>
      </div>
    </div>
  </div>
</div>

</div>

```

[src/app/app.component.css](#)

```
.footer {  
  background: #3F729B;  
  padding: 2em;  
}
```

[src/app/modules/general/home/home.component.html](#)

```
<div class="row pt-1 mb-2">  
  <div class="col-md-3 text-center mb-2">  
    <h1 class="h5">  
      <i class="fas fa-laptop fa-lg mr-2 text-primary"></i>  
      {{ name }}  
      <i class="fas fa-mobile-alt fa-lg ml-2 mr-1 text-primary"></i>  
      <i class="fas fa-tablet-alt fa-lg ml-2 mr-1 text-primary"></i>  
    </h1>  
  </div>  
  <div class="col-md-3 text-center text-danger">  
    <h2 class="h5">  
      {{ angular }}<i class="fab fa-angular fa-lg ml-2 mr-1 text-danger"></i>  
    </h2>  
  </div>  
  <div class="col-md-3 text-center text-primary">  
    <h2 class="h5">  
      {{ bootstrap }}<i class="fab fa-bootstrap fa-lg ml-2 mr-1 text-primary"></i>  
    </h2>  
  </div>  
  <div class="col-md-3 text-center text-warning">
```

```

    <h2 class="h5">
      {{ fontawesome }}<i class="fab fa-font-awesome-flag fa-lg ml-2 mr-1 text-warning"></i>
    </h2>
  </div>
</div>
<hr>
<div class="row mb-1">
  <div class="col-md-12 text-center">
    <h3 class="h6">Features&nbsp;&nbsp; <i class="fas fa-list"></i></h3>
  </div>
</div>

<div class="row pt-2">
  <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
    <div class="card bg-light mb-3">
      <a routerLink="/bootstrap-prototype">
        <div class="card-header">
          <div class="row">
            <div class="col-10 col-xl-10">
              <h5 class="card-title">Bootstrap Prototype</h5>
            </div>
            <div class="col-2 col-xl-2">
              <i class="fab fa-bootstrap fa-lg text-primary"></i>
            </div>
          </div>
        </div>
        <div class="card-body">
          <p class="card-text">Bootstrap Prototype Description</p>
        </div>
      </a>
    </div>
  </div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
    <div class="card bg-light mb-3">
      <div class="card-header">
        <div class="row">
          <div class="col-10 col-xl-10">
            <h5 class="card-title">Components</h5>
          </div>
          <div class="col-2 col-xl-2">
            <i class="fas fa-share-alt-square fa-lg text-primary"></i>
          </div>
        </div>
      </div>
      <div class="card-body">
        <p class="card-text">Components Description</p>
      </div>
    </div>
  </div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
    <div class="card bg-light mb-3">
      <div class="card-header">
        <div class="row">
          <div class="col-10 col-xl-10">
            <h5 class="card-title">Services</h5>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <div class="col-2 col-xl-2">
          <i class="far fa-address-card fa-lg text-primary"></i>
        </div>
      </div>
    </div>
    <div class="card-body">
      <p class="card-text">Services Description</p>
    </div>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
  <div class="card bg-light mb-3">
    <div class="card-header">
      <div class="row">
        <div class="col-10 col-xl-10">
          <h5 class="card-title">Reactive Form</h5>
        </div>
        <div class="col-2 col-xl-2">
          <i class="far fa-address-card fa-lg text-primary"></i>
        </div>
      </div>
    </div>
    <div class="card-body">
      <p class="card-text">Reactive Form Description</p>
    </div>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
  <div class="card bg-light mb-3">
    <div class="card-header">
      <div class="row">
        <div class="col-10 col-xl-10">
          <h5 class="card-title">Template Driven Forms</h5>
        </div>
        <div class="col-2 col-xl-2">
          <i class="far fa-address-card fa-lg text-primary"></i>
        </div>
      </div>
    </div>
    <div class="card-body">
      <p class="card-text">Template Driven Forms Description</p>
    </div>
  </div>
</div>
</div>
</div>

```

```

import { Component, OnInit } from '@angular/core';
import { environment } from '../../environments/environment';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  name = environment.application.name;
  angular = environment.application.angular;
  bootstrap = environment.application.bootstrap;
  fontawesome = environment.application.fontawesome;

  constructor() { }

  ngOnInit(): void {
  }

}

```

src/app/modules/general/home/home.component.ts (Version 5)

```

import { Component, OnInit } from '@angular/core';
import { environment } from '../../environments/environment';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  name = environment.application.name;
  angular = environment.application.angular;
  bootstrap = environment.application.bootstrap;
  fontawesome = environment.application.fontawesome;

  constructor() { }

  ngOnInit(): void {
    this.loadScript('assets/params/js/index.js');
  }

  loadScript(name: string): void {
    const s = document.createElement('script');
    s.type = 'text/javascript';
  }
}

```


src/app/modules/general/home/home.component.css

```
.card {  
  display: block;  
  background-color: rgba(255, 255, 255, .8);  
  box-shadow: 0 1px 3px rgba(0, 0, 0, .12), 0 1px 2px rgba(0, 0, 0, .24);  
  border-radius: 2px;  
  transition: all .2s ease-in-out;  
  cursor: pointer;  
}  
  
.card:hover {  
  box-shadow: 0 10px 20px rgba(0, 0, 0, .19), 0 6px 6px rgba(0, 0, 0, .23);  
}
```

src/environments/environments.ts

```
export const environment = {  
  production: false,  
  application:  
  {  
    name: 'angular-starter',  
    angular: 'Angular 12.0.0',  
    bootstrap: 'Bootstrap 5.0.1',  
    fontawesome: 'Font Awesome 5.15.3'.  
  }  
}
```

src/environments/environments.prod.ts

```
export const environment = {  
  production: true,  
  application:  
  {  
    name: 'angular-starter',  
    angular: 'Angular 12.0.0',  
    bootstrap: 'Bootstrap 5.0.1',  
    fontawesome: 'Font Awesome 5.15.3',  
  }  
};
```

src/app/app.component.spec.ts

```
import { TestBed } from '@angular/core/testing';  
import { AppComponent } from './app.component';  
import { RouterTestingModule } from '@angular/router/testing';  
  
describe('AppComponent', () => {  
  beforeEach(async () => {  
    await TestBed.configureTestingModule({  
      imports: [  
        RouterTestingModule  
      ],  
      declarations: [  
        AppComponent  
      ],  
    }).compileComponents();  
  });  
  
  it('should create the app', () => {  
    const fixture = TestBed.createComponent(AppComponent);  
    const app = fixture.componentInstance;  
    expect(app).toBeTruthy();  
  });  
  
  it('should have as title \'angular-starter\'', () => {  
    const fixture = TestBed.createComponent(AppComponent);  
    const app = fixture.componentInstance;  
    expect(app.title).toEqual('angular-starter');  
  });  
});
```

src/app/modules/general/about/about.html

```
<p class="text-center">about works!</p>
```

src/app/modules/general/contact/contact.html

```
<div class="row">
  <div class="col-12 col-sm-12 col-md-3 col-lg-3 col-xl-3">
    <p class="text-center">contact works!</p>
    <ul>
      <li><a routerLink="/contact/mailing">Mailing</a></li>
      <li><a routerLink="/contact/map">Map</a></li>
    </ul>
  </div>
  <div class="col-12 col-sm-12 col-md-9 col-lg-9 col-xl-9">
    <router-outlet></router-outlet>
  </div>
</div>
```

src/app/modules/general/signin/signin.html

```
<p class="text-center">signin works!</p>
```

src/app/modules/general/not-found/not-found.html

```
<p class="text-center">not-found works!</p>
```

Composants

Le projet final intègre un module permettant de tester les composants de Bootstrap.

Le code étant trop long je ne l'indique pas dans ce tutoriel.
Néanmoins le source complet est disponible sur Github.

Il est rajouté via le routing dans les fichiers suivants

```
{
  path: 'bootstrap-prototype',
  loadChildren: () => import('./modules/application/example-bootstrap-prototype/example-
bootstrap-prototype.module')
    .then(mod => mod.ExampleBootstrapPrototypeModule)
},
```

```
<div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
  <div class="card bg-light mb-3">
    <a routerLink="/bootstrap-prototype">
      <div class="card-header">
        <div class="row">
          <div class="col-10 col-xl-10">
            <h5 class="card-title">Bootstrap Prototype</h5>
          </div>
          <div class="col-2 col-xl-2">
            <i class="fab fa-bootstrap fa-lg text-primary"></i>
          </div>
        </div>
      </div>
      <div class="card-body">
        <p class="card-text">Bootstrap Prototype Description</p>
      </div>
    </a>
```

Tests

Il ne reste plus qu'à tester les scripts suivants.

```
# Développement
npm run start
http://localhost:4200/

# Test de la page not-found
http://localhost:4200/lien-inconnu

# Tests
npm run lint
npm run test
npm run e2e

# Production
npm run build
```

Code source

En suivant chacun des conseils que je vous ai donnés dans ce guide vous obtenez au final un code source Angular.

Le code source obtenu à **la fin de ce tutoriel** est disponible sur github <https://github.com/ganatan/angular-react-bootstrap>

L' étape suivante va nous permettre d'améliorer notre structure.

Le tutoriel complet est à l'adresse suivante

- [Etape 5 : Modules avec Angular](#)

Les étapes suivantes vous permettront **d'obtenir une application prototype**

- [Etape 1 : Démarrer avec Angular](#)
- [Etape 2 : Routing avec Angular](#)
- [Etape 3 : Lazy loading avec Angular](#)
- [Etape 4 : Bootstrap avec Angular](#)
- [Etape 5 : Modules avec Angular](#)
- [Etape 6 : Components avec Angular](#)
- [Etape 7 : Services avec Angular](#)
- [Etape 8 : Template Driven Forms avec Angular](#)
- [Etape 9 : Charts avec Angular](#)
- [Etape 10 : Server Side Rendering avec angular](#)
- [Etape 11 : HttpClient avec Angular](#)
- [Etape 12 : Transfer State avec Angular](#)
- [Etape 13 : Progressive Web App avec Angular](#)
- [Etape 14 : Search Engine Optimization avec Angular](#)

La dernière étape permet **d'obtenir un exemple d'application**

- [Etape 15 : Créer une application Web complète avec Angular](#)

*Le **code source de cette application finale** est disponible sur GitHub*
<https://github.com/ganatan/angular-app>