

Atelier N°3 : Initiation MVC

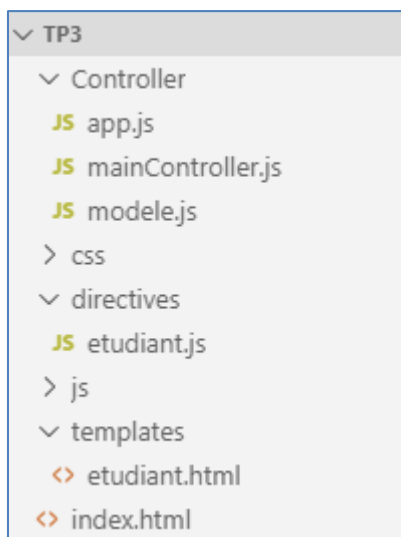
OBJECTIFS :

- Construire des modules et des contrôleurs dans AngularJS.
- Connaître le scope du contrôleur, le principe d'un template, le rôle d'une directive.

On veut créer une application AngularJS basée sur l'architecture MVC permettant dans une page HTML5 (la vue) d'afficher la liste des étudiants présents dans le modèle et d'y ajouter de nouveaux étudiants.

Ajouter un nouvel étudiant	Liste des Etudiants
Nom : <input type="text"/>	DHAHRI
Prénom : <input type="text"/>	Mbarek
<input type="button" value="Ajouter"/> <input type="button" value="Annuler"/>	

L'arborescence de notre application sera comme suit :



- Le fichier **app.js** définit les module Angular de notre application.
- Le fichier **mainController** contient le contrôleur permettant de lier la vue et le modèle.
- Le fichier **modele.js** définit le modèle.
- Le dossier **directives** sera utilisé pour créer de nouvelles balises (directives) à utiliser dans la vue.
- Le dossier **js** contient les frameworks AngularJS et Bootstrap.
- Le dossier **templates** sera utilisé pour stocker les consignes d'affichage des directives.

1. Le modèle

Le modèle sera créé dans le fichier « **modele.js** ». on créera un service qu'on peut l'appeler partout dans l'application. Ce service est créé grâce à la fonction « **factory()** » du framework.

Le service expose deux fonctions qu'on peut les appeler :

- **getEtudiants** qui renvoie tous les étudiants du modèle ;
- **addEtudiant** qui ajoute un étudiant au modèle

```
1  angular.module("app").factory("etudiantsFactory", function ()
2  {
3      // initialiser le modèle
4      var etudiants = [{id:1, nom: 'Dhahri', prenom: 'Mbarek'}];
5      // cette fonction retourne la liste des étudiants
6      var getEtudiants = function()
7      {
8          return etudiants;
9      };
10     // Cette fonction permet d'ajouter un étudiant au modèle
11     var addEtudiant = function(etudiant)
12     {
13         var etudiant = prepareEtudiant(etudiant);
14         etudiants.push ({id:etudiant.id, nom:etudiant.nom, prenom:etudiant.prenom});
15     };
16     function prepareEtudiant(etudiant)
17     {
18         etudiant.id = etudiants.length + 1;
19         return etudiant;
20     }
21     // les fonctions que le service expose et qu'on peut les appeler
22     return {
23         getEtudiants: getEtudiants,
24         addEtudiant: addEtudiant
25     };
26 });
```

2. Le contrôleur

Le contrôleur **mainController** à créer rend deux paramètres d'entrée :

- **\$scope** : est le mécanisme permettant d'exposer les données aux vues. \$scope est passé en paramètre de chaque contrôleur, toute propriété ou fonction y étant ajoutée sera accessible dans la vue.
- Le service déjà défini.

```
1  angular.module("app").controller("mainController", function ($scope, etudiantsFactory)
2  {
3      // définit la liste des étudiants
4      $scope.etudiants = etudiantsFactory.getEtudiants();
5      // permet d'ajouter un étudiant à la liste des étudiants
6      $scope.addEtudiant = function(etudiant)
7      {
8          etudiantsFactory.addEtudiant(etudiant);
9          $scope.newEtudiant.nom = '';
10         $scope.newEtudiant.prenom = '';
11     }
12 });
```

3. Les directives

La directive dans sa forme la plus simple, est un morceau de template HTML, utilisé de préférence à plusieurs endroits de l'application. C'est un moyen d'injecter du DOM dans l'application.

```
1 // créer une nouvelle balise etudiant
2 angular.module("app").directive("etudiant", function ()
3 {
4     return {
5         restrict: 'E',
6         templateUrl: 'templates/etudiant.html'
7     };
8 });
```

Le template représente le modèle HTML qu'on souhaite appliquer à cette directive.

```
1 <h4> {{ etudiant.nom | uppercase }} </h4>
2 |   {{ etudiant.prenom }}
3 <hr>
```

4. La vue

La vue est constitué d'une page HTML5 « **index.html** ».

Pour pouvoir afficher une donnée du modèle dans la vue, il faut passer par du **binding**.

Le principe du binding est de synchroniser une donnée du modèle et l'affichage de cette donnée. Grâce à ce binding et à l'attribut **ng-model**, la vue et le modèle sont synchronisés, que les modifications sont effectuées dans la vue ou dans le contrôleur.

```
1 <!DOCTYPE html>
2 <html ng-app="app">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Inscriptions </title>
8     <link rel="stylesheet" href="css/bootstrap.min.css">
9     <script src="js/bootstrap.min.js"></script>
10    <script src="js/angular.min.js"></script>
11    <script src="Controller/app.js"></script>
12    <script src="Controller/modele.js"></script>
13    <script src="controller/mainController.js"></script>
14    <script src="directives/etudiant.js"></script>
15 </head>
```

Formulaire pour mettre à jour le modèle :

L'attribut `_ng-click` permet de réagir à l'évènement clic du bouton. Cet attribut attend comme valeur une fonction du contrôleur à appeler lors du déclenchement de l'évènement.

```

16 <body>
17   <section class="container" ng-controller="mainController">
18     <div class="row">
19       <div class="col-md-6">
20         <form name="etudiantForm" class="form">
21           <h2>Ajouter un nouvel étudiant</h2>
22           <hr>
23           <div class="form-group">
24             <label for="nom">Nom :</label>
25             <input id="nom" type="text" class="input-sm form-control " ng-model="newEtudiant.nom">
26           </div>
27           <div class="form-group">
28             <label for="prenom">Prénom :</label>
29             <input id="prenom" type="text" class="form-control" ng-model="newEtudiant.prenom">
30           </div>
31           <div class="form-group">
32             <button class="btn btn-primary" type="submit" ng-click="addEtudiant(newEtudiant)"> Ajouter
33             <button class="btn btn-primary" type="reset"> Annuler </button>
34           </div>
35         </form>
36       </div>

```

Affichage de la liste des étudiants

```

<div class="col-md-6">
  <h2>Liste des Etudiants</h2>
  <hr>
  <article ng-repeat="etudiant in etudiants">
    <etudiant />
  </article>
</div>

```