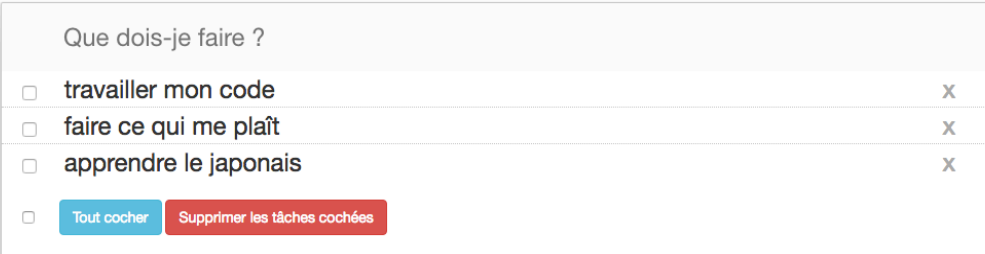


Sébastien Lafontaine  
Version 1.0.1  
07/02/2018  
AngularJS

## 1° Les préparatifs avant la fête

Bonjour à tous, aujourd'hui on se retrouve pour un petit tutoriel simple et rapide sur AngularJS. Celui-ci va nous permettre de créer un petit pense bête qui s'exécutera dans votre navigateur web.

### Liste des choses à faire



Que dois-je faire ?	
<input type="checkbox"/> travailler mon code	x
<input type="checkbox"/> faire ce qui me plaît	x
<input type="checkbox"/> apprendre le japonais	x
<input type="checkbox"/> Tout cocher	Supprimer les tâches cochées

Pour pouvoir arriver à ce résultat rassurez-vous il n'y aura rien de bien compliqué. Tout d'abord nous allons structurer notre page **HTML** qui va contenir notre controller, notre style **CSS** avec du **BOOTSTRAP** et l'**ANGULAR**.

Pour bien commencer il vous faudra créer un petit dossier pour votre projet dans lequel vous aller ajouter 3 autres sous dossier et votre index.html (que nous verrons plus tard).  
**assets** : dans lequel vous mettrez toutes les ressources externes (images, vidéos, etc...) ainsi que l'angular.

**controllers** : qui comme son nom l'indique stockera nos controller.

**styles** : dans lequel nous déposerons le css, ce css

<http://demo.tutoriel-angularjs.fr/uploads/todoList/web/css/style.css>

Voilà nous avons notre petit dossier prêt à être rempli.

## 2° Les framework ou la magie d'être aidé

Pour commencer à insérer le code venant de nos gentils amis framework, nous allons devoir leur aménager un hangar. Cette zone de dépôt c'est notre fameux index.html.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Pensesy</title>
    <link rel="stylesheet" href="styles/style.css">
    <link rel="icon" type="image/png" sizes="96x96" href="assets/images/icon.png">
    <script src="assets/angular.js"></script>
    <script src="controllers/todoListCtrl.js"></script>
  </head>
  <body>
    <header>
      <h1>Liste des choses à faire</h1>
    </header>
  </body>
</html>
```

Voilà à quoi ressemble notre fichier index html, il est composé de :

**<meta charset="utf-8" />** : balise qui s'occupe des syntaxes spécifiques (é, è).

**<title>Pensesy</title>** : nos balises titre que vous pouvez bien sûr personifier.

**<link rel="stylesheet" href="styles/style.css">** : notre style css avec le même chemin d'accès que je vous ai spécifié auparavant.

**<link rel="icon" type="image/png" sizes="96x96" href="assets/images/icon.png">** : la mise en place d'un icône pour notre onglet de navigateur web.

**<script src="assets/angular.js"></script>** : notre fameux angularJS que vous allez aller chercher à cette adresse

<https://ajax.googleapis.com/ajax/libs/angularjs/1.3.12/angular.js>

et copier/coller dans votre dossier.

**<script src="controllers/todoListCtrl.js"></script>** : et nous voilà rendu avec notre controller qui nous intéressera très vite.

## 3° Un controller pour les contrôler tous

Tiens c'était rapide ça ! Alors on commence doucement par se rendre dans le controller que nous avons créer et on va enfin pouvoir lui placer quelques lignes.

```
// js/todoList.js
'use strict';

/**
 * Déclaration de l'application demoApp
 */
var demoApp = angular.module('demoApp', [
  // Dépendances du "module"
  'todoList'
]);

/**
 * Déclaration du module todoList
 */
var todoList = angular.module('todoList', []);
```

Alors quelques points de détaille sur ce que nous avons fait :

**use strict'** : le strict permet d'éviter les variables globales en javascript qui sont fortement déconseillées.

**var demoApp = angular.module('demoApp', ['todoList']);** : ici une déclaration de notre application 'demoApp' que vous pouvez bien entendu modifier si vous le souhaitez.

#### 4° Joindre les deux bouts

Ici aucune raison d'avoir du mal à quoi que se soit. Nous allons retourner dans notre hangar html afin de spécifier où garer nos modules.

```
<section ng-controller="todoCtrl">
  <form id="todo-form" ng-submit="addTodo()">
    <input id="new-todo" placeholder="Que dois-je faire ?" ng-model="newTodo" />
  </form>
</section>
```

Et voilà une magnifique place pour notre controller.

Grâce à notre balise **<section>** nous avons ciblé la place dans laquelle notre controller allait agir. Puis nous allons ajouter au fur et à mesure les différents éléments qui nous intéressent.

Par exemple un `<form>` pour avoir notre zone pour écrire les nombreuses choses (ou pas) que nous aurons à faire.

`ng-model="newTodo"` : ajouter des variables dans le scope qui lui est associé ici il est question du `ng-controller='todoCtrl'`.

`ng-submit="addTodo()` : déclenche une fonction lorsque la condition est remplie, ici il s'agit d'appuyer sur entrer pour ajouter notre chose à faire.

Ensuite toujours dans les balises `<section>` nous allons insérer un `<article>` comme ci-dessous.

```
<article ng-show="todos.length">
  <ul id="todo-list">
    <li ng-repeat="todo in todos" ng-class="{completed: todo.completed}">
      <div class="view">
        <input class="mark" type="checkbox" ng-model="todo.completed" />
        <span>{{todo.title}}</span>
        <span class="close" ng-click="removeTodo(todo)">x</span>
      </div>
    </li>
  </ul>
</article>
```

`ng-class="{completed: todo.completed}"` : cela nous permet d'attribuer une classe à une balise selon une certaine condition. Ici il est question de la balise `<li>` qui obtient le `todo.completed == true`.

`ng-click="removeTodo(todo)"` : similaire au `onclick` de HTML il permet d'exécuter une fonction après un clique. C'est pas plus compliqué.

`ng-show="todos.length"` : elle, elle s'occupe d'afficher un élément si et seulement si la condition `todos.length == true`. En d'autres termes `<article>` ne s'affiche que lorsqu'il y a une chose à faire au moins.

Puis pour conclure avec le html, on insère les différents boutons pour, cocher, supprimer les tâches déjà effectuées.

```
<input class="mark" type="checkbox" ng-model="todo.completed" />
<span>{{todo.title}}</span>
<span class="close" ng-click="removeTodo(todo)">x</span>
```

Et voilà une petite vue d'ensemble de notre index.html, mais c'est de toute beauté !

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Pensesy</title>
    <link rel="stylesheet" href="styles/style.css">
    <link rel="icon" type="image/png" sizes="96x96" href="assets/images/icon.png">
    <script src="assets/angular.js"></script>
    <script src="controllers/todoListCtrl.js"></script>
  </head>
  <body>
    <header>
      <h1>Liste des choses à faire</h1>
    </header>
    <section ng-controller="todoCtrl">
      <form id="todo-form" ng-submit="addTodo()">
        <input id="new-todo" placeholder="Que dois-je faire ?" ng-model="newTodo" />
      </form>
      <article ng-show="todos.length">
        <ul id="todo-list">
          <li ng-repeat="todo in todos" ng-class="{completed: todo.completed}">
            <div class="view">
              <input class="mark" type="checkbox" ng-model="todo.completed" />
              <span>{{todo.title}}</span>
              <span class="close" ng-click="removeTodo(todo)">x</span>
            </div>
          </li>
        </ul>
      </article>
    </section>
  </body>
</html>
```

## 5° Retour vers le controller Marty !

```
/**
 * Contrôleur de l'application "Todo List" décrite dans le chapitre "La logique d'AngularJS".
 */
todoList.controller('todoCtrl', ['$scope',
  function ($scope) {

    // Pour manipuler plus simplement les todos au sein du contrôleur
    // On initialise les todos avec un tableau vide : []
    var todos = $scope.todos = [];
```

Voilà pour un petit début dans le controller et toujours dans notre 'todoList' nous allons ajouter le code qui suit.

```

// Ajouter un todo
$scope.addTo = function () {
  // .trim() permet de supprimer les espaces inutiles
  // en début et fin d'une chaîne de caractères
  var newTodo = $scope.newTodo.trim();
  if (!newTodo.length) {
    // éviter les todos vides
    return;
  }
  todos.push({
    // on ajoute le todo au tableau des todos
    title: newTodo,
    completed: false
  });
  // Réinitialisation de la variable newTodo
  $scope.newTodo = '';
};

```

**`$scope.addTo = function () {`** : Je déclare ma fonction comme une des propriétés du scope, ce qui permettra à AngularJS de la retrouver lorsqu'elle sera appelée.

**`todos.push({`** : On ajoute un 'todo' au tableau des 'todos'. Il s'agrandit à chaque entrée de valeur.

« Et attention en exclusivité live this is .... **TWO-WAY DATA BINDING** (son de trompette) »

Tout ça pour simplement vous dire que l'on peut lire et modifier les variables qui sont dans le scope.

**`var newTodo = $scope.newTodo.trim();`** : on lit d'abord puis on crée le nouveau 'todo'.

**`$scope.newTodo = '';`** : et là paf, on la modifie pour remettre à zéro le champ.

```
// Enlever un todo
$scope.removeTodo = function (todo) {
  todos.splice(todos.indexOf(todo), 1);
};

// Cocher / Décocher tous les todos
$scope.markAll = function (completed) {
  todos.forEach(function (todo) {
    todo.completed = completed;
  });
};

// Enlever tous les todos cochés
$scope.clearCompletedTodos = function () {
  $scope.todos = todos = todos.filter(function (todo) {
    return !todo.completed;
  });
};
```

**`$scope.markAll = function (completed) {`** : cela correspond au bouton 'Tout cocher'. On place donc **`completed`** pour chaque 'todo' ayant la même valeur que celle de la checkbox globale. Mais, mais mais mais mais il y a toujours un mais. Il faut bien faire attention à bien donner la valeur opposé avec le **`!`**, car lorsque la fonction est appelée la checkbox n'a pas encore changé.

**`$scope.clearCompletedTodos = function () {`** : on utilise le **`.filter()`** pour ne récupérer que nos 'todos' qui ne sont pas cochés. Ensuite on écrase l'ancien tableau avec le résultat de filter.

Tada ! Et voilà un petit aperçu final de notre partie controller.

```
// js/todoList.js
'use strict';

/**
 * Déclaration de l'application demoApp
 */
var demoApp = angular.module('demoApp', [
  // Dépendances du "module"
  'todoList'
]);

/**
 * Déclaration du module todoList
 */
var todoList = angular.module('todoList', []);

/**
 * Contrôleur de l'application "Todo List" décrite dans le chapitre "La logique d'AngularJS".
 */
todoList.controller('todoCtrl', ['$scope',
  function ($scope) {

    // Pour manipuler plus simplement les todos au sein du contrôleur
    // On initialise les todos avec un tableau vide : []
    var todos = $scope.todos = [];

    // Ajouter un todo
    $scope.addTodo = function () {
      // .trim() permet de supprimer les espaces inutiles
      // en début et fin d'une chaîne de caractères
      var newTodo = $scope.newTodo.trim();
      if (!newTodo.length) {
        // éviter les todos vides
        return;
      }
      todos.push({
        // on ajoute le todo au tableau des todos
        title: newTodo,
        completed: false
      });
      // Réinitialisation de la variable newTodo
      $scope.newTodo = '';
    };

    // Enlever un todo
    $scope.removeTodo = function (todo) {
      todos.splice(todos.indexOf(todo), 1);
    };

    // Cocher / Décocher tous les todos
    $scope.markAll = function (completed) {
      todos.forEach(function (todo) {
        todo.completed = completed;
      });
    };

    // Enlever tous les todos cochés
    $scope.clearCompletedTodos = function () {
      $scope.todos = todos = todos.filter(function (todo) {
        return !todo.completed;
      });
    };
  }
]);
});
```



---

Alors qu'est ce que je vous avais dit rien de bien compliqué (surtout pour ceux qui ont copier/coller tout le code). Je sais tout, pas la peine de vous cacher dans la rangé du fond... Et je vous vois là !