



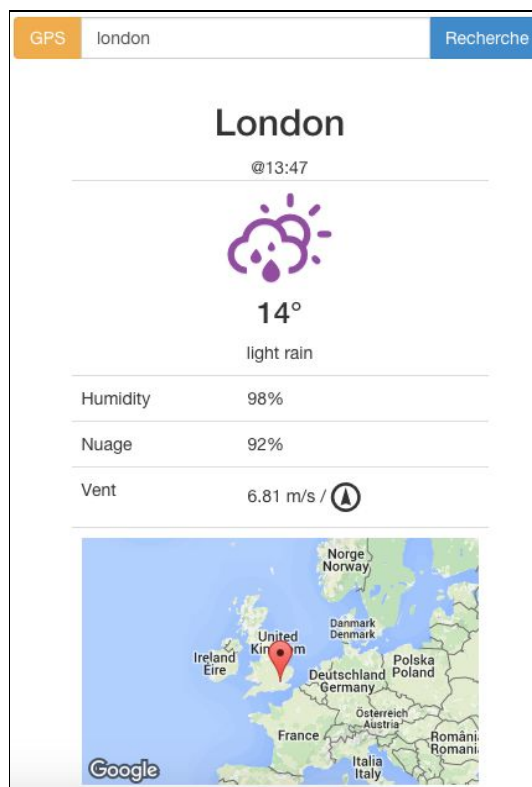
Master 3IR TP AW #6 - Angular

Développement d'une application météo avec AngularJS & OpenWeatherMap.

Site officiel <https://angularjs.org/>, tutoriel tutorials.jenkov.com/angularjs/index.html

Utilisation des données météo de OpenWeatherMap <http://openweathermap.org/>

Détail de l'API : <http://openweathermap.org/current>



Affichage de la météo pour Londres

1. Plate-forme de développement :

- 1.1. Éditeur de texte (visual studio code, notepad, eclipse...)
- 1.2. Navigateur web (de préférence Mozilla Firefox)
- 1.3. En option :
 - 1.3.1. utiliser le serveur web [pseudo].github.io de votre compte github
 - 1.3.2. ou installer un serveur sur votre poste (tomcat ou autre serveur web)

2. Application web avec **AngularJS**

2.1.1. Dans un dossier TP_ANGULAR, créer un fichier index.html

Dans cet exemple nous allons utiliser les librairies depuis leur site web

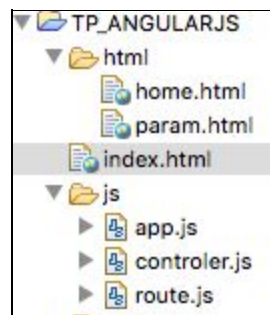
```
<!DOCTYPE html>
<html ng-app="meteoApp">
  <head>
    <title>Weather APP</title>
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
    <script src="https://code.angularjs.org/1.4.7/angular-route.js"></script>

    <script src="js/app.js"></script>
    <script src="js/route.js"></script>
    <script src="js/controler.js"></script>
  </head>
  <body>
    <div ng-view></div>

    <script type="text/ng-template" id="html/homeFailOver.html">...</script>
  </div>
</body>
</html>
```

index.html

2.1.2. A la fin du TP, l'arborescence du projet sera



2.2. Fichier js/app.js

L'initialisation d'une application AngularJS se fait via le script suivant dans le fichier app.js

```
/* App Module */
var meteoApp = angular.module('meteoApp', ['ngRoute', 'meteoControllers']);
```

app.js

2.3. Fichier js/route.js

La gestion de la navigation (changement de page/URL) se fait via le script **route.js**. Pour chaque page, il faut lui associer un template HTML et un contrôleur qui va mettre à jour le contenu de la template.

Pour ce TP, on ne déclarera que la page *home.html* et son controleur *MainController*.

```

meteoApp.config(['$routeProvider',
function($routeProvider) {
    /* URL associée : http://127.0.0.1:8080/#/home */
    $routeProvider.when('/home', {
        controller : 'MainController',    templateUrl : 'html/home.html'
    })
    /* URL associée : http://127.0.0.1:8080/#/param, code mis en commentaire en attendant le TP6 */
    /*.when('/param', {
        controller : 'ParamController',    templateUrl : 'html/param.html'
    })*
    /* Si l'URL est inconnue, on redirige vers la page #/home */
    .otherwise({
        redirectTo : '/home'
    });
});

```

route.js

2.4. Fichier template HTML : html/home.html

Basez vous sur bootstrap pour développer un champ de recherche avec un bouton de recherche et un bouton de géolocalisation GPS

Le champ de saisie “city” aura un attribut **ng-model** permettant à Angular de récupérer/mettre à jour la valeur du champ.

```
<input type="search" ng-model="city" placeholder="City" class="form-control"/>
```

- > Sur bouton “Recherche” ajouter l’attribut **ng-click="recherche()"**
- > Sur bouton “GPS” ajouter l’attribut **ng-click="gps()"**
- > Dans le contrôleur JS, vous définirez le comportement de chacune de ces fonctions.

2.5. Fichier js/controler.js

Dans ce fichier javascript, définissez le comportement de la fonction recherche() dans le 'MainController':

Pour appeler un webservice en AJAX avec AngularJS, utiliser [\\$http.get](#)

Tutoriel : <http://tutorials.jenkov.com/angularjs/ajax.html>

L’API météo est appelé à cette URL :

[https://demo.bilez.fr/owmap/?q=paris&appid=\[VOTRE_API_KEY\]](https://demo.bilez.fr/owmap/?q=paris&appid=[VOTRE_API_KEY])

Elle retournera les données au format JSON (objet javascript)

Détail des données : http://openweathermap.org/current#current_JSON

- > Écrire la fonction gps(), qui demandera la géolocalisation de l’utilisateur et affichera la météo relative à sa position.

```

var meteoControllers = angular.module('meteoControllers', []);

meteoControllers.controller('MainController', ['$scope', '$http',
function($scope, $http) {

    $scope.recherche = function() {

/* appel AJAX à l'API openweathermap */
$http.get('https://demo.bilelz.fr/owmap/?q='+$scope.city+'&units=metric&appid=[VOTRE_API_KEY]')
    .success(function(data) {
        /* on met dans l'objet meteo les données retournées
        par openweathermap */
        $scope.meteo = data;
    }).error(function(data) {
        /* en cas d'erreur */
        $scope.errorMsg = "Hum. Error... please retry.";
    });

    }

});

```

controler.js

2.6. Affichage des informations d'openweathermap dans le template html/home.html

Dans le template **home.html**, mettre en forme les informations reçues par l'API. Les données ayant été mises dans la variable **\$scope.meteo**, le template HTML peut les afficher en écrivant **{{meteo.name}}** pour afficher le nom de la ville par exemple.

```

<div ng-if="meteo">
    {{meteo.name}}
    
    <h3>{{meteo.main.temp | number:0}}°</h3>
    {{meteo.weather[0].description}}
</div>

<p ng-if="errorMsg"> {{errorMsg}} </p>

```

Nice



15°
light rain

html/home.html | rendu HTML

Code HTML pour afficher les informations disponibles dans la variable {{meteo}}

2.7. Utilisation de filtre

<http://tutorials.jenkov.com/angularjs/views-and-directives.html#filtering>

> Utiliser des filtres, si nécessaire, pour mettre en forme les données disponibles.

Exemple avec la température qui est donnée avec 2 nombre après la virgule

{{meteo.main.temp}} affiche **16,45** par exemple

Pour afficher la température sans chiffres après la virgule, utilisez le filtre **number** :

{{meteo.main.temp | number:0}} affichera **16**

2.8. Bonus : affichage d'autres icônes pour l'icône météo

Icônes d'illustration de ce TP erikflowers.github.io/weather-icons/ :

```
<i class="wi wi-owm-day-{{meteo.weather[0].id}}"></i>
```