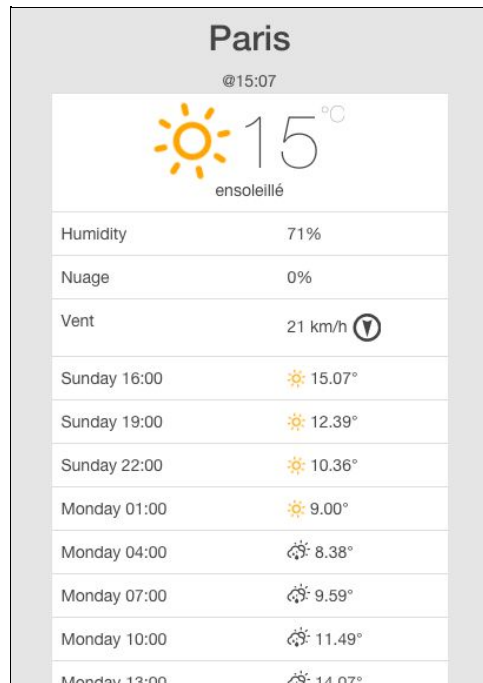




Master 3IR - TP AW #7

Validation de formulaire avec AngularJS et affichage de la météo des 5 prochains jours. Site officiel <https://angularjs.org/>



Affichage des informations météo des 5 prochains jours

1. Installation d'un serveur web

1.1. PC de l'université : dans les salles de TP de l'université, utiliser de préférence Windows et Firefox

1.2. Sur vos PC perso : installer le serveur web "lite-server"

1.2.1. installer NodeJs <https://nodejs.org/en/> version 6.9.4 LTS

1.2.2. dans un terminal de commande, exécuter :

```
npm install -g lite-server
```

1.2.3. pour démarrer le serveur, exécuter la commande "lite-server" dans n'importe quel dossier contenant vos projets web

```
cd "votre_dossier_web"  
lite-server
```

2. Reprendre votre code du TP 6 précédent

Code source : <https://github.com/bilelz/tpaw2017/tree/master/tp6>

Demo : <https://bilelz.github.io/tpaw2017/tp6/>

3. Github TP7

Créer un répertoire intitulé 'TP 7'

Une base pour votre TP est disponible ici :

<https://github.com/bilelz/tpaw2017/tree/master/tp7>

4. Exercice : Validation de formulaire avec AngularJS avec angular-message.js

- 4.1. Dans votre fichier **index.html**, ajouter une référence à la librairie **angular-message.js**, cette librairie gère les messages d'erreur liés aux formulaires

Dans cet exemple nous allons utiliser les librairies depuis les serveurs Google

```
<!DOCTYPE html>
<html ng-app="meteoApp">
  <head>
    <meta charset='utf-8'>
    <title>☀ Weather'13</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <script src="https://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="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular-messages.js"></script>

    <script src="js/app.js"></script>
    <script src="js/route.js"></script>
    <script src="js/controler.js"></script>
  </head>
  <body>
    <div class="container">
      <h1>☀ Weather'13</h1>
      <div ng-view></div>
    </div>
  </body>
</html>
```

index.html

4.2. Ajout de la librairie ngMessages dans js/app.js

Dans le script d'initialisation de votre application Angular, ajouter une référence à ngMessage

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

app.js

4.3. Validation et Messages d'erreur dans le fichier index.html

A screenshot of a web form. At the top left is an orange button labeled 'GPS'. To its right is a text input field. At the top right is a blue button labeled 'Recherche'. Below the input field, a red error message 'Veuillez saisir une ville SVP' is displayed. A red arrow points from the right towards the error message.

Champ de saisie vide

Dans votre page web **home.html** (voir ci-dessous), ajouter un nom “myForm” à votre formulaire.

Ajouter au champ de saisie les attributs **ng-minlength** et **ng-maxlength** pour donner un nombre de caractères min. et max. à saisir pour avoir un formulaire valide.

L'attribut **required** indique que le champ de saisie est obligatoire.

```
<form data-ng-submit="recherche()" name="myForm">
  <div class="form-group">
    <div class="input-group">
      <span class="input-group-btn">
        <button class="btn btn-warning" type="button" data-ng-click="gps()"> GPS </button>
      </span>
      <input type="search" name="city" ng-model="city"
        required ng-minlength="2" ng-maxlength="20" class="form-control"/>
    </div>
    <span class="input-group-btn">
      <!-- l'attribut nb-disabled rendra cliquable le bouton "rechercher" seulement si le
      champ de saisie est correctement rempli -->
      <button class="btn btn-primary" type="submit" ng-disabled="!myForm.$valid">
        Recherche
      </button>
    </span>
  </div>
  <!-- ici, on gère l'affichage des messages d'erreur selon l'état du champ de saisie -->
  <div ng-messages="myForm.city.$error" role="alert">
    <div ng-message="required" class="label label-danger">Veuillez saisir une ville SVP</div>
  </div>
</form>
```

home.html

Exercice : Afficher un message quand le texte saisi est trop petit ou trop grand.

A screenshot of the same web form as before. The text input field now contains the letter 'P'. Below the input field, an orange error message 'Le texte saisi est trop court(3 car. minimum)' is displayed. A red arrow points from the right towards the error message.

Texte trop court

GPS

Parismkjdmsjmdljsqmdsqsd

Recherche

Le texte saisi est trop long (20 car. maximum)

Texte trop long

Lire la documentation pour connaître les différentes options fournies par ng-messages
<https://docs.angularjs.org/api/ngMessages/directive/ngMessages>

5. Exercice : Affichage de la météo pour les 5 prochains jours

5.1.1. Appel AJAX js/controler.js

OpenWeatherMap fourni, via une autre API (forecast), des informations météo pour les 5 prochains jours :

Détail de l'API : <http://openweathermap.org/forecast5>

Dans votre code fichier **js/controler.js**, remplacer l'URL vers l'API par :

[https://demo.bilelz.fr/owmap/forecast/?q=\[.....\]](https://demo.bilelz.fr/owmap/forecast/?q=[.....])

Puis stocker l'information reçu dans la variabel `$scope.forecast`

```
$http.get('https://demo.bilelz.fr/owmap/?q='+$scope.city+'&units=metric
&lang=fr&appid=2de143494c0b295cca9337e1e96b00e0')
    .success(function(data) {
        $scope.meteo = data;
        $scope.loadingMsg = "";
    }).error(function(data) {
        $scope.meteo = {name:"Hum. Error... please retry."};
        $scope.loadingMsg = "";
    });

// appel AJAX pour recevoir les données des 5 prochaines jours
$http.get('https://demo.bilelz.fr/owmap/forecast/?q='+$scope.city+'&units=metric&lang=fr&appid=2de143494c0b295cca93
37e1e96b00e0')
    .success(function(data) {
        $scope.forecast = data;
    }).error(function(data) {
        $scope.loadingMsg = "Erreur pour les données sur 5 jours...";
    });
```

5.1.2. Modifier votre code html pour afficher les informations météo contenues dans la variable `{{forecast}}`

Détail des informations JSON renvoyés par l'API forecast :

<http://openweathermap.org/forecast5#JSON>

La liste des informations à afficher se trouvera dans la variable `{{forecast.list}}`

Ecrire le code HTML qui fera une boucle sur cette liste de données.