

Mise en pratique - Découverte HTML5 - CSS3 - FlexBox - BootStrap - JavaScript -  
JQuery - JQuery Mobile - Cordova

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »

## TABLE DES MATIERES

1. A LA DECOUVERTE DES FRAMEWORK JQUERY MOBILE ET CORDOVA (PLUGIN CORDOVA : LE VIBREUR).....	3
2. APPLI METEO – JQUERY MOBILE, CORDOVA ET API WEB (PLUGIN CORDOVA : GEOLOCALISATION).....	12
3. FORMULAIRE DE CONTACT – JQUERY MOBILE, CORDOVA. (PLUGIN CORDOVA : CONTACTS).....	13

# 1. A LA DECOUVERTE DES FRAMEWORK JQUERY MOBILE ET CORDOVA (PLUGIN CORDOVA : LE VIBREUR).

**Préambule :** Si vous avez réussi à installer un environnement de développement stable pour les applications mobiles Android (NodeJS, Cordova, JDK, SDK Android, Gradle (Ant)) vous pouvez démarrer votre projet de façon classique en utilisant la **CLI** de **Cordova**.

Une page **jQuery Mobile** doit nécessairement déclarer un **doctype HTML5**, ce afin de bénéficier de tous les avantages qu'apporte la nouvelle version du langage à balises, d'ailleurs très bien implémentés dans le **Framework** mobile.

L'entête html (**<head>...</head>**) doit ensuite importer les éléments nécessaires au fonctionnement de **jQM** en suivant cet ordre:

- La feuille de style jQuery Mobile (CSS)
- Le fichier jQuery (JavaScript)
- Le fichier jQuery Mobile (JavaScript)

```
<!DOCTYPE html>
<html>
<head>
  <title>JQuery Mobile</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" href="jQueryMobile/jquery.mobile-1.4.5.css" />
  <script src="JS/jquery.js"></script>
  <script src="jQueryMobile/jquery.mobile-1.4.5.js"></script>

  <link rel="stylesheet" type="text/css" href="css/style.css" />
  <link rel="icon" type="image/png" href="img/jquery-logo.png">
</head>
<body>

</body>
</html>
```

Notons au passage la présence de la métadonnée **viewport** qui spécifie le niveau de zoom de la page au sein du navigateur, ainsi que les dimensions de celle-ci.

Mise en pratique – Découverte HTML5 - CSS3 – FlexBox - BootStrap - JavaScript –  
jQuery - JQuery Mobile - Cordova

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »

Nous précisons ici que la page s'affichera normalement, sans zoom (valeur à 1), et que sa largeur sera égale à celle du **device** (**width=device-width**) depuis lequel la page sera consultée.

Ensuite, vous pouvez commencer à travailler votre page dans les balises **<body></body>** en vous référant au site : <https://demos.jquerymobile.com/1.4.5/pages/>

Avec **jQuery Mobile**, chaque page est associée à une vue de l'application. Définie au sein du tag **<body>**, une page (vue) est identifiée par l'attribut **data-role="page"**

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Ma première vue</h1>
    </div><!-- /header -->
    <div data-role="content">
      <p>Le contenu de la page s'affichera ici.</p>
    </div><!-- /content -->
    <div data-role="footer">
      <h4>&copy; 2019 JQuery Mobile - Un beau pied de page</h4>
    </div><!-- /footer -->
  </div><!-- /page -->
</body>
```

Il est possible de définir **plusieurs pages** au sein d'un même document. Elles sont toutes définies à l'aide de l'attribut **data-role = "page"**. Chaque bloc de page nécessite d'être identifié à l'aide d'un **unique ID**, qui sera utilisé pour lier les pages entre elles (Afin de vous en convaincre, copier ce code dans votre IDE, ce code correspond au document initial de tout projet à page unique **jQuery Mobile**) :

#### index.html

```
<body>
  <div data-role="page" id="page1">
    <div data-role="header">
      <h1>Titre - Page 1</h1>
    </div><!-- /header -->
    <div data-role="content">
      <p>Contenu de la première page.</p>
      <p>Afficher la <a href="#page2">Page n°2</a></p>
    </div><!-- /content -->
    <div data-role="footer">
      <h4>&copy; 2019 JQuery Mobile - Un beau pied de page</h4>
    </div>
  </div>
```

Mise en pratique – Découverte HTML5 - CSS3 – FlexBox - Bootstrap - JavaScript –  
jQuery - JQuery Mobile - Cordova

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »

```

    </div><!-- /footer -->
</div><!-- /page 1 -->
<div data-role="page" id="page2">
  <div data-role="header">
    <h1>Titre - Page 2</h1>
  </div><!-- /header -->
  <div data-role="content">
    <p>Contenu de la seconde page.</p>
    <p><a href="#page1">Retour à la page 1</a></p>
  </div><!-- /content -->
  <div data-role="footer">
    <h4>&copy; 2019 JQuery Mobile - Un beau pied de page</h4>
  </div><!-- /footer -->
</div><!-- /page 2 -->
</body>

```

La transition d'une page à l'autre se fait en **Ajax**, qui se sert des différents **ID** pour trouver la page recherchée (**href="#ID\_de\_la\_page\_recherchée"**).

Mais le passage à une autre page peut se faire différemment si les pages ne sont pas dans le même document.

Si l'on souhaite par contre atteindre un contenu qui n'est pas dans le document courant, **jQuery Mobile** utilise par défaut la technologie **Ajax**, pour passer d'une vue à une autre. Le contenu est importé et rajouté au DOM.

```
<a href="externe.html" rel="external">Une page externe</a>
```

L'attribut **rel="external"** permet de préciser qu'il s'agit bien d'une page externe au document courant. Si l'on ne souhaite pas utiliser **Ajax** pour passer d'un contenu à un autre, il faut mettre l'attribut **data-ajax** à **false**. Soit avec l'exemple précédent :

```
<a href="externe.html" rel="external" data-ajax="false">Une page externe</a>
```

jQuery Mobile propose plusieurs effets de transitions inspirés des transitions disponibles au sein d'applications mobiles natives.

Il existe actuellement 8 transitions : **fade**, **pop**, **flip**, **turn**, **flow**, **slide**, **slideup** et **slidedown**.

L'effet de transition doit être défini dans le lien qui servira au passage de la vue courante vers une autre :

```
<a href="index.html" data-transition="pop">Effet pop</a>
```

On chargera donc le contenu du fichier index.html avec **l'effet pop**.

Afin de vous familiariser avec les différents effets, créer une page externe **page1.html** au document **index.html** initial et ensuite créer un lien externe dans **index.html** qui

pointe vers **page1.html** puis vous pouvez tester les différents **effets de transition** énoncés ci-dessus.

Vous pourrez aussi vous familiariser avec les **boîtes de dialogue** en utilisant dans votre page **index.html** :

```
<a href="page1.html" data-rel="dialog">Boite de Dialogue (Effet Pop)</a>
```

Par défaut c'est l'effet **Pop** qui est utilisé, si vous voulez définir un effet **Flip** vous devrez écrire :

```
<a href="page1.html" data-rel="dialog" data-transition="flip"> Boite de Dialogue  
(Effet Flip)</a>
```

Maintenant que vous avez acquis les bases nécessaires à l'initiation d'un projet **jQuery Mobile** vous allez pouvoir créer une **liste** :

Pour cela, je vous conseille de créer un nouveau document **liste.html** ;-)

Vous allez devoir créer une **liste d'Animaux**, je vous laisse maître des animaux de votre choix.

En revanche, vous devrez générer une liste (**data-role=listview**) avec un thème plutôt sombre (**data-theme= « b »**) et une jolie bordure entre chaque contenu (**data-inset= « true »**) et doté d'un champ de recherche (**data-filter= « true »**)

Concernant les **thèmes** utilisables, ils sont identifiables par des lettres [**a,b,c,d,e**], vous pouvez associer le thème de votre choix sur l'élément de votre choix, ce qui permet de mélanger des thèmes sur une même page. Vous pouvez aussi créer vos propres thèmes, pour cela RDV sur : <http://themeroller.jquerymobile.com/>

Après avoir réalisé cette liste, vous devrez réaliser la page de consultation de l'animal sélectionné dans la liste.

Pour cela, vous devrez implémenter une solution qui présente le **nom** de l'animal, une **description** sommaire de l'animal ainsi qu'une **photo** de l'animal. Pensez à prévoir un **bouton de retour** à la liste dans le **header** de cette page.

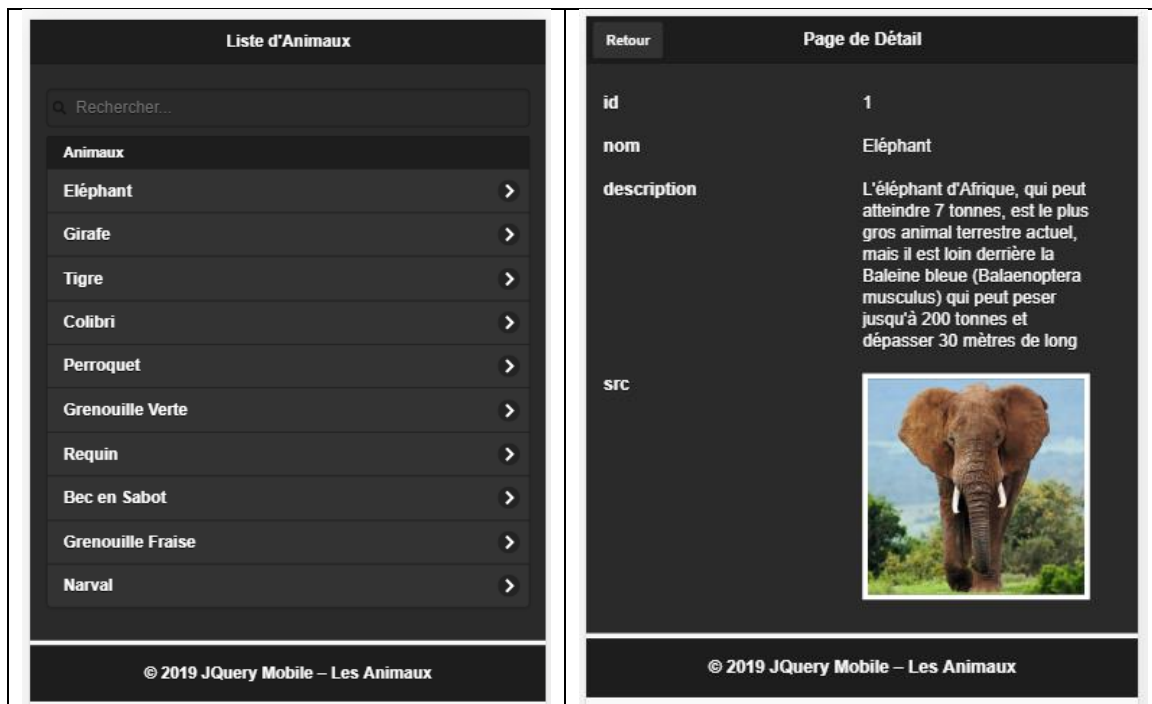
Je vous donne ici les **mots clefs**, mais vous devrez être en autonomie afin d'approfondir vos connaissances sur le sujet (ce qui vous permettra de monter rapidement en compétence sur l'utilisation de ce **Framework jQuery Mobile**).

Vous comprenez vite les limitations de ce système, puisqu'à chaque animal doit correspondre une page de détail spécifique. Et il est bien entendu exclu de développer autant de pages de description qu'il y a d'animaux dans la liste.

Il est donc temps d'implémenter un **système dynamique par programmation**, avec une liste générée **dynamiquement** à partir d'un **tableau d'objets JSON** pour la page d'accueil, puis il faut prévoir une page de détail générique qui exposera les données spécifiques à l'animal sélectionné dans la liste. Vous pourrez utiliser une **grille à 2 colonnes (ui-grid-a)** proposée par **jQM** pour obtenir le rendu final.

Je vous propose ce modèle **JSON** :

```
var animaux = [
  {
    "id": 1,
    "nom": "Eléphant",
    "description": "L'éléphant d'Afrique, qui peut atteindre 7 tonnes, est le plus gros animal terrestre actuel, mais il est loin derrière la Baleine bleue (Balaenoptera musculus) qui peut peser jusqu'à 200 tonnes et dépasser 30 mètres de long ",
    "src": "img/Elephant.jpg"
  },
  {
    "id": 2,
    "nom": "Girafe",
    "description": "La Girafe Bla bla bla ...",
    "src": "img/Girafe.jpg"
  },
  {
    "id": 3,
    "nom": "Tigre",
    "description": "Le Tigre Bla bla bla ...",
    "src": "img/Tigre.jpg"
  }
]
```



Bien que sommaire, nous aimerions maintenant que cette petite application puisse s'exécuter sur un **Smartphone** et nous pourrions tester comme cela certaines fonctionnalités liées au mobile.

Dans le cadre de cette démarche, nous allons utiliser le Framework **Apache Cordova**. Pour vous faciliter la tâche, nous allons utiliser la distribution **Adobe PhoneGap** de ce Framework puisqu'il met à notre disposition des outils très pratiques afin de simplifier le développement d'une application mobile.

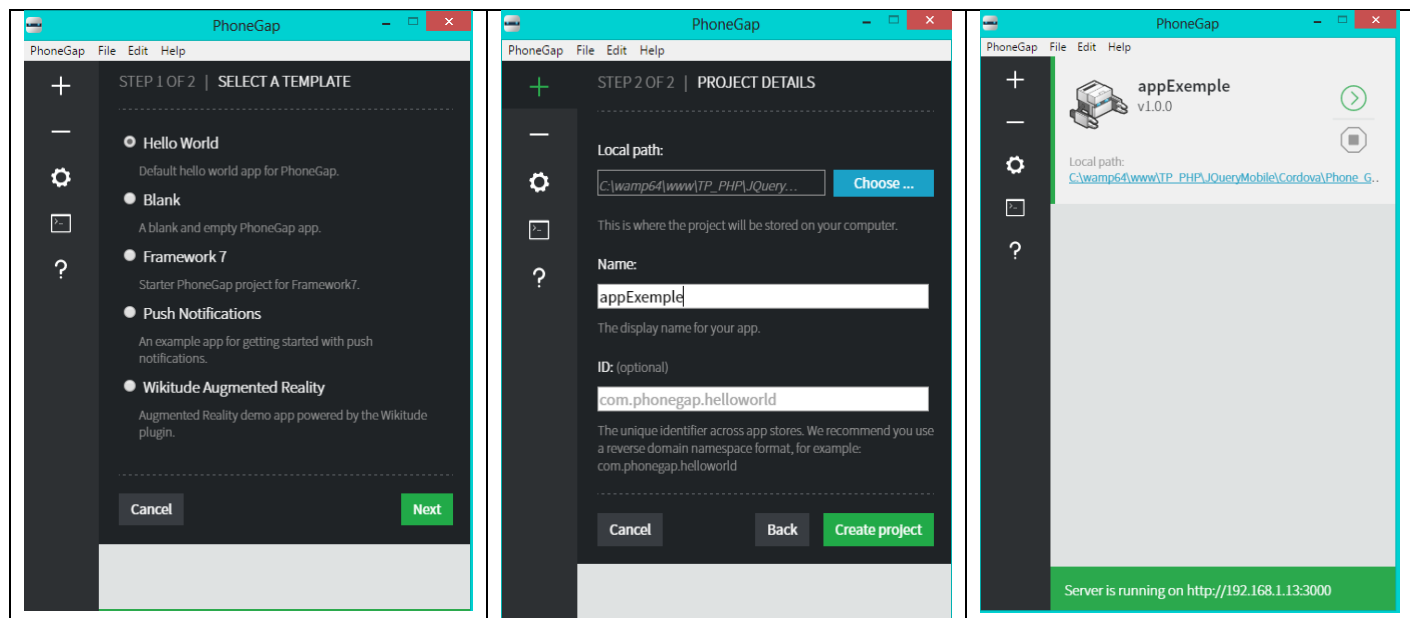
Dans un premier temps vous allez télécharger l'application de bureau **PhoneGap Desktop**, (mais vous pourriez tout aussi bien utiliser le package **npm CLI Cordova** qui vous permettrait de créer et d'enrichir votre application **Cordova** en ligne de commande).

Du côté de votre **Smartphone**, vous allez installer l'application pour mobile : **PhoneGap Developer**. Cette application vous permettra d'exécuter votre application mobile et de vous offrir un rendu sur votre **Smartphone** en vous connectant au serveur que vous proposera l'application de bureau **PhoneGap Desktop** pour peu que votre mobile soit sur le même réseau que votre ordinateur et son serveur local.



Maintenant que ces deux applications sont installées, vous allez utiliser l'application de Bureau (**PhoneGap Desktop** pour démarrer un nouveau projet Cordova, ce sera le classique **HelloWorld**).

Les 3 captures d'écran suivantes vous permettrons de bien choisir les étapes à suivre :



Sur la dernière capture d'écran, vous remarquez que l'application vous fournit en bas de page sur le rectangle à fond vert l'adresse du serveur qui « héberge votre application », cette adresse vous sera utile pour passer à la deuxième étape et pour paramétrer votre application **PhoneGap Developer**.

Cependant, vous pouvez toujours exécuter votre application à partir de votre **Navigateur** en rentrant l'adresse **localhost : 3000**.

Lancer votre application **PhoneGap Developer** sur votre mobile et suivez les étapes suivantes :



A la première étape, rentrez l'adresse du serveur fournit par votre application de bureau : Une adresse du genre : **192.168.1.13 :3000** puis appuyez sur le bouton **Connect**. En cas de succès, vous devriez obtenir l'écran de la dernière capture.

Maintenant, à l'aide de votre IDE préféré ; ouvrez le répertoire de votre projet **PhoneGap Cordova** nouvellement créé et afficher le fichier **index.html** du répertoire **www**.

A ce niveau, vous pouvez essayer de modifier ce fichier (par exemple modifier le titre **<h1>PhoneGap</h1>** par **<h1>Salut Votre prénom</h1>**) et constater le résultat de cette modification dans la fenêtre de votre navigateur (à l'adresse **localhost : 3000**) puis les répercussions sur votre mobile ;-).

Habituez-vous à ce flux de travail, en jouant pour l'instant sur les scripts (**index.html**, **index.css**, **index.js**), vous constatez que vous êtes finalement en train de faire du **développement Web** avec un résultat immédiat sur votre **Smartphone**.

Il est donc temps de passer à l'étape suivante :

Vous allez intégrer dans cette application **Cordova** le travail que vous avez déjà fait pour l'application « **Les Animaux** ». Vous allez devoir intégrer dans le fichier **index.html** de votre application **PhoneGap Cordova**, tous les éléments nécessaires à l'exécution de cette application (bibliothèque JQuery, JQuery Mobile, et feuilles de style). Ensuite, dans une balise **<script></script>** le code **JQuery** déjà implémenté pour l'application « **Les Animaux** » puis dans la balise **<body></body>** le code relatif à l'implémentation des deux vues (la **liste des animaux** et la **page de détail**).

Mise en pratique – Découverte HTML5 - CSS3 – FlexBox - BootStrap - JavaScript – JQuery - JQuery Mobile - Cordova

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »

Pour le reste, vous ne devez absolument pas modifier le code ci-dessous fournit par l'application **Cordova** :

```
***
<script type="text/javascript" src="cordova.js"></script>
  <script type="text/javascript" src="js/index.js"></script>
  <script type="text/javascript">
    app.initialize();
  </script>
</body>
</html>
```

A ce stade-là, vous pourrez constater que vous obtenez bien une parfaite exécution de votre programme, à la fois dans votre **Navigateur** à l'adresse **localhost : 3000** et sur votre **Smartphone** ;-) Ce qui est déjà une satisfaction !!!

Tout l'intérêt du **Framework PhoneGap Cordova**, c'est qu'il vous permet d'accéder aux **fonctionnalités matérielles** de votre **Smartphone** (Base de Contacts, Camera, GPS, Statut de la Batterie, Accéléromètre, Vibreur, ...)

Pour vous en rendre compte, vous allez donc implémenter une **vibration** de **0.5 s** à chaque fois que l'utilisateur clique sur **un élément** de la **liste des animaux**.



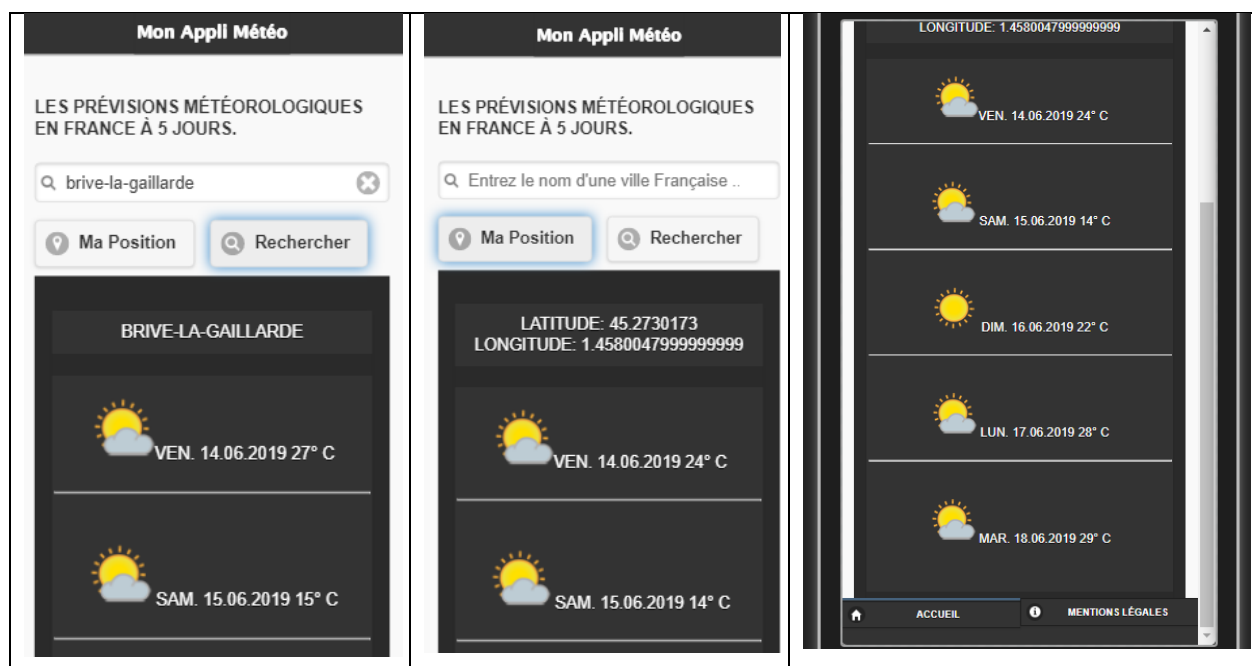
Ehhhh !!! Ça vibrrrrrrrrrrre !!!!! ;-)

## 2. APPLI METEO – JQUERY MOBILE, CORDOVA ET API WEB (PLUGIN CORDOVA : GEOLOCALISATION)

Avec ce TP, vous allez mettre en pratique vos compétences acquises sur les Framework **JQuery Mobile** et **Cordova**, puis sur l'exploitation d'un flux JSON obtenu à partir d'une **API Web** de météo (<https://www.prevision-meteo.ch/>).

Le but sera de développer une petite application mobile qui donnera la possibilité de connaître la **météo et les prévisions à 4 jours** à partir d'une **ville Française** et/ou à partir des données de **géolocalisation** de votre **mobile**.

Les 3 captures d'écran suivantes vous donneront les indications nécessaires pour pouvoir construire votre interface utilisateur :



Comme vous pouvez le constater, cette application disposera de 2 boutons (**Accueil** et **Mentions Légales**), par défaut, au chargement de l'application, l'utilisateur sera sur la page d'accueil qui présentera le champ de recherche et 2 boutons (**Ma Position**, **Rechercher**).

Mise en pratique – Découverte HTML5 - CSS3 – FlexBox - BootStrap - JavaScript –  
jQuery - JQuery Mobile - Cordova

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »

Le bouton **Rechercher** permettra de lancer une recherche à partir du nom d'une **ville Française**. Le bouton **Ma Position** permettra d'obtenir les **coordonnées GPS (Longitude et Latitude)**.

Dans les deux cas, l'utilisateur obtiendra les **prévisions météo à 5 jours** comme le présentent les captures d'écran.

Les informations relatives à l'API Web vous donneront toutes les indications afin que vous puissiez exploiter au mieux les données exposées.

Pour ce TP, les connaissances acquises jusqu'ici vous permettront d'être suffisamment autonome pour pouvoir gérer les **erreurs (erreurs de saisie, problème de chargement asynchrone des données, ...)**.

Vous pourrez tester cette application sur votre mobile en utilisant les technologies rencontrées lors du TP précédent.

Bon courage à vous ...

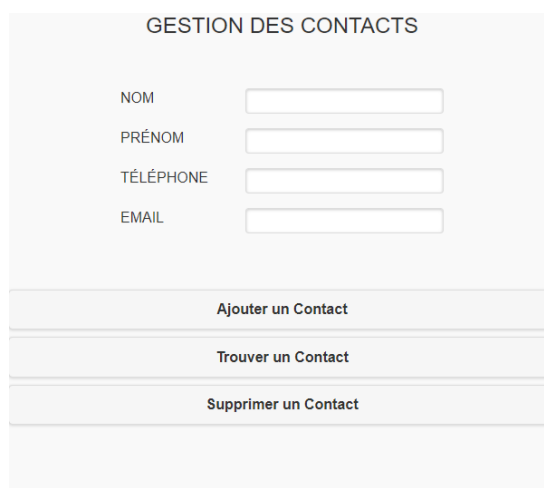
### 3. FORMULAIRE DE CONTACT – JQUERY MOBILE, CORDOVA. (PLUGIN CORDOVA : CONTACTS)

Maintenant que vous êtes familiarisé avec les Framework **JQuery Mobile** et **Cordova**, vous devrez implémenter un **formulaire de contact** très simple (**Nom, Prénom, Téléphone, Adresse Mail**) avec tous les contrôles de validation nécessaires.

Les données une fois validées devront être enregistrées dans la **Base de Contacts** de votre téléphone, l'utilisation du **plugin Contacts de Cordova** sera requise.

Outre la **méthode d'ajout** d'un contact, vous pourrez également implémenter une **méthode de suppression** du Contact et pourquoi pas aussi une **méthode de recherche** d'un Contact dans la base de contacts de votre smartphone.

Vous devrez évidemment tester cette petite application sur votre smartphone ...



GESTION DES CONTACTS

NOM

PRÉNOM

TÉLÉPHONE

EMAIL

Ajouter un Contact

Trouver un Contact

Supprimer un Contact

Mise en pratique – Découverte HTML5 - CSS3 – FlexBox - BootStrap - JavaScript – JQuery - JQuery Mobile - Cordova

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »

## **Reproduction interdite**

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

### **CREDITS**

#### **ŒUVRE COLLECTIVE DE L'AFPA**

**Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services**

#### **Equipe de conception (IF, formateur, mediatiseur)**

Formateur : Alexandre RESTOUEIX

**Date de mise à jour : 29/05/2020**

Mise en pratique – Découverte HTML5 - CSS3 – FlexBox - BootStrap - JavaScript –  
JQuery - JQuery Mobile - Cordova

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »