

## Projet Covoiturage TP 5

### Description du thème

Ce TP est le cinquième d'une série mettant en œuvre le développement mobile en utilisant la bibliothèque jQuery Mobile.

Propriétés	Description
<b>Intitulé long</b>	Des TP permettant la découverte d'un mini-framework mobile jQuery Mobile (JQM)
<b>Formation concernée</b>	BTS Services informatiques aux organisations
<b>Matière</b>	SLAM 2, PPE, SLAM 4
<b>Présentation</b>	Les TP proposent de développer une application mobile cross-plateform à différentes itérations du cycle de développement
<b>Notions</b>	Savoirs • D4.1 - Conception et réalisation d'une solution applicative • D4.2 - Maintenance d'une solution applicative Savoir-faire • Programmer un composant logiciel • Exploiter une bibliothèque de composants • Adapter un composant logiciel • Valider et documenter un composant logiciel • Programmer au sein d'un framework
<b>Pré-requis</b>	Développement web, PHP, JavaScript
<b>Outils</b>	Un environnement de développement pour le web, Firebug pour suivre les appels jQuery et Ajax
<b>Mots-clés</b>	Application mobile, jQuery, jQuery Mobile, Ajax
<b>Durée</b>	4h
<b>Auteur(es)</b>	Patrice Grand
<b>Version</b>	v 1.0
<b>Date de publication</b>	Mars 2014

### Énoncé

Cette seconde itération (full-Ajax, document multi-pages) se poursuit par la gestion des offres d'un abonné au service, c'est la partie 3 du découpage fonctionnel présenté dans le TP 4.

Votre chef de projet vous propose le plan suivant :

- La page HTML de gestion des offres
- L'ajout d'une nouvelle offre.

## Partie 3

### La page de gestion des offres

Cette page ne doit présenter qu'une coquille vide pour laquelle le code jQuery va permettre de valoriser ou construire des éléments HTML à partir d'un appel Ajax.

Vous finalisez son code en vous inspirant largement de la page existante.

```
<div data-role="page" id="pagegerermesoffres">
  <div data-role="content" id="divmesoffres">
<?php
include "vues/entetepageavecbuttonretour.html";
?>
  <legend >Mes offres au départ de l'entreprise</legend>
  <div data-role="controlgroup" id="listdepart" data-theme="b">

  </div>
<legend>Mes offres à l'arrivée à l'entreprise</legend>
  <div data-role="controlgroup" id="listarrivee">

  </div>
  <p>
    <a data-role="button" id="btnSupprimer" data-icon="delete" data-inline="true"
      data-mini="true">Supprimer</a>
  </p>
  <p>
    <a href="#pageajouteroffre" data-role="button" id="btnAjouter" data-icon="plus"
      data-inline="true" data-mini="true">Ajouter une offre</a>
  </p>
</div><!-- /fin content -->
<?php
include "vues/pied.html";
?>
</div><!-- /fin page -->
```

Le code jQuery valorise les éléments HTML, vous avez commencé à l'écrire :

```
$("#body").on("click", "#gerermesoffres", function(e) {
  e.preventDefault();
  e.stopPropagation();
  $.post("ajax/traitergerermesoffres.php",
    foncRetourMesOffres, "json");
});
function foncRetourMesOffres(data) {
  $.mobile.changePage("#pagegerermesoffres");
  var lesOffresDepartEntreprise = data['departentreprise'];
  var lesOffresArriveeEntreprise = data['arriveeentreprise'];
  $("#listdepart").empty();
  for(var i = 0; i < lesOffresDepartEntreprise.length; i++){
    var uneOffre = lesOffresDepartEntreprise[i];
    var legende = uneOffre['jour'] + " " + uneOffre['date'] + " " + uneOffre['heure'];
    var html = "<input id=" + uneOffre['id'] + " type=checkbox data-theme=b>";
    html+="<label for=" + uneOffre['id'] + " data-theme=b> "+ legende + "</label>";
    $("#listdepart").append(html);
  }
}
```

Il ne vous reste plus qu'à terminer le code de la fonction *foncRetourMesOffres* afin de gérer les offres *arriveeentreprise*.

Vous avez poursuivi par l'écriture du code de la fonction Ajax, *traitergerermesoffres.php*.

```

<?php
require_once '../util/fonctions.php';
session_start();
$id = $_SESSION['id'];
$toutesMesOffres = array();
$toutesMesOffres['departentreprise'] = array();
$toutesMesOffres['departentreprise'] = getMesOffresDepartEntreprise($id);
$toutesMesOffres['arriveeentreprise'] = array();
$toutesMesOffres['arriveeentreprise'] = getMesOffresArriveesEntreprise($id);
echo json_encode($toutesMesOffres);
?>

```

La suppression des offres est la même que dans l'application existante.

## Question 1

Terminer l'écriture du code de la fonction `foncRetourMesOffres`. Votre chef de projet vous rappelle ce qu'il vous avait signalé concernant la construction des objets jQuery (voir Remarque TP4). Intégrer le code de suppression des offres. Cette tâche est estimée à 1h.

*La page d'ajout d'une offre*

La page HTML `pageajouteroffre.php` reste inchangée, ainsi que le code jQuery.

## Question 2

Écrire le code qui permet d'ajouter une offre ; vous utiliserez largement le code existant de la première itération. Ce travail est estimé à 30 minutes.

*Les accès non souhaités aux pages*

Un petit souci demeure : lorsqu'un internaute, sans s'être connecté au préalable, demande directement une page de l'application, le serveur lui sert cette ressource sans demande d'authentification. Ainsi, l'URL `monDomaine/covoiturage/#pageaccueil` retourne la page demandée !!

Votre chef de projet vous demande de régler ce problème en vous résumant la situation :

- Il y a deux URL qui peuvent être accédées sans authentification : `pageinscription` et la première page servie `monDomaine/covoiturage/`
- La partie derrière le hash (#) est *client-side*, c'est-à-dire que le serveur ne la reçoit pas. Un traitement côté serveur est inapplicable.
- Ainsi seul un traitement jQuery peut tester le hash
- La méthode est de vérifier au chargement d'une page la partie hash et d'orienter éventuellement l'internaute vers la page index (donc de login) s'il n'est pas connecté, dans le cas où cette partie est différente de « `pageinscription` » ou « ».
- Une requête Ajax doit être utilisée pour tester si l'internaute est connecté ; une fonction existe déjà : `estConnecte()`.

Votre chef de projet vous donne le début de code qui s'exécutera à tout nouveau chargement de page :

```

$(document).on( "pageinit", function(e) {
    var page = window.location.hash.substr(1); // récupère la partie après #

```

## Question 3

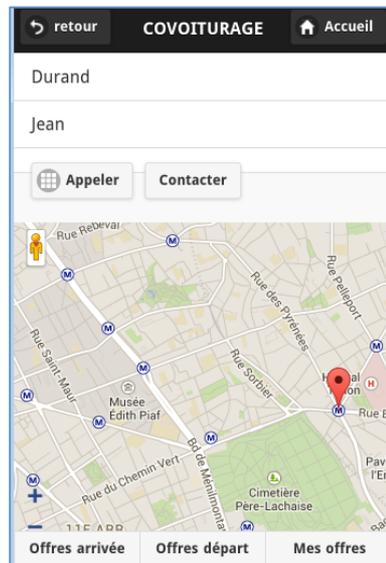
Compléter le code fourni en utilisant un appel Ajax comme proposé. Cette tâche est évaluée à une heure.

## Ajout d'un module de cartographie

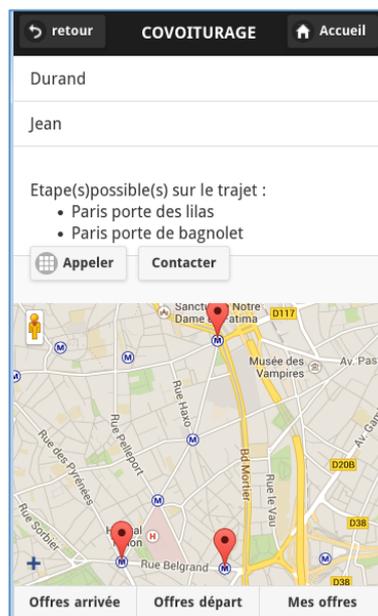
Votre chef de projet envisage une nouvelle fonctionnalité : la possibilité qu'apparaisse, pour chaque offre, une carte qui indique, grâce à un repère, le lieu de dépôt ou le lieu de prise en charge ainsi que les lieux de ramassage sur le parcours.

Cette carte apparaîtrait à la suite des informations sur l'offre (*pageoffre.php*) :

- Cas d'une offre partant de l'entreprise, seul le lieu de destination apparaît :



- Cas d'une offre partant du domicile, le lieu de départ et les points de ramassage apparaissent :



Votre chef de projet vous conseille d'utiliser un plugin jQuery ; plusieurs sont disponibles mais il vous encourage à utiliser gmp3 dont le site en français est <http://gmap3.net/fr/>

Il ajoute quelques précisions :

- Il n'est pas nécessaire de faire un appel Ajax.
- La carte doit s'afficher dans une simple balise *div*.
- Il faudra bien penser à utiliser la méthode *destroy* pour supprimer la carte éventuellement présente dans la *div*.

- Les adresses des lieux doivent respecter une certaine cohérence « *Ville lieu, France* ». Il vous laisse tout loisir de modifier si besoin les données construites en dur dans les tableaux de données.

## **Question 4**

*Réaliser cette tâche évaluée à 1h30.*