

Projet Covoiturage TP 1

Description du thème

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

Propriétés	Description
Intitulé long	Des TP permettant la découverte d'un mini-framework mobile jQuery Mobile (JQM)
Formation concernée	BTS Services informatiques aux organisations
Matière	SLAM 2, PPE, SLAM 4
Présentation	Les TP proposent de développer une application mobile cross-plateform à différentes itérations du cycle de développement
Notions	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
Pré-requis	Développement web, PHP, JavaScript
Outils	Un environnement de développement pour le web, Firebug pour suivre les appels jQuery et Ajax
Mots-clés	Application mobile, jQuery, jQuery Mobile, Ajax
Durée	7h
Auteur(es)	Patrice Grand
Version	v 1.0
Date de publication	Mars 2014

Énoncé

L'entreprise XXX possède son siège dans la proche région parisienne ; sa localisation offre des services de transport en commun très inégaux selon les lieux d'accès ; de nombreux employés parmi les 1200 viennent au travail en voiture. L'entreprise pratique des horaires flexibles qui permettent aux employés de gérer –en partie- leurs horaires de départ et d'arrivée. L'entreprise souhaite mettre en place un service de co-voiturage ouvert à ses seuls employés. Un extrait du cahier des charges est présenté en annexe 1.

Une société de service est chargée de développer une application conforme au cahier des charges. Votre mission, en tant que stagiaire, consiste à participer au module « développement mobile ».

L'application mobile

Pour répondre au cahier des charges, votre société a décidé d'utiliser jQueryMobile –JQM-, jeune Framework dont la première version d'exploitation stabilisée date d'octobre 2011 (RC1).

Question 1

On vous demande de rédiger une courte note -10 lignes- en direction du maître d'œuvre justifiant du choix de jQuery Mobile en regard du cahier des charges. Vous ferez des recherches sur internet. Cette tâche a été évaluée à 1h30.

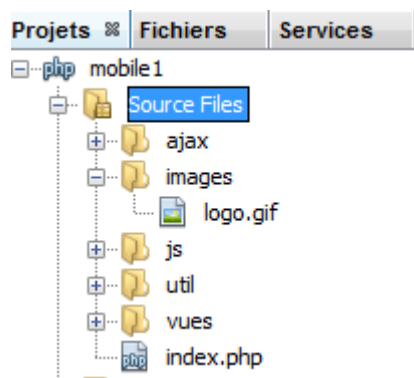
Votre entreprise pratique des méthodes agiles, basées sur de courtes itérations. Une première itération a comme objectif de produire une version bêta B1. Vous allez travailler sur la version B.0.1 déjà entamée.

Version B.0.1

Cette version utilise une approche MVC (côté serveur) sans framework dédié avec un contrôleur unique ; elle ne fait pas appel à une base de données mais utilise des fonctions qui retournent les résultats attendus à partir de jeux de tests, la base de données est envisagée dans une future itération. La technologie Ajax est encore peu mise en œuvre. L'un des objectifs de cette version est de produire une première application testant des composants jQuery mobile.

Votre chef de projet a demandé à ses développeurs de bien commenter le code, ceci parce que la technologie jQuery Mobile n'a pas encore été utilisée dans l'entreprise.

Vous commencez par prendre connaissance du code existant. Le code a été développé en utilisant l'environnement NetBean.



Question 2

En ouvrant les différents répertoires, indiquer les fonctionnalités de chacun d'entre eux.

Vous commencez par lancer le formulaire de connexion, en remplissant les deux champs au hasard. En comparant le code du contrôleur :

```
switch($action)
] {
]   case 'accueil':
      include "vues/entete.html";
      include "vues/pageconnexion.php";
      include "vues/pagemenuaccueil.php";
-     break;
```

et les deux pages annoncées, vous constatez un comportement particulier de JQM avec ce que vous retourne l'application au démarrage. Vous affichez la source du code pour confirmer vos impressions.

Question 3

Indiquer en quoi ce comportement est particulier. Dire ce que JQM fait. Préciser quelle balise et quel attribut jouent un rôle très important dans ce fonctionnement.

Un attribut *data-theme* est présent dans l'entête et le pied de page.

Question 4

Quelle est la valeur de *data-theme* dans l'application ? Modifier cette valeur en choisissant différentes lettres entre « a » et « e ». Que constatez-vous ? Quelle explication peut-on envisager ?

Les premières tâches

Après avoir pris connaissance de cette version 0.1 aux fonctionnalités limitées, votre chef de projet vous assigne un certain nombre de tâches.

Votre chef de projet vous demande de corriger une erreur sur le formulaire affichant la liste des offres ; le lieu d'arrivée n'est pas mentionné à ce moment. Il souhaiterait que vous fassiez évoluer cette vue afin d'obtenir :



Question 5

Réaliser cette modification. Votre chef de projet a évalué cette tâche à 30 minutes.

Certains mobiles ne disposent pas d'une touche « retour ». On vous demande de régler ce problème. Le chef de projet voudrait obtenir (pour les seules pages concernées par un retour) l'écran suivant :



Question 6

Créer un nouveau fichier d'en-tête en ajoutant le bouton demandé. Modifier ses appels dans les fichiers concernés. Votre chef de projet vous indique qu'il suffit d'ajouter une balise *<a* en précisant son attribut *data-rel* = « back » et d'ajouter une icône dédiée. Cette tâche est évaluée à 30 minutes.

Lorsque l'utilisateur clique sur le bouton « connexion », un appel Ajax est fait sur l'événement *click*.

Question 7

- 7.a Dans quel fichier est traité cet appel ?
- 7.b Combien la fonction Ajax fournit-elle d'arguments ?
- 7.c A qui les fournit-elle ?
- 7.d Que contient le paramètre *data* et d'où provient sa valeur ?

Lorsque l'utilisateur demande la liste des offres, l'application retourne cette liste –triée par jour de la semaine- :

mardi
permanent à 15h
15/02/2014 à 16h
mercredi
26/02/2014 à 17h30
permanent à 12h
vendredi
28/02/2014 à 17h30
permanent à 15h30

Votre chef de projet pense que cette vue risque de poser des problèmes ergonomiques si un grand nombre d'offres sont présentées. C'est pourquoi, vous êtes chargé de modifier cette vue afin de présenter un menu de type « accordéon » qui se déroulerait seulement pour le jour sélectionné. Par ailleurs, la destination n'apparaît pas, ce qui n'est pas conforme au cas d'utilisation. Votre chef de projet vous fournit une maquette de la vue attendue :



Il vous précise également qu'en faisant des recherches rapides, il a noté qu'il fallait intervenir sur l'attribut *data-role* et utiliser le rôle « *collapsible* » pour des balises *div* et utiliser un conteneur de *data-role = set-collapsible*.

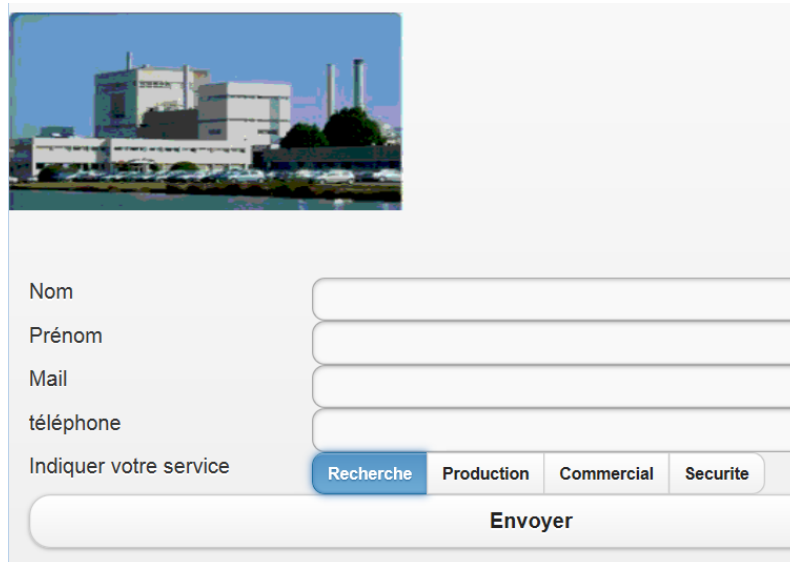
Question 8

Réalisez la modification demandée. Cette tâche est évaluée à 1h30.

Actuellement, l'application ne traite pas l'inscription d'un nouvel utilisateur. Une partie du code est ébauchée : l'appel Ajax sur l'événement click du bouton d'id *btninscription*, le *case* dans le contrôleur et une partie de la page PHP *enregistreruser.php* qui s'exécute avec Ajax.

Il reste à compléter la page de saisie des informations *inscription.php* ainsi qu'à compléter le fichier *enregistreruser.php* qui doit aussi générer le nom de *user* (la première lettre du prénom suivi du nom complet) ainsi que le mot de passe, aléatoire sur 4 caractères (lettres ou chiffres).

Votre chef de projet vous fournit une maquette :



Nom

Prénom

Mail

téléphone

Indiquer votre service

Recherche Production Commercial Securite

Envoyer

Il vous demande de gérer le service avec un élément *fieldset* de *data-role controlgroup*.

Question 9

Réalisez cette nouvelle tâche évaluée à deux heures. Votre chef de projet vous demande ne pas procéder aux contrôles des données ; ce travail sera réalisé un peu plus tard.

Annexe 1 : extrait du cahier des charges

Les objectifs

L'application devra fournir au personnel de l'entreprise la possibilité de consulter des offres de trajet entre l'entreprise et leur domicile. L'application ne sera accessible que par les salariés de l'entreprise. Quiconque pourra déposer une offre de transport (gratuit) entre deux lieux, dont l'un d'entre eux sera obligatoirement l'entreprise. Les offres pourront être consultées sur un ordinateur de bureau, sur un téléphone portable ou sur une tablette.

L'application de type « bureau » contiendra toutes les fonctionnalités attendues :

- La connexion ;
- La création d'une offre ; cette offre pourra être régulière (le mardi à 15h par exemple) ou ponctuelle (le mardi 28 janvier 2014 à 18h par exemple) ;
- La suppression d'une offre par son auteur et par l'administration –date dépassée- ;
- La consultation des offres et du moyen de contacter son auteur (mail et téléphone) ;
- L'administration de l'application.

L'application mobile aura des fonctionnalités réduites :

- La connexion ;
- La création d'une offre ; cette offre pourra être régulière (le mardi à 15h par exemple) ou ponctuelle (le mardi 28 janvier 2014 à 18h par exemple) ;
- La suppression d'une offre par son auteur ;
- La consultation des offres et du moyen de contacter son auteur (mail et téléphone).

L'application mobile pourra être accessible sur toutes les plateformes actuelles pour un minimum de coût ; par ailleurs la maîtrise d'ouvrage souhaite présenter assez rapidement l'application à ses employés.

Annexe 2 : présentation rapide de jQuery

Ce mini-framework se propose de simplifier l'utilisation de Javascript et présente une organisation du code qui rompt avec celle de Javascript.

Tous les appels jQuery sont faits dans une seule fonction et non plus éparpillés au milieu de code HTML. Tout le code est à l'intérieur de cette fonction qui débute par :

```
$(function(){ et se termine par });
```

Cette fonction est dite anonyme –sans nom- ; nous avons la garantie que le code qu'elle contient est exécuté après la construction du DOM.

Il peut y avoir des instructions, des déclarations de variables, mais le plus souvent il s'agit de fonctions exécutées sur des événements.

Comme jQuery centralise toutes ses instructions, il faut un mécanisme pour indiquer que nous désirons faire exécuter du code sur un événement associé à un ou des éléments particuliers du DOM. C'est le rôle du sélecteur jQuery qui va désigner un ou des éléments du DOM.

3 types de sélecteurs sont prévus :

- **A partir de l'attribut id** de l'élément HTML :
\$(«#idDiv1 ») sélectionne la div d'id *idDiv1*
- **A partir du type d'élément** :
\$(«li ») sélectionne tous les éléments de type li ; ici, il est probable que plusieurs éléments seront sélectionnés
- **A partir de la classe** :
\$(« :cla1 ») sélectionne tous les éléments de classe cla1

Il est possible d'utiliser des opérateurs pour préciser les objets atteints par le sélecteur :

- \$(« #idDiv1 >li ») sélectionne tous les éléments li, enfants de la div d'id idDiv1
- \$(« input,label ») sélectionne les input et les labels

Nous ne fournirons pas ici la liste de tous les sélecteurs possibles avec leurs syntaxes, de nombreux sites le font de manière bien plus systématique. Ce qu'il faut retenir c'est l'importance de bien définir le sélecteur avant de lancer des méthodes ou ajouter un gestionnaire d'événement. De nombreuses méthodes appliquées à ces objets jQuery définis par leur sélecteur permettent de rendre de nombreux services, modifier le style (<mon sélecteur>.css(« description »), mettre un flux html (<mon sélecteur>.html(« code html »), retourner des valeurs (<mon sélecteur>.html()), etc.

On peut ajouter un gestionnaire d'événement en faisant :

```
<mon sélecteur>.<nom événement>( fonction(){ code à exécuter } ) ; // fonction anonyme
```

Tous les objets correspondant au sélecteur s'abonneront à la fonction anonyme.

Enfin, jQuery propose une mise en œuvre d'Ajax relativement simple.

Quelques liens utiles :

<http://jquerymobile.com/>

<http://mobile.jquery-fr.com/demos/>

<http://www.w3schools.com/jquerymobile/default.asp>

<http://www.mobile-tuts.com/ebooks/126-développer-avec-jquery-mobile.html>

Par ailleurs, *Opera* fournit un émulateur :

<http://www.opera.com/fr/developper/mobile-emulator>