jQuery Mobile

1. Introduction

jQuery Mobile est une bibliothèque JavaScript permettant de créer des pages web avec un design et une apparence qui s'adaptent parfaitement à toutes les plateformes mobiles. Elle permet ainsi de simplifier le développement des applications web mobiles.

jQuery Mobile a été annoncé en 2010, la version 1.0 est sortie en Novembre 2011. La version actuelle est la 1.4. C'est une bibliothèque stable et performante et utilisée par des milliers de développeurs à travers le monde, ce qui prouve la simplicité de sa prise en main et de sa mise en place. On trouve ainsi une documentation riche et mise à jour sur le site officiel jquerymobile.com et aussi sur plusieurs autres sites (en particulier stackoverflow.com)

jQuery Mobile est un ensemble de plug-ins et de widgets de jQuery visant à déployer une API compatible à toutes les plateformes mobiles permettant de simplifier ainsi le développement des applications Web mobiles. Cette API s'appuie sur HTML5, CSS3 et JavaScript en offrant un niveau d'abstraction plus élevé.

Au niveau du code de programmation, jQuery Mobile est très similaire à jQuery UI, alors que ce dernier se concentre sur le développement d'applications bureautiques, jQuery Mobile se cantonne exclusivement au développement pour les appareils mobiles.



2. jQueryMobile et DOM

jQuery Mobile s'appuie sur les fonctionnalités de HTML 5 et de CSS 3 afin de mettre en place des balises HTML visant à déployer une interface mobile optimisée aux appareils mobiles. Il permet de créer des documents HTML conformes au doctype HTML5.

jQuery Mobile, est une bibliothèque JavaScript et donc adopte naturellement le principe de programmation évènementielle. En effet, en cliquant sur un bouton ou en sélectionnant un choix à partir d'une liste déroulante par exemple, on déclenche un événement. jQuery Mobile peut modifier les éléments DOM en réaction aux évènements déclenchés par l'utilisateur.



3. Comment utiliser la bibliothèque jQuery Mobile ?

La version complète de jQuery Mobile comprend une douzaine de fichiers : des fichiers JavaScript, des feuilles de styles CSS, et des images au format PNG correspondant à des icônes pratiques.

Dans le document HTML du projet, il faut inclure le fichier JavaScript de référence, ou sa version réduite. Étant donné que jQuery Mobile dépend de la bibiothèque jQuery, celle-ci doit être invoquée avant l'importation de jQM. jQuery Mobile comprend aussi une feuille de styles qui définit la mise en forme et l'apparence des éléments d'une page. Elle prépare également

les transitions et les animations à l'aide de transformations en CSS 3. jQuery Mobile inclut aussi une série de petits éléments graphiques que l'on utilise en tant qu'éléments d'interface utilisateur. Il s'agit d'icônes standard employées essentiellement pour soutenir la navigation.

On distingue deux solutions pour charger les bibliothèques jQuery Mobile, jQuery et le fichier de style :

- Télécharger (les fichiers JavaScript, CSS et les images) à partir du site officiel (http://jquerymobile.com/download) et les utiliser localement,
- Faire un lien vers les fichiers JavaScript et CSS directement hébergés par les CDN (Content Delivery Network). Cette méthode a l'avantage de nous éviter de télécharger des documents (JavaScript, images).

Remarque : Pour le mode offline (hors connexion) des applications mobiles, il faut télécharger la bibliothèque jQuery Mobile et l'associer au projet pour que l'application continue à fonctionner correctement en l'absence d'une connexion internet.

Voici un exemple de chargement de l'environnement nécessaire pour une application mobile JQM. Dans cet exemple, on inclut les fichiers JavaScript et CSS en utilisant le CDN.

```
<head>
<!- Inclure la bibliothèque jQuery -->
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<!- Inclure la bibliothèque jQuery Mobile -->
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
<!- Inclure le style jQuery Mobile -->
</script rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"></script>
</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"></script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"</script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"></script src="http://
```

4. Structure d'une application mobile

Pages

La structure d'une application jQueryMobile est basée sur le concept de page. Une application est composée de plusieurs pages, chacune propose un ensemble de fonctionnalités



Comme nous avons déjà mentionné, jQuery Mobile respecte les spécifications HTML5, tout particulièrement les spécifications relatives aux attributs **data-*** (* sera remplacée par différentes valeurs dans la suite). Grâce à cette méthode, il est possible d'intégrer des données dans des balises conformes au HTML 5. En effet, lors de son initialisation, jQuery Mobile sélectionne les éléments en fonction de leur attribut **data-*** et les met à jour en insérant des balises complémentaires, de nouvelles classes CSS, et en leur appliquant des gestionnaires d'événement.

Nous focalisons dans la suite sur l'attribut: **data-role**. Cet attribut peut prendre les valeurs suivantes : **page**, **header**, **content** et **footer**. En effet, une page jQM est généralement composée de 3 sections comme montre la figure suivante :



Voici un extrait d'un document HTML d'une application jQuery Mobile :

```
<body>
<div data-role="page">
<div data-role="header">
<h1> Page d'acceuil</h1>
</div>
<div data-role="main" class="ui-content">
Contenu de l'application
</div>
<div data-role="footer">
<h1>Footer </h1>
</div>
</div>
</div>
</div>
```

Dans l'exemple,

- data-role="page" désigne une page dans une application jQuery mobile.
- **data-role="header"** crée une barre d'outils (toolbar) en haut de la page (souvent utilisé pour le titre ou la recherche des boutons)
- data-role= "main" définit le contenu de la page, comme du texte, des images, des boutons, des formes, etc. La classe "ui-content" ajoute un remplissage et une marge supplémentaire pour le contenu
- **data-role= "footer"** crée une barre d'outils en bas de la page

Application multi-pages

Il faut distinguer, dans une application jQueryMobile, la différence entre un document HTML et une page jQuery Mobile. En effet, un document HTML peut comporter plusieurs pages. En donnant le rôle de page à un élément avec à l'attribut data-role, vous l'isolez et lui donnez une importance qui permettra au navigateur de le considérer comme un fichier à part entière. Chaque page doit avoir identifiant unique, par la suite on utilise **href** pour créer des liens entre ces pages

L'idée de créer plusieurs pages dans un même document HTML est intéressante dans les applications mobiles. En fait, en chargeant du DOM du document HTML, ainsi que les traitements associés, améliore considérablement la rapidité et la fluidité de l'application mobile.

Modèle multipages





Modèle pages séparées



agenda.html	carte.html
id="agenda"	id="carte"
Header	Header
Content	Content

Voici un extrait d'un document HTML qui permet la navigation entre deux pages jQM.

```
<div data-role="page" id="pageone">
<div data-role="header">
  <h1>Welcome To Page 1</h1>
</div>
<div data-role="main" class="ui-content">
       First page
       <a href="#pagetwo">Go to Page Two</a>
</div>
<div data-role="footer">
       <h1>Footer Text</h1>
</div>
</div>
<div data-role="page" id="pagetwo">
<div data-role="header">
  <h1>Welcome To Page 2</h1>
</div>
<div data-role="main" class="ui-content">
       Second page
       <a href="#pageone">Go to Page One</a>
</div>
<div data-role="footer">
       <h1>Footer Text</h1>
</div>
</div>
```

Welcome To Page 1	Welcome To Page 2
First page Go to Page Two	Second Page Go to Page One
Footer Text	Footer Text

Dialogs

Toute page peut être présentée comme une boîte de dialogue modale. Il suffit d'ajouter l'attribut data-dialog="true" pour créer une boite de dialogue.

```
<div data-role="page" id="pageone">
   <div data-role="main" class="ui-content">
        <a href="#pagetwo">Go to Page Two</a>
        </div>
</div>
</div>
</div>
</div data-role="page" data-dialog="true" id="pagetwo">
        <div data-role="main" class="ui-content">
        <a href="#pageone">Go to Page One</a>
        </div>
</div>
</div>
```

Tranisitions

Le framework jQuery Mobile comprend un ensemble d'effets de transition (inclus dans le CSS) qui peut être appliqué à tout lien de page ou formulaire en utilisant l'attribut data-transition.

Slide to Page Two

fade	Default. Fades to the next page	slidefade	Slides from right to left and fades in the next page
flip	Flips to the next page from back to front	slideup	Slides to the next page from bottom to top
flow	Throws the current page away and comes in with the next page	slidedown	Slides to the next page from top to bottom
рор	Goes to the next page like a popup window	turn	Turns to the next page
slide	Slides to the next page from right to left	none	No transition effect

Eléments graphiques

jQuery Mobile est une bibliothèque riche en éléments graphiques permettant de créer des interfaces très adaptés aux plateformes mobiles.

Boutons

jQuery Mobile distingue trois manières pour créer des boutons.

■ Avec l'élément <input>

<input type="button" value="Button">

■ Avec l'élément <button> et l'attribut class="ui-btn"

<button class="ui-btn">Button</button>

■ Avec l'élément <a> et l'attribut class="ui-btn"

Button

Les boutons permettent d'implémenter le modèle de navigation entre les pages.

<a class="ui-btn" href="#pagetw</th><th>wo">Go to Page Two		
	Buttons	
	Welcome!	
	Go to Page Two	
	Footer Text	

Le Bouton "Back", permettant de revisiter la page précédente, est implémenté en utilisant l'attribut **data-rel="back"** (permet d'ignorer la valeur de *href*)



Il est possible de grouper des boutons verticalement ou horizontalement. Ceci est spécifié par l'attribut **data-role="controlgroup"** avec l'attribut **data-type="horizontal|vertical"** pour spécifier un groupement horizontal ou vertical.

```
<div data-role="controlgroup" data-type="horizontal">
  <a href="#" class="ui-btn">Button 1</a>
  <a href="#" class="ui-btn">Button 2</a>
  <a href="#" class="ui-btn">Button 3</a>
  </div>
```

Le Framework jQM permet aussi d'aligner les boutons.

Go to Page Two

Les icônes sont très utiles pour une expérience utilisateur riche. jQM offre plusieurs types d'icônes qu'on peut utiliser pour décorer les boutons.

Icon class	Description	Icon
ui-icon-arrow-l	Left Arrow	O
ui-icon-arrow-r	Right Arrow	0
ui-icon-info	Information	Ó
ui-icon-delete	Delete	\odot
ui-icon-back	Back	Ð
ui-icon-audio	Speakers	\bigcirc
ui-icon-lock	Padlock	•
ui-icon-search	Search	0
ui-icon-alert	Alert	
ui-icon-grid	Grid	
ui-icon-home	Home	G

Le site officiel présente une documentation détaillée des différents boutons qu'on peut intégrer dans les pages de l'application mobiles.

Listes et Filtres

Il est possible d'ajouter des listes en utilisant l'attribut data-role="list-divider".

 data-role="list-divider">Europe 	Europe	
<pre>href="#">Norway</pre>	Norway	۲
Germany	Germany	\odot

On peut ajouter un filtre à une liste en utilisant comme montre l'exemple suivant :



Q Search for names	
Α	
Adele	\mathbf{O}
Agnes	Ø
В	
Billy	\mathbf{O}
Bob	Ø
c	
Calvin	٥

Remarque : Le site officiel présente une documentation détaillée avec des exemples et des illustrations de différents composants graphiques qu'on peut utiliser pour contruire une application mobile.

Formulaire

jQuery Mobile met en forme automatiquement les formulaires HTML pour les rendre attractifs et touch-friendly. On peut intégrer différents composants dans un formulaire : Text Inputs, Search Inputs, Radio Buttons, Checkboxes, Select Menus, Sliders, Flip Toggle Switches

jQuery Mobile Forms		
Full Name:	What's Your Name?	
Looking for anything?	Q Search for content	
Today's date:		
Choose Favorite Color:	Red	
Flip Switch:	Off	
Choose Favo	prite Movies:	
Choose Fave	orite Movies: nawshank Redemption	

Thèmes

Pour personnaliser une application mobile avec un thème donnée, jQueryMobile propose par défaut deux thèmes a et b. Il suffit de préciser le thème souhaité avec l'attribut data-theme et la valeur a ou b.

Page Header		Page Header
Some Text A Standard Text Link		Some Text <u>A Standard Text Link</u>
Link Button		Link Button
A List View:		A List View:
A		А
Adele	Ø	Adele
В		В
Billy	Ø	Billy
Input Field:		Input Field:
Name		Name
Toggle Switch:		Toggle Switch:
Off		Off

Notons qu'il possible d'intégrer d'autres payants ou gratuits. Il est aussi possible d'utiliser exemple l'outil themeroller, pour personnaliser un thème existant ou créer un nouveau thème.

ThemeRoller	Version 1.4.5	p Spector off Spector off Spector off Spector off Spector off Spector	ort Share Help center
Global A B C + Theme Settings Font Family	Drag a color onto an element below or pick from the	Adobe Kuler swatches > Recent Colors	
> Comer Radii	Show alternative icons in preview	See icon demos for usage	
 Icon Day Chadawa 		loi dougo.	
/ box snadow	A	в 🕕 в 💮	• c •
	Body Sample text and links.	Body Sample text and <u>links</u> .	Body Sample text and <u>links</u> .
	List Header	List Header	List Header
	Read-only list item	Read-only list item	Read-only list item
	Linked list item	Linked list item	Linked list item
	O Radio	Radio	O Radio
	Checkbox	Checkbox	Checkbox
	On Off	On Off	On Off
	Option 1	Option 1	Option 1

5. Gestion des évènements

jQM donne aux développeurs une flexibilité pour contrôler le comportement de l'application mobile en fonction des interactions des utilisateurs et ce via les évènements.

Evènements tactiles

jQuery Mobile distingue différentes catégories d'évènements. Le premier type est lié au pointage ou touché de l'écran, il s'agit des évènements tactiles (*Touch events*). Par exemple lorsque l'utilisateur touche l'écran, l'evenement tap se déclenche. On distingue aussi les evenements, taphold, swipe, swipeleft/swiperight)

```
$("p").on("tap",function(){
    $(this).hide();
});

$("p").on("swipe",function(){
    $("span").text("détection de swipe");
});
```

```
$("p").on("taphold",function(){
    $(this).hide();
});
```

Heithein Aubes

Evènements d'orientation

Le deuxième type est lié au changement d'orientation de l'écran (*Orientation events*). Ce type d'évènement est déclenché lorsque le dispositif tourne verticalement ou horizontalement

Evènements de défilement

Un troisième type est lié au défilement de l'écran (*Scroll events*). Lorsque on défile en haut ou en bas les deux évènements (scrollstart, scrollstop se déclenchent pour indiquer le démarrage de scroll ainsi que sa fin.



Evènements de pages

Le dernier type est lié aux pages. Ces événements sont déclenchés au moment de

- l'Initialisation de page
- Ou de Chargement/Déchargement de page
- Ou de la transition d'une page à une autre

• Evènements d'initialisation

L'initialisation d'une page passe par deux étapes; avant la création de la page et une fois la page est créée. Chaque étape est associée à un événement, à savoir **pagebeforecreated** et **pagecreated**

• Evènement de chargement

Le deuxième type d'évènements concerne le chargement des pages externes (page load/unload). Il y 3 évènements associés à ce type :

- 1- l'évènement **pagecontainerbeforeload** : déclenché Avant le chargement de la page externe dans le DOM
- 2- pagecontainerload est déclenché si chargement est réalisé avec succès
- 3- pagecontainerloadfailed est déclenché s'il y a un problème de chargement

• Evènements de transition

Comme 3^{ème} type des évènements de page, on trouve les évènements de transition. La transition implique deux pages : la page source et la page destination. Les transitions animent le passage de la page source à la page destination.

Voici les 4 évènements de transition

 Pagebeforeshow : déclenché sur la page destination avant que l'animation de transition commence

- Pageshow, est aussi déclenché sur la page destination mais après la terminaison de l'animation de transition
- **Pagebeforehide** est déclenché sur la page source avant que la tranisition commence
- **Pagehide** est également déclenché sur la page source après la terminaison de l'animation de transition

Chargement du DOM

Avec jQuery, on attend que l'évènement « **ready** » est déclenché, ç-a-d que le DOM HTML de la page associée est chargé dans le navigateur pour que jQuery commence l'exécution de ses fonctions JS sur les composants du DOM.

On garde le même principe avec jQueryMobile, sauf que ici on attend le déclenchement de l'évènement « **pagecreate** »



<script></th><th></th></tr><tr><td>\$(document).on("pagecreate","#page1",functio</td><td>n(){</td></tr><tr><td>// Appel d'évènements jQueryMobile</td><td></td></tr><tr><td>});</td><td></td></tr><tr><td></script> <td></td>	
---	--