HTML5 for Mobile Device

Yu-Hsiang Huang
HTML5 for Mobile Device

Yu-Hsiang Huang
Outline

- Introduction HTML5
- jQuery Mobile
- Native APP vs Web APP
- Hybrid APP
- Connect to Social Network
- Connect to Flash Server
What is HTML5?

HTML5 will be the new standard for HTML.

The previous version of HTML, HTML 4.01, came in 1999. The web has changed a lot since then.

HTML5 is still a work in progress. However, the major browsers support any of the new HTML5 elements and APIs.
Minimum HTML5 Document

<!DOCTYPE html>
<html>
<head>
    <title>Title of the document</title>
</head>

<body>
    The content of the document......
</body>

</html>
HTML Demos

http://slides.html5rocks.com/#notifications-api
Intro. jQuery Mobile

- A unified, HTML5-based user interface system for all popular mobile device platforms, built on the rock-solid jQuery and jQuery UI foundation. Its lightweight code is built with progressive enhancement, and has a flexible, easily themeable design.
jQuery Mobile pros

- jQuery extension, the lower the cost of learning
- Free, Open-Source, lightweight
- Can be shared with the computer code
- No developer account to test
- Cross-platform

- Cross-Device (cross-browser resolution)
  - Responsive Design
Responsive Design
jQuery Mobile cons

- Efficiency than the Native Code
- Maintenance and efficiency Trade-off
- Can not operate the phone hardware
- Must connect to the Internet
- Difficult to commercialization
- Effect applies to content-based App
jQuery Mobile

• http://www.jqmgallery.com/
jQuery UI Component

http://jquerymobile.com/test/
Native APP vs. Web APP

<table>
<thead>
<tr>
<th>功能</th>
<th>Web App</th>
<th>Native App</th>
</tr>
</thead>
<tbody>
<tr>
<td>靜態圖片、文字展示</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>內容可直接線上更新</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>可上架販售</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>離線閱覽</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>加入主畫面螢幕</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>顯示自定啓動圖片</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
Native APP

Pros:
- Better performance (at least for now), snappier animations, transitions, and faster load times. The performance difference between native and web apps is far more pronounced on slower devices (e.g. iPhone 3G running iOS4)
- Can store more data offline
- Can be featured and searched for in the app store
- Full access to the device’s hardware and OS features
- Implicit installation of an app to the device’s home screen. On iOS devices you can add any web app to your home screen, but it’s a manual process
- The App Store handles purchase transactions on your behalf

Cons:
- Typically more expensive to build, even for a single platform. Build costs increase significantly for each new platform.
- Because the codebase needs to be re-worked for each OS, the time to build an app for multiple devices can also be quite involved.
- Your app must be accessed through the device’s app store, which has two important considerations: your app must go through an approval process, which can be lengthy and arbitrary, and if your app generates revenue you must share a percentage with the store (30% for Apple’s App Store, including in-app purchases). App updates must go through a new approval process each time.
Web APP

Pros:
- A single codebase which can be accessed by any browser-enabled mobile device.
- Uses web technologies (HTML/CSS/Javascript), which are arguably easier to learn than native languages like Objective-C or Java.
- Performance issues are becoming less of an issue as mobile browsers become faster and their Javascript engines keep improving.
- No approval process needed, and updates to the app can happen instantaneously.
- No revenue sharing with an app store.

Cons:
- Using web technologies means interpreted code (as opposed to compiled code for native apps), which some people believe means web apps will always be slower than native apps.
- Don’t have full access to all the methods exposed by the device’s operating system, meaning you are limited to the APIs made available by the browser. As it stands now in Mobile Safari, this means no camera, compass, video capture, microphone, user contacts, file uploading or push/local notifications.
- Can’t be found on the app store. If you’re lucky enough to be a featured app in Apple’s store, for example, it is a huge marketing boost.
- If you are looking to generate revenue, it’s up to you to build a commerce model.
Connect to Facebook

- OAuth Authentication
- Facebook SDK
- Register Facebook APP
- Example
OAuth Authentication

User

ANTrip Webapp

GET ANTrip Login page

User Click FACEBOOK Login button

GET OAuth Dialog

Redirect include Access Token in URL

GET /me?access_token

User information

GET FacebookRegister

return sid

Facebook

PLASH Server
Facebook SDK

https://developers.facebook.com/

Facebook SDK 3.0 for iOS
Native UI views. Better API support. iOS 4+ ready.
Learn More or See What's New

Introducing the App Center
Now there is a new way to grow your app on web or mobile. Learn more

Platform Status: Facebook Platform is Healthy
Game Spotlight: SongPop
Julien Codorniou 於 7月14日發表
Platform Updates: Operation Developer Love
Lei Lei 於 7月12日發表

範例展示
Spotify
Pinterest
ticketmaster
Register Facebook APP
### 應用程式 > ANTrip 雲水途誌

#### 設定

<table>
<thead>
<tr>
<th>欄位</th>
<th>內容</th>
</tr>
</thead>
<tbody>
<tr>
<td>應用程式 ID / API 鑰匙</td>
<td>314048998686760</td>
</tr>
<tr>
<td>App Namespace</td>
<td>plash-antrip</td>
</tr>
<tr>
<td>Site Domain</td>
<td>plash.iis.sinica.edu.tw [and others]</td>
</tr>
<tr>
<td>技術支援信箱</td>
<td><a href="mailto:yhhuang79@hotmail.com">yhhuang79@hotmail.com</a></td>
</tr>
<tr>
<td>應用程式密鑰</td>
<td>4e96541fcb660c814545365f2c97e302</td>
</tr>
<tr>
<td>網站 URL</td>
<td><a href="http://plash.iis.sinica.edu.tw/">http://plash.iis.sinica.edu.tw/</a></td>
</tr>
<tr>
<td>電子郵件</td>
<td><a href="mailto:yhhuang79@hotmail.com">yhhuang79@hotmail.com</a></td>
</tr>
</tbody>
</table>

#### 開放社交關係圖

你尚未新增任何動作、物件或資料。

#### 角色

<table>
<thead>
<tr>
<th>角色</th>
<th>Admins:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="%E5%9C%96%E5%83%8F" alt="圖像" /></td>
</tr>
</tbody>
</table>

#### 洞察報告

<table>
<thead>
<tr>
<th>Facebook 用戶</th>
<th>分享</th>
</tr>
</thead>
<tbody>
<tr>
<td>0每日新用戶</td>
<td>0每日分享的內容</td>
</tr>
</tbody>
</table>
應用程式 > ANTrip 雲水途誌 > 基本

基本資料

Display Name: ANTrip 雲水途誌
Namespace: plash-antrip
聯絡信箱: yhhuang79@hotmail.com
App Domains: plash.is.sinica.edu.tw plash2.is.sinica.edu.tw plash3.is.sinica.edu.tw antrip.plash.tw

類別: 旅遊
Hosting URL: You have not generated a URL through one of our partners (Get one)

請選擇你的應用程式如何跟 Facebook 結合

- Website with Facebook Login
  網站位址 (URL): http://plash.is.sinica.edu.tw

- Mobile Web
  Mobile Web URL: http://antrip.plash.tw

- Facebook 上的 App
  Use my app inside Facebook.com.

- Native iOS App
  Publish from my iOS app to Facebook.

- Native Android App
  Publish from my Android app to Facebook.

- 專頁分頁
  Build a custom tab for Facebook Pages.

儲存
Connect to Plash Server

- JSON and JSONP
- Ajax
- Android http client
JSON and JSONP

- **JSON** - JavaScript Object Notation
  - JSON is syntax for storing and exchanging text information. Much like XML.
  - JSON is smaller than XML, and faster and easier to parse.
- **JSONP** - JSON with padding

**JSON Example**

```json
{
  "employees": [
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
  ]
}
```
Ajax

- AJAX = Asynchronous JavaScript and XML.
- AJAX is a technique for creating fast and dynamic web pages.

## Ajax example

```javascript
var XMLHttpRequest = require('xmlhttprequest').XMLHttpRequest;
var xmlHttp = new XMLHttpRequest();

xmlHttp.onreadystatechange = function() {
  if (xmlHttp.readyState == 4 && xmlHttp.status == 200) {
    // Process the returned data
  }
}

xmlHttp.open('GET', 'http://example.com', true);
xmlHttp.send(null);
```
Ajax example

$.ajax({
data:{username: username, password: password},
type: 'GET',
dataType: 'jsonp',
cache: false,
success:function(result){
  if(result.sid != "0"){
    $.cookie("sid", result.sid);
    $.mobile.changePage("#mainpage");
  } else {
    alert("Login Fail");
  }
}
});
Android HTTP Client

```java
//Danny: Use Yu-Hsiang's HttpClient method
//Date: 2011/06/23
public static HttpClient getHttpClient() {
    try {
        KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
        trustStore.load(null, null);
        SSLSocketFactory sf = new AndroidSSLSocketFactory(trustStore);
        sf.setHostnameVerifier(SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);
        HttpParams params = new BasicHttpParams();

        //set connection timeout
        int connectionTimeout = 3000;
        HttpConnectionParams.setConnectionTimeout(params, timeoutConnection);
        //set socket timeout
        int socketTimeout = 10000;
        HttpConnectionParams.setSoTimeout(params, timeoutSocket);

        HttpProtocolParams.setVersion(params, HttpVersion.HTTP_1_1);
        HttpProtocolParams.setContentCharset(params, HTTP.UTF_8);
        SchemeRegistry registry = new SchemeRegistry();
        registry.register(new Scheme("http", PlainSocketFactory.getSocketFactory(), 80));
        registry.register(new Scheme("https", sf, 443));
        ClientConnectionManager ccm = new ThreadSafeClientConnManager(params, registry);
        return new DefaultHttpClient(ccm, params);
    } catch (Exception e) {
        return new DefaultHttpClient(ccm, params);
    }
}
```
public static HttpClient getHttpClient() {
    try {
        KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
        trustStore.load(null, null);
        SSLSocketFactory sf = new AndroidSSLSocketFactory(trustStore);
        sf.setHostnameVerifier(SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);
        HttpParams params = new BasicHttpParams();

        // set connection timeout
        int timeoutConnection = 3000;
        HttpConnectionParams.setConnectionTimeout(params, timeoutConnection);

        // set socket timeout
        int timeoutSocket = 10000;
        HttpConnectionParams.setSoTimeout(params, timeoutSocket);

        HttpProtocolParams.setVersion(params, HttpVersion.HTTP_1_1);
        HttpProtocolParams.setContentCharset(params, HTTP.UTF_8);
        SchemeRegistry registry = new SchemeRegistry();
        registry.register(new Scheme("http", PlainSocketFactory.getSocketFactory(), 80));
        registry.register(new Scheme("https", sf, 443));
        ClientConnectionManager ccm = new ThreadSafeClientConnManager(params, registry);
        return new DefaultHttpClient(ccm, params);
    } catch (Exception e) {
        return new DefaultHttpClient();
    }
}
synchronized public static JSONObject connectTo(final PLASHGETConnectionSetup setup) throws NullDataException, IllegalArgumentExceptionException
    {  
        if (setup == null){
            throw new IllegalArgumentException("URL string cannot be NULL");
        } else if (!isNetworkAvailable(setup.getContext())){
            throw new NetworkErrorException("no good");
        } else{
            JSONObject j = null;
            for (int retry = 0; retry < 3 && j == null; retry++) {
                try {
                    HttpClient client = getHttpClient();
                    HttpGet request = new HttpGet();
                    request.setURI(new URI(setup.getUrl()));
                    Log.e("url@SetConnection = ", setup.getUrl());
                    HttpResponse response = client.execute(request);
                    BufferedReader in = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));

                    String readInputLine = null;
                    readInputLine = in.readLine();
                    Log.e("in", readInputLine);

                    j = new JSONObject(new JSONTokener(readInputLine));

                    Log.d("end@SetConnection = ", "");
                    in.close();
                } catch (IOException e) {
                    e.printStackTrace();
                } catch (JSONException e) {
                    e.printStackTrace();
                } catch (URISyntaxException e) {
                    e.printStackTrace();
                }
            }
            if (j == null) {
                throw new NullDataException("Null query returned from server");
            } else{
                return j;
            }
        }
    }