Jiří Vávrů

jQuery Mobile

Vytvoření plně funkční mobilní aplikace

Nasazení aplikace do příslušného obchodu

Referenční manuál komponent a API

WWW

Ke stažení zdrojové kódy příkladů z knihy

computer press

Jiří Vávrů

jQuery Mobile

Computer Press
Brno
2013

jQuery Mobile

Jiří Vávrů

Obálka: Martin Sodomka
Odpovědný redaktor: Martin Herodek
Technický redaktor: Jiří Matoušek

Objednávky knih:
http://knihy.cpress.cz
www.albatrosmedia.cz
eshop@albatrosmedia.cz
bezplatná linka 800 555 513
ISBN 978-80-251-3811-3

Vydalo nakladatelství Computer Press v Brně roku 2013 ve společnosti Albatros Media a. s.
se sídlem Na Pankráci 30, Praha 4. Číslo publikace 16 818.
© Albatros Media a. s. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

1. vydání

ALBATROS MEDIA a.s.

Obsah

Úvod **11**

O autorovi **11**
Koncept knihy **11**
Zpětná vazba od čtenářů **12**
Zdrojové kódy ke knize **12**
Errata **12**

ČÁST I
VÝVOJ MOBILNÍ APLIKACE

KAPITOLA 1
Vývojové prostředí a výběr frameworku **15**

PhoneGap **15**
jQuery Mobile **16**
Výběr vývojového prostředí **17**
Emulátory – Android emulátor **17**
Nastavení a konfigurace **19**
Jednoduchá aplikace Hi World **29**
Spuštění v emulátoru **31**
Eclipse 31
Emulátor Ripple 32

KAPITOLA 2
Vývoj reálné aplikace krok za krokem **33**

Vlastnosti a popis aplikace **33**
Use Case diagram aplikace **34**
Use Case přihlášení uživatele 34
Use Case hlavní stránky aplikace 35
Use Case výpisu historie mých příspěvků 36
Use Case nastavení aplikace 37
Založení nového projektu **38**
Import důležitých knihoven pro běh aplikace **40**

Úvodní strana – rozložení ovládacích prvků **42**
Inicializace aplikace a ověření funkčnosti **45**

KAPITOLA 3
Přihlašování uživatele **51**

Založení účtu pro aplikaci na Twitteru **51**
Instalace pluginu ChildBrowser **53**
Instalace pluginu jsOAuth **54**
Přihlášení uživatele pomocí OAuth **55**
Odhlášení uživatele **61**
Doplňující funkce a stránky **62**

KAPITOLA 4
Načítání tweetů: JSON + AJAX a zpracování dat **65**

Datový formát JSON **65**
Zasílání requestu a zpracování odpovědi ze vzdáleného serveru **68**
Výpis načtených dat do ListView **71**

KAPITOLA 5
Odesíláme tweety na server **75**

Přidání nové stránky s formulářem **75**
Získání a ošetření vstupu od uživatele **78**
Odeslání dat a získání odpovědi **80**

KAPITOLA 6
Práce s GPS a geolokací **83**

Práce s GPS, získání GPS koordinátů **83**
Úprava odesílání tweetů, přidání GPS souřadnic **84**

KAPITOLA 7
Načítání vlastních tweetů a práce s Google Maps **89**

Načítání vlastních tweetů **89**
Zobrazování na mapě **93**

KAPITOLA 8
Testujeme aplikaci v emulátoru Android **99**

Využití ladicí konzole v Eclipse **99**
Využití ladících nástrojů v prohlížeči **101**

KAPITOLA 9
Testujeme aplikaci v reálném zařízení **103**

Testování na zařízení **103**

KAPITOLA 10
Export apk souboru **105**

Vytvoření certifikátu pro export digitálně podepsané aplikace **106**

KAPITOLA 11
Příprava aplikace a samotný upload aplikace do Android marketu **109**

Registrace vývojářského účtu **109**
Upload aplikace do Android marketu **110**

KAPITOLA 12
Rady na závěr **111**

Odeberte pluginy PhoneGap, které vaše aplikace nevyužívá **111**
Inicializační funkce umisťujte do onDeviceReady **112**
Zvolte pro svoji aplikaci správný UI framework **112**
Používejte událost tap namísto onClick **112**
Zmenšete zdrojový kód pomocí komprimačních nástrojů **113**

KAPITOLA 13
Tipy a triky **115**

Přesun aplikace na SD kartu **115**
Upravte soubor AndroidManifest.xml 115
Změňte hodnotu Build target 116
Nahrajte do zařízení 116
Linkování aplikací v Android marketu **116**

Vložení reklamního banneru pro zobrazení Google reklamy **117**
Registrace adMob účtu 117
Založení nové kampaně 117
Import AdMob Android SDK 118
Úprava souboru AndroidManifest.xml 118
Úprava souboru MainActivity 119
Vytvoření instalačního souboru bez Eclipse a Android SDK **120**
Registrace 120
Upload zdrojových kódů aplikace 121
Nastavení informací o aplikaci 121
Upload vývojářských certifikátů 122
Omezení služby PhoneGap Build 122

ČÁST II
REFERENČNÍ MANUÁL

KAPITOLA 14
Komponenty **125**

Stránky **125**
Anatomie stránky 125
Titulky stránek 129
Linkování stránek 129
Transformace a přechody mezi stránkami 130
Dialogová okna 132
Cachování obsahu a přednačítání stránek 133
Ajax, hashe & historie 134
Panely nástrojů **135**
Základní informace o nástrojových panelech 136
Header panely 136
Footer panely 139
Navigační lišty 141
Fixované pozicování 144
Tlačítka **147**
Základy 147
Ikony tlačítek 150
Inline tlačítka 154
Sdružená tlačítka 155
Formátování obsahu **156**
Základní formátování obsahu pomocí HTML značek 156

Skládací obsah 160
Rozbalovací sady (accordeon) 162

Formulářové elementy 165
Základy 165
Struktura 165
Značkovací konvence 165
Zmenšené formulářové elementy 165
Skryté labely 166
Deaktivace formulářových elementů 166
Refresh formulářových elementů 167
Zabránění autoinicializaci formulářových elementů 167
Přehled elementů 168
Textové vstupy 169
Vyhledávací vstupní pole 173
Slider 175
Flip toogle switch 178
Přepínače (radio buttons) 181
Zaškrtávací políčka (check boxy) 184
Select menu 186

Listview 192
Základy 192
Vnořené seznamy 192
Číslované seznamy 192
Seznamy pouze pro čtení 193
Seznamy s tlačítky 193
Oddělování položek seznamu 194
Vyhledávací filtr 195
Formátování textu a zobrazení count bubbles 197
Miniatury a ikony v seznamech 198
Vložené seznamy 200
Aktualizace seznamů 200
Konfigurace 201
Metody 204
Události 204

KAPITOLA 15
API jQuery Mobile 205

Výchozí konfigurační nastavení 205
Práce s autoinicializací jQuery Mobile 205

Události **209**
Dotykové události 209
Události virtuální myši 210
Události změny polohy zařízení 211
Události při scrollování 211
Události při načítání stránky 211
Události při změně stránky 214
Události při přechodu stránky 215
Události při inicializaci stránky 217
Události odebrání stránky 217
Události šablon 218
Události animací 218

Metody a utility **218**
$.mobile.changePage (method) 218
$.mobile.loadPage (method) 220
$.mobile.showPageLoadingMsg (method) 221
$.mobile.hidePageLoadingMsg (method) 222
$.mobile.fixedToolbars.show (method) 222
$.mobile.fixedToolbars.hide (method) 222
$.mobile.path.parseUrl (method) 223
$.mobile.path.makePathAbsolute (method) 225
$.mobile.path.makeUrlAbsolute (method) 226
$.mobile.path.isSameDomain (method) 226
$.mobile.path.isRelativeUrl (method) 227
$.mobile.path.isAbsoluteUrl (method) 227
$.mobile.path.get (method) 228
$.mobile.silentScroll (method) 229
$.mobile.activePage (property) 229

KAPITOLA 16
jQuery Mobile Cheat Sheet **231**

Rychlý přehled jQuery Mobile **231**
Instalace 231
Základní struktura stránky 231
Linkování stránek 232
Tlačítka 232
ToolBars 233
List View 233
Themes 234
Special Fields 234

KAPITOLA 17
Důležité pojmy **235**

API **235**
IDE **235**
JavaScript **235**
Framework **236**
JSON **236**
AJAX **236**
USE CASE **237**
DOM **237**

Rejstřík **239**

Úvod

HTML5 se prosazuje velkou měrou na mobilních platformách, jako jsou iPhone, Android, BlackBerry nebo Bada. Díky všeobecně výborné podpoře tohoto formátu je možné vytvářet univerzální aplikace pro mobilní zařízení. Prostředí PhoneGap a jQuery Mobile vám dovolí psát aplikace, které budou přenosné na různé mobilní platformy, budou skvěle vypadat a navíc budou schopné využívat specifické vlastnosti konkrétních zařízení. Pomocí PhoneGap je možné aplikaci v HTML5 zabalit tak, že bude plně použitelná jako nativní aplikace a bude možné ji distribuovat do marketů. Jednoduše řečeno můžete vyvíjet pro všechny dnes dostupné mobilní platformy v rámci obrovsky úspěšného frameworku PhoneGap a pomocí jQuery Mobile, to vše v JavaScriptu a HTML5.

O autorovi

Jiří Vávrů je programátor mobilních aplikací pro iPhone a Android založených na platformě HTML5 a JavaScript a na frameworku PhoneGap. Vývoji mobilních aplikací a mobilních webů se věnuje dva roky a vytvořil několik aplikací na výše uvedených platformách. Dále se věnuje vývoji mobilních aplikací za použití frameworku Sensa Touch a jeho oblíbený UI framework je Kendo UI a jQuery Mobile. Jeho další oblíbený programovací jazyk je PHP, především pak framework CakePHP, na kterém vytváří API rozhraní pro serverové části mobilních aplikací.

Koncept knihy

Kniha je pojata jako návod pro ty vývojáře, kteří by rádi začali vytvářet své vlastní mobilní aplikace pro Android, jednu z nejrozšířenějších mobilních platforem současnosti, a jeho zařízení. Kniha je koncipována tak, aby uživatel, který má základní znalosti v oblasti JavaScriptu a HTML5, byl schopen vytvořit krok za krokem funkční mobilní aplikaci a nahrát ji do Google Play – ať už k prodeji, či ji chce nabídnout zdarma. To vše bez znalostí nativního programování pro platformu Android. Kniha je vhodná i pro pokročilé vývojáře, kteří již mají s JavaScriptem a HTML5 zkušenosti a chtěli by se naučit použít své předchozí znalosti pro vývoj mobilních aplikací pro Android a další platformy a také se dozvědět spoustu užitečných informací o uživatelském rozhraní, optimalizaci aplikace pro běh na reálném zařízení či rozšiřujících pluginech (a ušetřit tak obrovské množství času). Vše potřebné tato publikace obsahuje.

Kniha vás provede krok za krokem několika kapitolami, od základního popisu jednotlivých nástrojů pro vývoj přes návrh UseCase aplikace a samotný vývoj až po návod pro nasazení aplikace do Google Play. Na konci knihy naleznete referenční manuál UI frameworku jQuery Mobile, a budete tak schopni vytvářet kvalitní mobilní aplikace pro několik platforem současně, to vše za použití JavaScriptu a HTML 5.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu připravilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press
Albatros Media a.s., pobočka Brno
IBC
Příkop 4
602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Zdrojové kódy ke knize

Z adresy *http://knihy.cpress.cz/K2055* si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo v kódu, budeme rádi, pokud nám ji oznámíte. Ostatní uživatele tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese *http://knihy.cpress.cz/K2055* po klepnutí na odkaz Soubory ke stažení.

ČÁST

VÝVOJ MOBILNÍ APLIKACE

V této části:

- **KAPITOLA 1** – Vývojové prostředí a výběr frameworku
- **KAPITOLA 2** – Vývoj reálné aplikace krok za krokem
- **KAPITOLA 3** – Přihlašování uživatele
- **KAPITOLA 4** – Načítání tweetů: JSON + AJAX a zpracování dat
- **KAPITOLA 5** – Odesíláme tweety na server
- **KAPITOLA 6** – Práce s GPS a Geolokací
- **KAPITOLA 7** – Načítání vlastních tweetů a práce s Google Maps
- **KAPITOLA 8** – Testujeme aplikaci v Android emulátoru
- **KAPITOLA 9** – Testujeme aplikaci v reálném zařízení
- **KAPITOLA 10** – Export apk souboru
- **KAPITOLA 11** – Příprava aplikace a samotný upload aplikace do Android marketu
- **KAPITOLA 12** – Rady na závěr
- **KAPITOLA 13** – Tipy a triky

KAPITOLA 1

Vývojové prostředí a výběr frameworku

V této kapitole:
- PhoneGap
- jQuery Mobile
- Výběr vývojového prostředí
- Emulátory – Android emulátor
- Nastavení a konfigurace
- Jednoduchá aplikace Hi World
- Spuštění v emulátoru

Pro vývoj nativní mobilní aplikace bude v našem případě použito dvou frameworků.

První má název PhoneGap (ve verzi 1.8) a díky jeho vlastnostem je možno přistupovat ke standardním funkcím telefonu, jako jsou například GPS, fotoaparát, souborový systém, akcelerometr a mnohé další.

Dalším frameworkem, který bude použit, je jQuery Mobile (ve verzi 1.1.0). Tento framework nám bude sloužit především k vytváření jednotného a přívětivého uživatelského rozhraní.

Jako vývojové prostředí pro naši aplikaci bude použito Eclipse IDE (je ale možno použít jakýkoliv jiný editor).

PhoneGap

PhoneGap je aplikační platforma založená na JavaScriptu a HTML5, která umožňuje autorům aplikací přistupovat k nativním funkcionalitám mobilních zařízení a tyto aplikace umístit do aplikačních obchodů, v našem případě Android Market. PhoneGap umožňuje vývoj aplikací i pro další platformy, jako iPhone, BlackBerry, Symbian, Windows Phone 7 a Bada, a to vše s minimálními změnami v kódu.

Zjednodušeně je tedy možno říci, že vývojáři stačí aplikaci napsat pouze jednou a poté jen provést nezbytné změny pro nasazení na další platformu. Jistě v tomto sami vidíte obrovskou výhodu, a to především v jednotnosti vývojového procesu a ohromné časové úspoře.

To vše při použití HTML a JavaScriptu.

PhoneGap je open source implementací otevřených standardů, což znamená, že vývojáři a firmy mohou používat PhoneGap pro mobilní aplikace, které jsou zdarma, placené, open source nebo jakákoliv jiná kombinace výše zmíněných.

PhoneGap jako multiplatformní nástroj pro vývoj mobilních aplikací umožňuje přistupovat k funkcím telefonu, tato podpora se platforma od platformy liší. Pro lepší ilustraci je seznam podporovaných funkcí pro Android i další platformy uveden níže v přehledné tabulce.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 5.x	Blackberry OS 6.0+	WebOS	Windows Phone 7	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓	X

Obrázek 1: Seznam podporovaných vlastností framework u PhoneGap

Více informací naleznete na webové adrese *http://PhoneGap.com/*.

jQuery Mobile

Unifikovaný framework pro tvorbu uživatelského mobilního rozhraní využívajícího HTML5 a CSS3 vlastností, určený pro všechny současné populární mobilní platformy. Framework staví na jQuery a jQuery UI. jQuery Mobile přináší také výhodu v lehce přizpůsobitelném a skinovatelném designu pomocí on-line nástroje ThemeRoller (*http://jQuerymobile.com/themeroller/index.php*).

Více informací naleznete na webové adrese *http://jQuerymobile.com/*.

Výběr vývojového prostředí

Samotný aplikační balíček určený pro spuštění aplikace na reálném zařízení Android je standardně nejlepší zkompilovat a vytvořit z vývojového studia Eclipse. Pokud ale Eclipse nevlastníte, nemusíte věšet hlavu. Jelikož aplikační logika aplikace bude psaná pomocí JavaScriptu, je možno velké množství funkcionalit vyzkoušet i rámci emulátoru s názvem Ripple.

S ohledem na maximální možnou použitelnost této publikace budeme nejprve zde uváděné příklady testovat i v rámci zmiňovaného emulátoru Ripple. Aplikační balíček pro testování na reálném zařízení vytvoříme až v pozdějším kroku, jakmile dokončíme základní aplikaci v rámci emulátoru Ripple. Tím taktéž celý proces velmi zrychlíme, jelikož změny v rámci Ripple emulátoru dokážeme rychleji zapracovat, než kdybychom znovu a znovu kompilovali kód v rámci Eclipse. Nyní si ale pojďme představit blíže některá vývojová prostředí.

S ohledem na výše uvedené jsme tedy v současné chvíli zcela nezávislí na výběru našeho editoru. Zde uvedené příklady jsou psány v open source vývojovém prostředí s názvem Eclipse, které je možno stáhnout na adrese *http://www.eclipse.org/downloads/*. Jedná se o open source vývojovou platformu, která je pro většinu lidí známa jako vývojové prostředí (IDE) určené pro programování v jazyce Java. Flexibilní návrh této platformy dovoluje rozšířit seznam podporovaných programovacích jazyků pomocí pluginů. Právě pluginy umožňují toto vývojové prostředí rozšířit například o návrh UML či zápis HTML nebo XML a získat tak velice robustní nástroj pro vývoj mobilních aplikací. Dalším aspektem pro výběr tohoto vývojového prostředí je i ta skutečnost, že za použití pluginu sloužícího vývoji pro platformu Android získáme plnohodnotné vývojové prostředí, včetně emulátoru Android zařízení pro vývoj nativních aplikací i pro tuto platformu. Více informací naleznete na adrese *http://developer.android.com/sdk/index.html*.

Dalším nástrojem, který je možno doporučit, především díky jeho integraci jQuery Mobile, je nástroj od firmy Adobe s názvem DreamWeaver. Jeho předností je již zmiňovaná integrace jQuery Mobile společně s dobře fungujícím IntelliSense doplňováním kódu a v neposlední řadě také souběžné zobrazení editovaného kódu a zároveň okamžité zobrazení provedených změn v designu aplikace a export aplikačního balíčku pro iPhone i Android přímo z DreamWeaveru. Nutno podotknout, že tento nástroj je placený, 30denní testovací verze je ale ke stáhnutí zdarma. Více informací naleznete na adrese *http://www.adobe.com/cz/products/dreamweaver.html*.

Emulátory – Android emulátor

V rámci Eclipse studia je možno testovat aplikace na Android emulátoru v rámci Android SDK. Pro každé emulované zařízení je možno mít samostatné nastavení.

Obrázek 2: Android emulátor v rámci Android SDK prostředí

Emulátor Ripple

Ripple je multiplatformní mobilní emulátor, který je vyroben na míru pro vývoj a testování mobilních aplikací pomocí HTML5. Ripple si klade za cíl eliminovat úskalí, jimž čelí mobilní vývojáři díky roztříštěnosti dnešní platformy na trhu s mobilními zařízeními.

Ripple je zaměřen mimo jiné také na práci s frameworkem PhoneGap, který, jak již bylo uvedeno, nám umožňuje pracovat s nativními funkcemi dnešních chytrých telefonů, a to vše v rámci JavaScriptu a HTML5!

Emulátor Ripple také umožňuje využití stávajících nástrojů k provádění ladění JavaScriptu, HTML DOM inspekce, automatizovaného testování, stejně tak jako testování na více zařízeních a rozlišení displeje v reálném čase, bez nutnosti znovu restartovat emulátor.

Díky Ripple máte možnost simulovat funkce jako GPS, kompas, akcelerometr, datové připojení, polohu displeje a další. Emulátor jako takový je nabízen zdarma a jeho používání je velice intuitivní a jednoduché.

Emulátor je dostupný ve dvou provedeních, v první jako plugin do Google Chrome prohlížeče, který jednoduše nainstalujete z tohoto odkazu:
https://chrome.google.com/webstore/detail/geelfhphabnejjhdalkjhgipohgpdnoc.

V dalším provedení jako samostatná aplikace, která je ke stáhnutí na tomto odkazu:

https://bdsc.webapps.blackberry.com/HTML5/download/.

Obrázek 3: Ripple emulátor v rámci prohlížeče Google Chrome

Nastavení a konfigurace

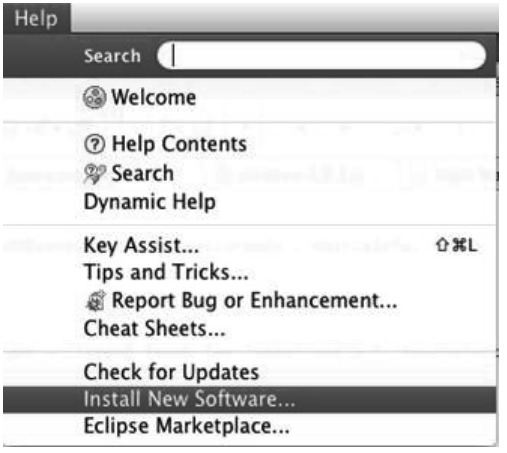
Abychom mohli naši první aplikaci otestovat, potřebujeme si připravit nový projekt a nastavit si vše potřebné. Připravíme si tedy podklady pro náš první projekt (dále jen workspace). Pojďme se podívat, co k tomu budeme potřebovat a co je nutné udělat.

Stáhnutí a instalace Eclipse – Eclipse je možno stáhnout přímo z oficiálních stránek, a to pomocí odkazu; je doporučeno stáhnout verzi pro Java Developers (*http://www.eclipse.org/downloads/*). Poté nainstalujeme klasickým způsobem.

Instalace Android SDK a ADT – Stáhněte a nainstalujte následující:

Pro naše potřeby je nutné nainstalovat ADT plugin (Android Development Tools), což je Eclipse plugin, jenž poskytuje kompletní IDE pro vývoj aplikací Android. Za použití ADT je možno debugovat aplikaci a spouštět emulátor.

Instalaci provedete tak, že kliknete na položku **„install new software“** z menu Help, viz obrázek.



Obrázek 4: Výběr položky z menu pro instalaci ADT

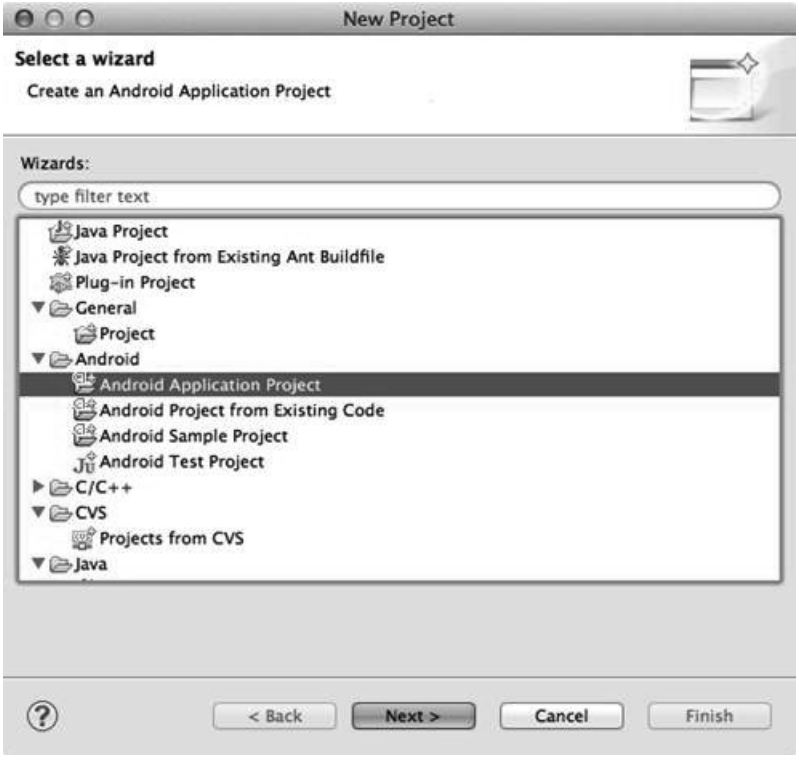
Pro instalaci poté vložte URL: *http://dl-ssl.google.com/android/eclipse/* a poté pokračujte pomocí průvodce.

Jakmile máte ADT nainstalováno, bude třeba nainstalovat platformu Android a ostatní komponenty. To provedeme pomocí:

Window → Android DK and AVD manager, zvolíme platformu a API level.

1. **Stáhnutí a instalace PhoneGap** – PhoneGap plugin nalezneme ke stáhnutí na adrese *http://PhoneGap.com/download*. V závislosti na použitém operačním systému je nabídnut ke stáhnutí balíček, který si poté rozbalíme u sebe na disku.
2. **Založení nového projektu** – Z menu zvolíme položku **New Project:**

Poznámka: Pokud máte raději automatizovaný způsob, který provede níže uvedené operace za vás, je možno si stáhnout plugin pro Eclipse s názvem MDS AppLaud – PhoneGap for Android. Ten umožňuje udělat níže uvedené kroky zcela automatizovaně a rovnou do projektu volitelně integruje i například knihovny jQuery Mobile. V případě použití tohoto pluginu můžete následující kroky přeskočit.

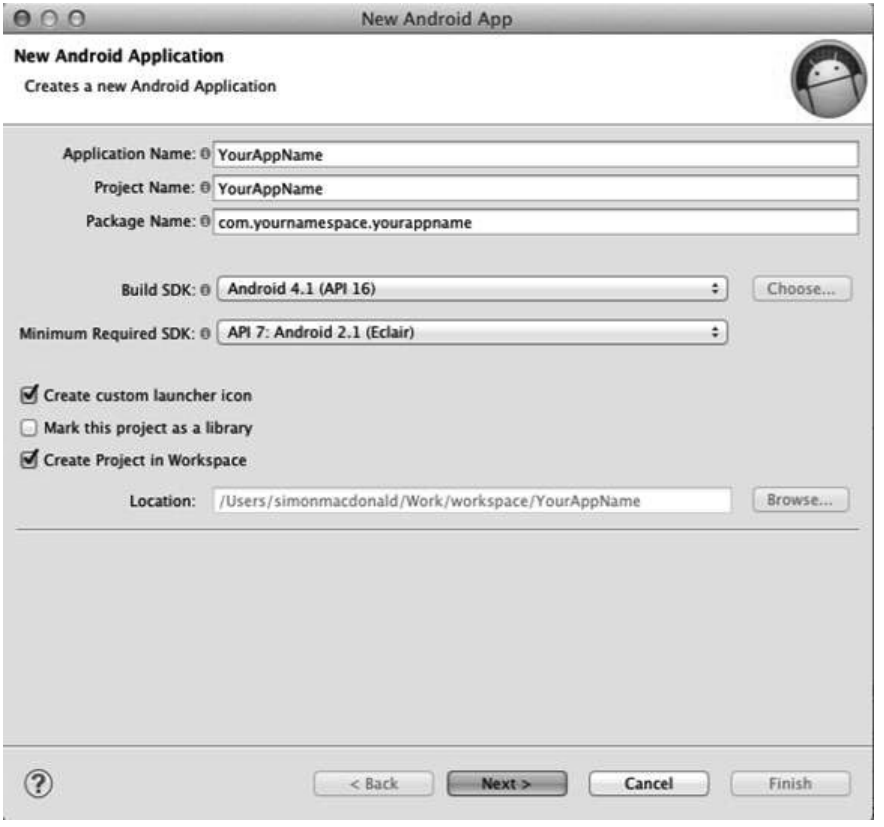


Obrázek 5: Založení nového projektu Android v Eclipse

Instalaci pluginu provedete opět přímo v Eclipse, url pluginu je:

http://svn.codespot.com/a/eclipselabs.org/mobile-web-development-with-phonegap/tags/r1.2/download

Poté specifikujeme základní údaje o novém projektu:

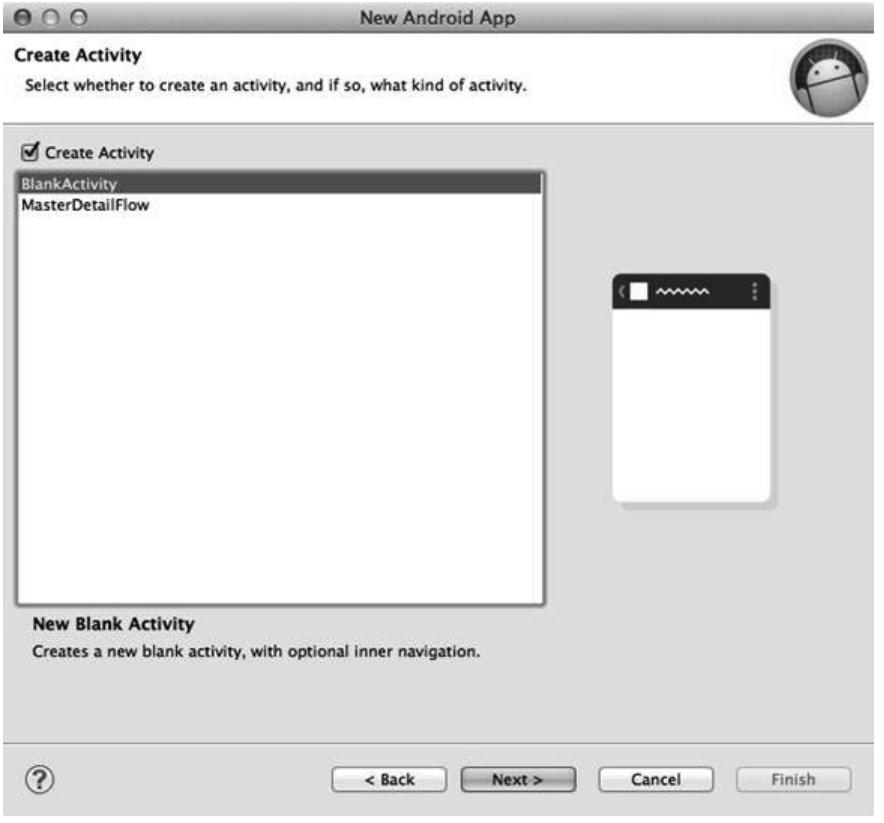


Obrázek 6: Založení nového projektu Android v Eclipse

Volitelně můžeme zvolit grafiku (můžeme upravit kdykoliv později).

Obrázek 7: Nastavení ikon projektu Android v Eclipse

Vytvoříme prázdnou aktivitu:



Obrázek 8: Vytvoření prázdné aktivity

Ujistěte se, že aktivity nedědí z ničeho. Jakmile je to hotovo, klepněte na tlačítko **Finish**.

Obrázek 9: Informace o aktivitě

V rootu projektu vytvořte dva nové adresáře:

- `/libs`
- `/assets/www`

Poznámka: V adresáři www budou uloženy veškeré soubory úzce související s naší aplikací (javaScriptové knihovny, externí soubory atd.).

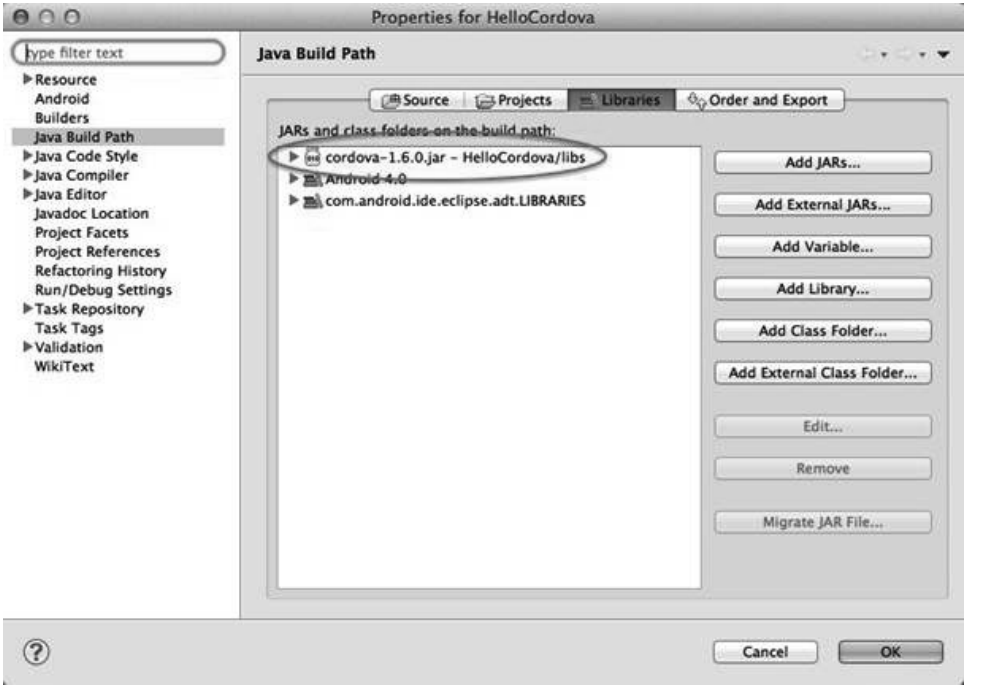
Zkopírujte soubor `cordova-1.8.1.js` z balíčku PhoneGap, který jsme si stáhli dříve, a zkopírujeme jej do adresáře /assets/www, který jsme si před chvílí vytvořili, viz obrázek 10:

Obrázek 10: Struktura frameworku PhoneGap

Zkopírujte soubor `cordova-1.8.1.jar` z balíčku PhoneGap, který jsme si stáhli dříve do adresáře /libs, který jsme si před chvílí vytvořili.

Z balíčku PhoneGap, který jsme si stáhli, zkopírujte adresář xml do adresáře /res ve vašem projektu.

Ověřte, že `cordova-1.8.1.jar` je nadefinována jako „Build path" pro váš projekt. Pravým tlačítkem klikněte na /libs adresář a běžte na položky **Build Paths/ → Configure Build Path**. Poté přejděte do tabu **Libraries** a zde přidejte soubor `cordova-1.8.1.jar` do vašeho projektu.



Obrázek 11: Přidání jar knihovny PhoneGap do projektu Android

Pokud se změna neprojeví, zkuste po kliknutí na ikonu projektu aktualizovat projekt pomocí klávesy F5.

Upravte hlavní Java soubor projektu v adresáři /src v rámci Eclipse projektu.

Přidejte import org.apache.cordova.*;

Změňte zápis extend z Activity na DroidGap.

Nahraďte `setContentView()` řádek zápisem `super.loadUrl("file:///android_asset/www/index.html")`.

Obrázek 12: Úprava hlavního Java souboru projektu pro načtení souboru index.html z adresáře assets/www/ a přidání jar knihovny PhoneGap do projektu Android

Pravým tlačítkem klikneme na soubor AndroidManifest.xml a zvolíme **Open With → Text Editor**.

Vložíme následující přístupové definice mezi tagy `<uses-sdk.../>` a `<application.../>`.

```
<supports-screens
   android:largeScreens="true"
   android:normalScreens="true"
   android:smallScreens="true"
   android:resizeable="true"
   android:anyDensity="true" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="
    android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name=
```

```
    "android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name=
    "android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
```

Poznámka: Přidáváte seznam oprávnění pro vaši aplikaci. Oprávnění, která aplikace nepoužívá, by měla být odstraněna před odesláním aplikace do Google Play.

Podporu změny orientace displeje do své aplikace přidáte vložením následujícího kódu dovnitř tagu `<activity>`:

```
android:configChanges=
  "orientation|keyboardHidden|keyboard|screenSize|locale"
```

Soubor `manifest.xml` by měl nyní vypadat jako na obrázku 13.

Tímto krokem jsme úspěšně založili nový PhoneGap projekt pro Android v rámci Eclipse studia. Ověření, že vše proběhlo v pořádku a náš projekt obsahuje soubory, které potřebujeme, provedeme tak, že na našem projektu nalezneme adresář s názvem **www (assets/www/)**, který by měl obsahovat soubor s názvem `index.html` (pokud soubor uveden není, vytvoříme ho na daném umístění). Pokud jste soubor nalezli, je vše připraveno k vytvoření naší první aplikace a otestování v simulátoru. O tom bude pojednávat následující podkapitola.

Obrázek 13: Soubor Manifest.xml

Jednoduchá aplikace Hi World

V této podkapitole bude popsána velmi jednoduchá aplikace a její otestování v emulátoru Android.

Hello World, či v našem případě Hi World, je aplikace, která vypíše na výstupní zařízení (v našem případě displej telefonu) text „Hi, world!“ Aplikace „Hi World“ se používá jako ukázka při výuce programování v určitém programovacím jazyce a mnoho studentů ji píše jako svůj první malý program.

V prvním kroku nalezneme v rámci našeho projektu adresář s názvem **www** a v něm soubor s názvem `index.html`.

Tento soubor otevřeme buďto přímo v editoru Eclipse, popřípadě v jakémkoliv jiném textovém editoru.

Soubor `index.html` nahradíme níže uvedeným kódem.

Jedná se o klasickou HTML stránku, která obsahuje klasické HTML tagy a několik meta tagů.

Kód je okomentovaný, aby byl dobře pochopitelný i pro začátečníky.

```
<!DOCTYPE HTML>
<HTML>
  <head>
    <title>Titulek stránky</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
  <body>
    <h1>Hi, World!</h1>
  </body>
</HTML>
```

Níže jsou uvedeny vysvětlivky k jednotlivým tagům.

```
<!DOCTYPE HTML>:
<!-- Doctype definice HTML dokumentu, jedná se o nepárový tag -->
<HTML>:
<!-- Uvozovací značka HTML dokumentu -->
<head>:
<!-- Hlavička HTML dokumentu, bude obsahovat všechny v budoucnu obsažené
  skripty (PhoneGap, jQuery Mobile a další) -->
<title>:
<!-- Titulek stránky, při použití kódu pro nativní mobilní aplikace se
  jedná o téměř nepodstatný tag, situace by se změnila, pokud by se jednalo
  o mobilní web, v tom případě by tento tag měl relativně velký význam -->
<meta name="viewport":
<!-- Toto nastavení říká prohlížeči, jak má rozmístit obsah na displej,
  a informuje prohlížeč o tom, že stránka je optimalizovaná pro mobily. Náš
  příklad říká prohlížeči, aby nastavil viewport na šířku zařízení
  a zachoval měřítko -->
<body>:
<!-- Uvnitř tohoto tagu se bude vyskytovat všechen viditelný obsah stránky
  naší mobilní aplikace. Ať už to tedy bude hlavička stránky, tlačítka,
  seznamy, vše bude obsaženo uvnitř tohoto tagu. V tuto chvíli obsahuje
  pouze nápis "Hi, World!" obalený v tagu nadpisu <h1>. Tento nadpis tak
  bude jediná viditelná část z naší aplikace, jakmile aplikaci spustíme
  v emulátoru -->
```

Náš nový kód uložíme a přejdeme ke spuštění v emulátoru.

Poznámka: Zájemcům o hlubší problematiku mobilního webu v HTML5 doporučuji k pročtení článek Mobilizujeme web v HTML5 na webu Zdroják.cz, více na *http://www.zdrojak.cz/clanky/mobilizujeme-web-v-HTML5/*.

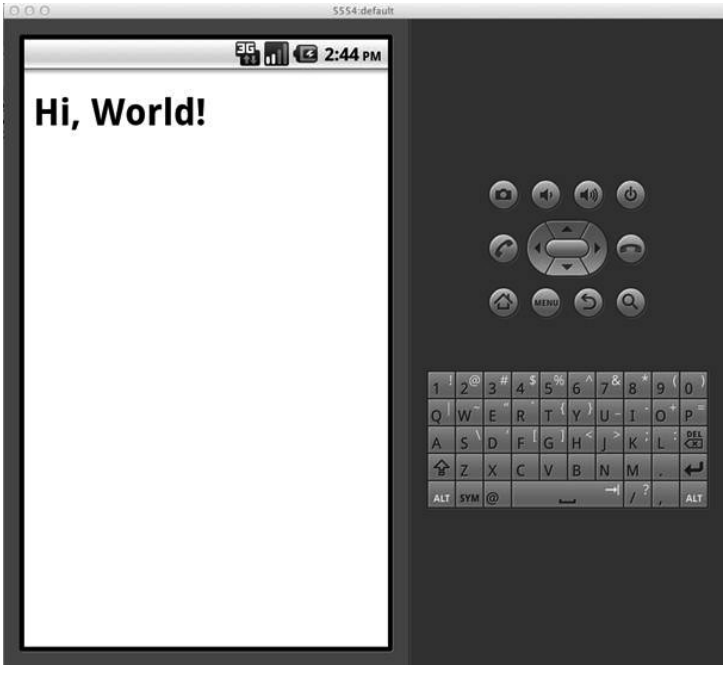
Spuštění v emulátoru

Eclipse

Pravým tlačítkem klikněte na ikonu projetu a zvolte položku **Run As → Android Application**.

Eclipse se vás zeptá na volbu virtuálního zařízení (AVD). Pokud žádné nemáte nastaveno, je třeba jej vytvořit.

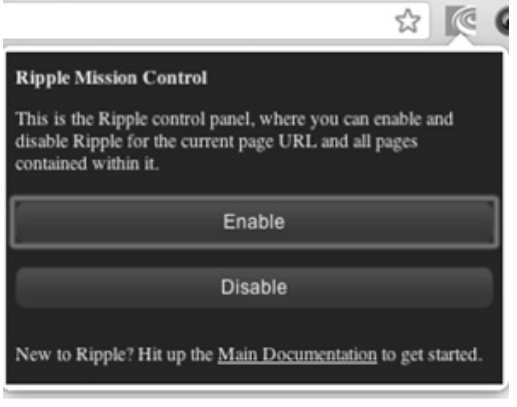
Po stisknutí tohoto tlačítka se spustí sestavení projektu a během několika okamžiků byste měli vidět svou **první aplikaci v emulátoru Android**, viz obrázek 14.



Obrázek 14: Ukázka první aplikace v emulátoru Android prostředí Eclipse

Emulátor Ripple

1. Klikněte pravým tlačítkem na soubor `index.html` a z menu zvolte položku Run as → JavaScript Web Application. Načte se klasická HTML stránka ve vašem výchozím prohlížeči. URL této stránky zkopírujte do adresního řádku Chrome prohlížeče, ve kterém máte nainstalován Ripple plugin.
2. Nyní stiskněte tlačítko pro aktivaci emulátoru Ripple, které je umístěno vpravo nahoře, viz obrázek 15.
3. Z nabídky zvolte možnost **enable** a z výběru prostředí pro běh emulátoru zvolte **PhoneGap**.
4. Inicializace emulátoru může chvilku trvat, a pokud jste vše udělali správně, měli byste svůj první projekt vidět v emulátoru Ripple, viz obrázek 16.



Obrázek 15: Aktivace emulátoru Ripple

Obrázek 16: Aplikace Hi World v prostředí emulátoru Ripple

KAPITOLA 2

Vývoj reálné aplikace krok za krokem

V této kapitole:
- Vlastnosti a popis aplikace
- Use Case diagram aplikace
- Založení nového projektu
- Import důležitých knihoven pro běh aplikace
- Úvodní strana – rozložení ovládacích prvků
- Inicializace aplikace a ověření funkčnosti

Vlastnosti a popis aplikace

Aplikaci Hi World máme úspěšně za sebou. Bohužel, aplikace kromě spuštění sebe sama a vypsání textu na displej neumí nic jiného. Byla by tedy škoda si nevytvořit aplikaci, jejíž funkcionalita bude přece jen na vyšší úrovni.

Jako vhodnou aplikaci zvolíme klienta pro sociální síť Twitter. Tato aplikace byla zvolena proto, že v rámci jejího vývoje si procvičíme nejpoužívanější funkcionality dnešních mobilních aplikací, zejména práci s technologiemi:

- JSON a AJAX
- GPS
- Google Maps

V neposlední řadě si procvičíme také práci s lokálním úložištěm.

Aplikace tedy bude kompletně funkční Twitter klient, který bude umožňovat následující funkcionality:

- Přihlášení uživatele pod svůj uživatelský účet na Twitteru
- Získání nových příspěvků (tweetů) od ostatních uživatelů
- Odeslání nového příspěvku na Twitter včetně GPS souřadnic uživatele
- Vizualizace příspěvků v rámci Google Maps

V průběhu celého procesu vytváření aplikace budete moci také nahlédnout do referenčního manuálu jQuery Mobile na konci této knihy.

Manuál jako takový slouží k popisu jednotlivých funkcí frameworku jQuery Mobile.

Pojďme se tedy pustit do práce na naší reálně použitelné mobilní aplikaci.

Use Case diagram aplikace

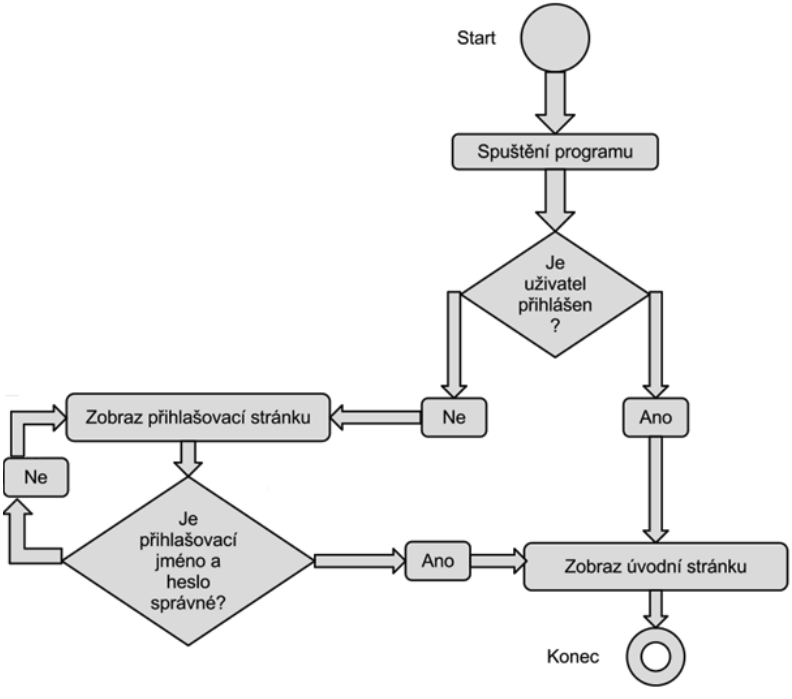
Před samotným programováním aplikace je třeba si navrhnout takzvaný **Use Case** diagram aplikace, který nám slouží k zobrazení dynamické (funkční) struktury systému z pohledu uživatele. Je primárně určen k definici chování systému, aniž by odhaloval jeho vnitřní strukturu.

Na níže uvedeném diagramu si tedy zjednodušeně ukážeme, jak v naší budoucí aplikaci bude fungovat přihlášení uživatele pod účet Twitter.

Use Case přihlášení uživatele

1. Po spuštění aplikace je uživatel vyzván k zadání svých přihlašovacích údajů.
2. Pokud už své přihlašovací údaje zadal dříve (již se úspěšně přihlásil), je automaticky přesměrován na hlavní stránku aplikace.
3. Pokud se přihlášení nezdařilo, je uživateli zobrazena zpráva.

Use Case přihlášení by tedy vypadal zhruba jako na následujícím obrázku.

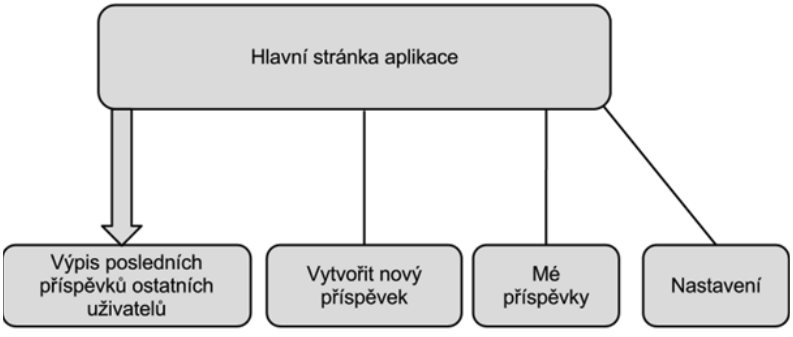


Obrázek 17: Diagram pro přihlášení

Diagram by mohl být samozřejmě daleko komplexnější a pokrývat stavy jako nezadané vstupní hodnoty, off-line režim a další. Tyto stavy ale nejsou pro pochopení principu **Use Case** potřeba a ošetříme je přímo v kódu aplikace.

Use Case hlavní stránky aplikace

Hlavní stránka aplikace bude plnit úlohu jakéhosi rozcestníku naší aplikace. Jak můžete vidět na diagramu níže, na stránce se bude zobrazovat seznam posledních příspěvků (tweetů). Stránka bude dále obsahovat jednoduché menu, pomocí kterého se budeme moci přesunout na stránku pro vložení nového příspěvku na náš Twitter, zobrazení nastavení aplikace a historie našich příspěvků.



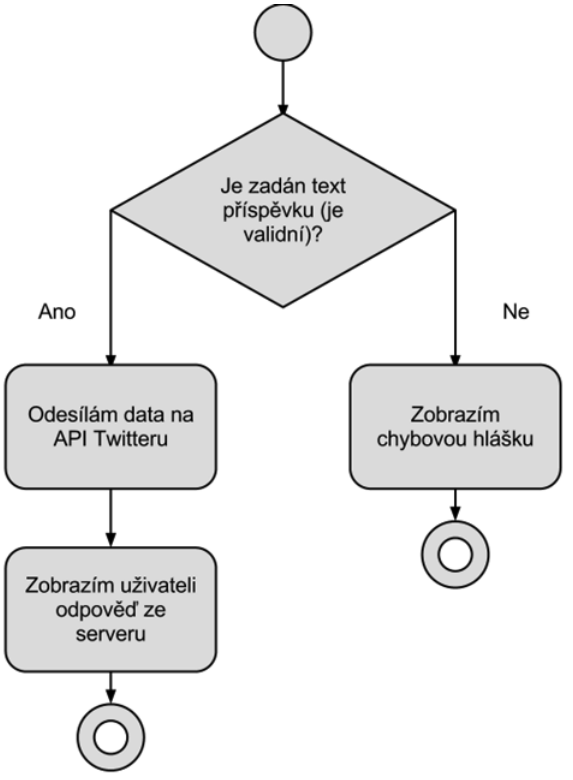
Obrázek 18: Diagram pro hlavní stránku aplikace

Use Case přidání nového příspěvku

V tomto **Use Case** bude hlavní funkčnost zaměřena na zadání textu nového příspěvku, jeho ověření před odesláním, odeslání a získání odpovědi ze serveru a informování uživatele o výsledku akce.

Use Case by v tomto případě měl následující průběh:

1. Uživatel klikne na tlačítko **Odeslat příspěvek**.
2. Aplikace ověří, zda byl zadán text; pokud ne, aplikace informuje uživatele.
3. Pokud je vše v pořádku, text v podobě statusu se s dalšími potřebnými informacemi odešle na API Twitteru.
4. Server vrátí odpověď o výsledku akce v podobě JSON výstupu.
5. Odpověď zpracujeme a o výsledku informujeme uživatele.



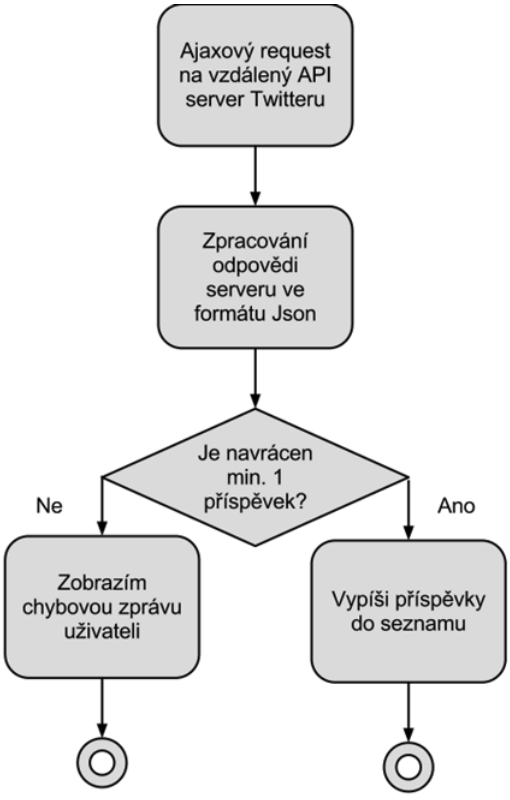
Obrázek 19: Diagram přidání nového tweetu

Use Case výpisu historie mých příspěvků

V tomto případě aplikace vypíše seznam 10 posledních příspěvků, které jsme na Twitter vložili.

Funkční stránka aplikace bude následující:

1. Na vzdálený server Twitteru (API) zašleme pomocí AJAX requestu žádost o navrácení posledních deseti námi vložených příspěvků.
2. Odpověď nám bude navrácena v JSON formátu, který poté aplikace zpracuje a vypíše jako přehledný výpis do komponenty ListView.



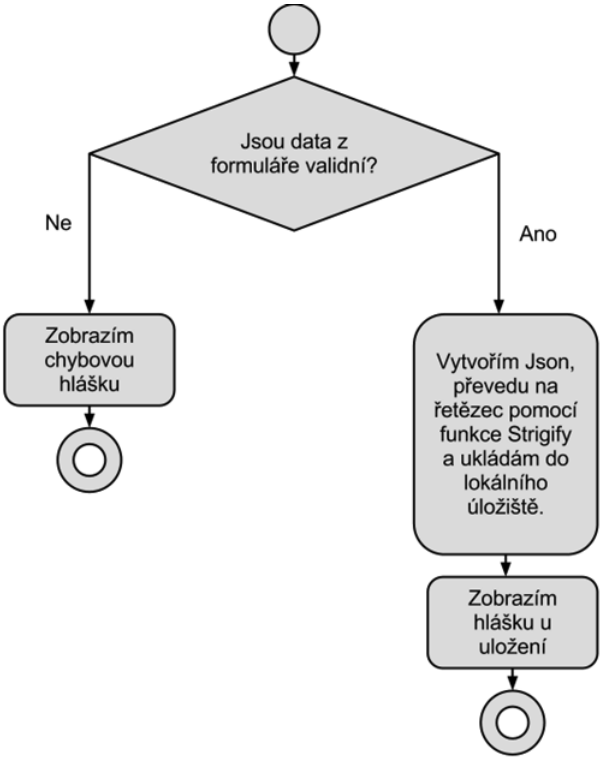
Obrázek 20: Diagram pro výpis historie

Use Case nastavení aplikace

Tento **Use Case** má znázornit fungování stránky s uživatelským nastavením.

Stránka bude obsahovat formulář s přepínači a tlačítky pro změnu hodnot nastavení.

Po stisku tlačítka s názvem **Uložit nastavení** se data převzatá z formuláře uloží ve tvaru JSON a pomocí JavaScriptové metody stringify se uloží do lokálního úložiště v telefonu, odkud bude možno data kdykoliv po rozparsování opět číst.



Obrázek 21: Diagram pro nastavení

Tímto krokem jsou základní **Use Case** diagramy popsány.

Založení nového projektu

Stejně jako v kapitole 1.6 musíme pro tuto aplikaci vytvořit nový projekt s názvem **Twitter klient.** Jak již bylo zmíněno, aplikace bude umožňovat následující:

- přihlášení pod uživatelský Twitter účet,
- načítání aktuálních tweetů ostatních uživatelů,
- přidávaní nových tweetů.

Pokud nemáme ještě stáhnutý framework PhoneGap, který budeme pro naše budoucí použití potřebovat, provedeme tak na webové adrese *http://PhoneGap.com/download*.

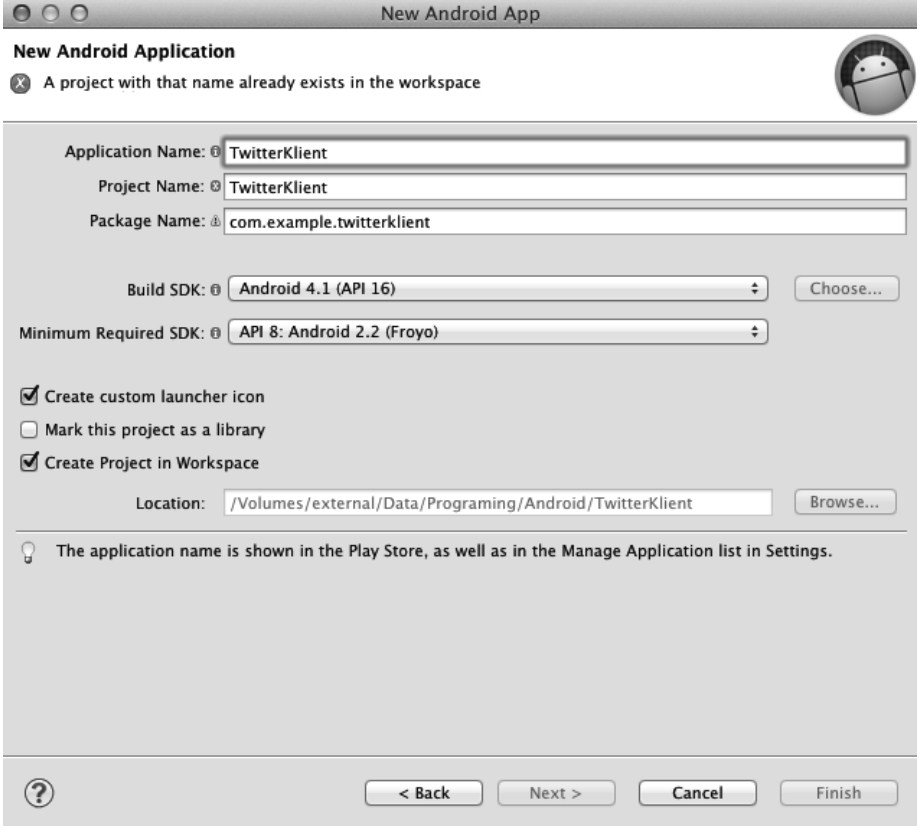
V dalším kroku založíme nový projekt v prostředí Eclipse:

1. Spustíme Eclipse.
2. Klikneme na položku **File** v horní liště Eclipse.

3. Zvolíme položku **New** a poté klikneme na položku **Project**.
4. Ze zobrazené nabídky přednastavených šablon zvolte **Android** a poté položku s názvem **Android Application Project**.
5. Klikněte na tlačítko **Next**.

V dalším kroku vyplňte položky **Application Name, Project Name** a **Package Name** pro naši aplikaci.

- **Application name** by měl obsahovat název aplikace, v tomto případě **TwitterKlient**.
- Project Name slouží pouze pro potřeby Eclipse a musí být unikátní v rámci workspace; můžeme opět ponechat **TwitterKlient.**
- Položka **Package Name** by měla být ve formátu **com.nazevaplikace.nazevFirmyCiVyvojare** a neměla by obsahovat žádné mezery.



Obrázek 22: Okno nového projektu

Klikneme na tlačítko **Next** a pokračujeme stejně jako při vytváření první ukázkové aplikace, tzn. přidáme adresáře a potřebné soubory frameworku PhoneGap, tak aby adresář www obsahoval minimálně soubor `index.html`.

Import důležitých knihoven pro běh aplikace

Jak již bylo uvedeno na začátku, naše aplikace bude pro svůj běh kromě frameworku **PhoneGap** (který nám umožňuje přistupovat k funkcím telefonu) potřebovat i import javascriptových knihoven **jQuery** a **jQuery Mobile**, které nám umožní vytvořit uživatelsky a graficky příjemné prostředí pro naši aplikaci.

Tyto knihovny budou nalinkovány v hlavičce `index.html` souboru naší aplikace.

Začneme tedy následujícím postupem:

- Ze stránek *http://jQuerymobile.com/download/* si stáhneme zip balíček `jQuery.mobile-1.1.0.zip`, který obsahuje všechny potřebné soubory pro vytváření mobilního UI. Tyto soubory poté vložíme do hlavičky našeho souboru `index.html`.
- Abychom jQuery Mobile mohli plně využívat, je třeba stáhnout ještě klasickou javascriptovou knihovnu jQuery ze stránek jQuery.com, konkrétně knihovnu z odkazu *http://code.jQuery.com/jQuery-1.7.2.js*.

Tato knihovna je potřebná k běhu naší aplikace a obsahuje důležité funkce pro práci s JSON, AJAX requesty a mnohé další.

Balíček jQuery Mobile si rozbalíme do libovolné složky na disku a zároveň se přepneme do našeho projektu v Eclipse.

Začneme importem souborů knihovny jQuery Mobile a jQuery do našeho projektu.

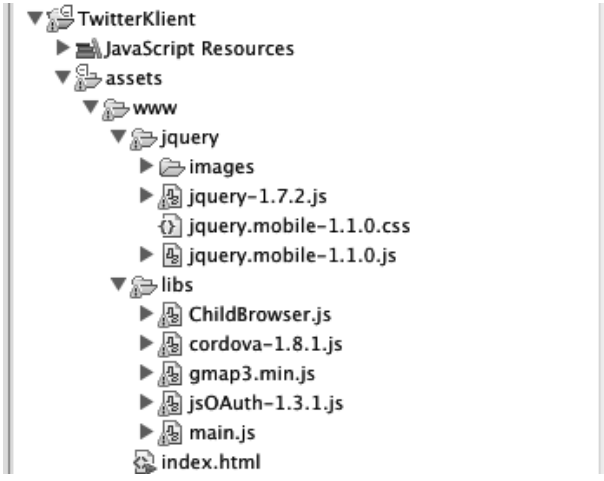
V inspektoru našeho projektu v Eclipse otevřeme adresář www, ve kterém je náš projekt umístěn, a vytvoříme v něm novou složku s názvem jQuery. Do této složky zkopírujeme následující soubory a adresáře z balíčků, které jsme před chvílí stáhli na disk:

- Adresář `/images`
- soubor `jQuery.mobile-1.1.0.css`
- soubor `jQuery.mobile-1.1.0.js`

Adresář **images** obsahuje ikony a grafiku uživatelského prostředí mobilní aplikace, `jQuery.mobile-1.1.0.css` pro změnu obsahuje CSS stylopis uživatelského prostředí a soubor `jQuery.mobile-1.1.0.js` obsahuje javascriptové funkce typické pro jQuery Mobile.

Do námi vytvořeného adresáře zkopírujeme také staženou knihovnu jQuery, tzn. `jquery-1.7.2.js`.

Výsledná struktura našeho adresáře by tedy nyní měla vypadat jako na obrázku 23.



Obrázek 23: Detail projektu

Nyní provedeme import výše uvedených knihoven do našeho projektu. Po otevření souboru `index.html` vložíme mezi tag `<head>` následující kus zdrojového kódu:

```
<link rel="stylesheet" href="jQuery/jQuery.mobile-1.1.0.css" type="text/css"
media="screen" title="no title">
<script type="text/javascript" charset="utf-8"
  src="jQuery/jQuery-1.7.2.js"></script>
<script type="text/javascript" charset="utf-8" src=
  "jQuery/jQuery.mobile-1.1.0.js"></script>
```

Poznámka: Soubor `jQuery-1.7.2.js` musí být v zápisu uveden vždy před uvedením knihovny jQuery Mobile, jelikož tvoří její jádro.

Pokud bychom toto neprovedli, jQuery Mobile by nefungovalo korektně.

Celkově náš `index.html` soubor bude v tuto chvíli vypadat následně (zdrojový kód můžete zkopírovat a použít jako šablonu):

```
<!DOCTYPE HTML>
<HTML>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title> Twitter klient</title>
    <!-- JQUERYMOBILE CSS STYLOPIS ZE SLOZKY /JQUERY -->
    <link rel="stylesheet" href="jQuery/jQuery.mobile-1.1.0.css"
      type="text/css" media="screen" title="no title">
<!-- PHONEGAP KNIHOVNA ZE SLOZKY /LIBS -->
<script type="text/javascript" charset="utf-8"
  src="libs/cordova-1-7.0.js"></script>
```

```
<!-- JQUERY KNIHOVNA ZE SLOZKY /JQUERY POZOR!! MUSI BYT PRED JQM-->
<script type="text/javascript" charset="utf-8"
  src="jQuery/jQuery-1.7.2.js"></script>
<!-- JQUERYMOBILE KNIHOVNA ZE SLOZKY /JQUERY -->
<script type="text/javascript" charset="utf-8"
  src="jQuery/jQuery.mobile-1.1.0.js"></script>
  </head>
<body>
</body>
</HTML>
```

Úvodní strana – rozložení ovládacích prvků

Aplikace je připravena pro vytvoření první stránky, a to stránky úvodní (viz **Use Case** aplikace), která bude obsahovat navigační tlačítka, výpis tweetů a další.

Na začátek by se ale hodilo vysvětlit, jak to celé funguje v rámci **jQuery Mobile**.

Každá jednotlivá stránka v **jQuery Mobile**, se kterou budeme pracovat, má přesně danou strukturu.

```
<div data-role="page">
   <div data-role="header">...</div>
   <div data-role="content">...</div>
   <div data-role="footer">...</div>
</div>
```

Tato struktura vychází z poznatku, že každá stránka mobilní aplikace v rámci jQuery Mobile se skládá ze tří `<DIV>` tagů, z nichž každý má svoji specifickou roli přiřazenou dle hodnoty atributu `data-role`.

data-role="header" – je používán například pro zobrazování názvu stránky, navigační tlačítka atd.

data-role="content" – je používán pro vlastní obsah aplikace, v našem případě například pro výpis aktuálních tweetů do komponenty List-view.

data-role="footer" – používá se pro umístění dalších navigačních prvků jako spodní menu, stavová lišta atd.

Každý z těchto uvedených `data-role` atributů má samozřejmě široké spektrum kombinací a použití a je jen na naší kreativitě, jak tyto kombinace využijeme.

Další změnou oproti klasickému zvyku při vytváření aplikací pomocí **HTML** je to, že všechny jednotlivé stránky naší aplikace budou součástí jednoho souboru. To znamená, že náš souboru `index.html` bude obsahovat více jednotlivých stránek obalených v tagu `<body>`, viz příklad níže:

```
<body>
<!-- Začátek první stránky -->
<div data-role="page" id="stranaJedna">
```

```
  <div data-role="header">
    <h1>Nadpis stránky 1</h1>
  </div><!-- /header -->
  <div data-role="content">
    <p>Text zobrazený ve stránce s id stranaJedna</p>
  <!-- /content -->
  <div data-role="footer">
    <h4>Patička stránky 1</h4>
  </div><!-- /footer -->
</div><!-- /page -->
<!-- Začátek druhé stránky -->
<div data-role="page" id="stranaDva">
  <div data-role="header">
    <h1>Nadpis stránky 2</h1>
  </div><!-- /header -->
  <div data-role="content">
    <p>Text zobrazený ve stránce s id stranaDva</p>
  </div><!-- /content -->
  <div data-role="footer">
    <h4>Patička stránky 2/h4>
   </div><!-- /footer -->
</div><!-- /page -->
</body>
```

Tento způsob nám přináší několik výhod, například možnost měnit vlastnosti ostatních stránek pomocí JavaScriptu, potažmo jQuery, bez nutnosti znovu načítat stránku či například přistupovat k vlastnostem jednotlivých stránek a jejich prvků pomocí jednoznačného ID každé z nich.

Příklad takové stránky je také možno vyzkoušet například na této adrese: *http://jQuerymobile.com/demos/1.1.0/docs/pages/multipage-template.html*.

Je samozřejmě možné používat také klasický způsob a mít každou obrazovku aplikace jako samostatný HTML soubor, nicméně pokud se jedná o mobilní aplikaci, předpokládá se, že její rozsah bude několik základních obrazovek, tudíž je možné udržovat velice přehledný kód i v rámci jednoho souboru.

Abychom mohli vytvořit úvodní stránku našeho **Twitter klienta**, která bude sloužit jako jakýsi rozcestník, a dosáhli výsledku jako na obrázku níže, je třeba zapsat tento **HTML** kód do naší první stránky v souboru `index.html` mezi tagy `<body> a </body>`:

```
<!-- UVODNI STRANKA S ID tweetsFromOthers,
  SLOUZI JAKO UVODNI STRANKA APLIKACE -->
<div data-role="page" id="tweetsFromOthers">
<!-- HEADER S NADPISEM -->
<div data-theme="a" data-role="header">
  <h3>
    Twitter klient
```

```
  </h3>
<!-- HORIZONTALNI MENU APLIKACE -->
  <div data-role="navbar" data-iconpos="bottom">
    <ul>
<!-- POLOZKA MENU -->
      <li>
        <a href="#tweetsFromOthers" data-theme="b" data-icon="refresh"
          class="ui-btn-active">
          Výpis
        </a>
      </li>
      <li>
        <a href="#addNewtweet" data-theme="b" data-icon="plus">
          Nový tweet
        </a>
      </li>
      <li>
        <a href="#mytweets" data-theme="b" data-icon="home">
          Mé tweety
        </a>
      </li>
      <li>
        <a href="#settings" data-theme="" data-icon="gear">
          Nastavení
        </a>
      </li>
    </ul>
  </div>
</div>
<!-- OBSAH STRANKY, V TOMTO PRIPADE VYPIS NOVYCH TWEETU -->
  <div data-role="content">
    <p>
      Zde bude výpis aktuálních tweetů
    </p>
  </div>
<!-- PATICKA STRANKY -->
  <div data-theme="a" data-role="footer" data-position="fixed">
    <h3>
     Paticka
    </h3>
  </div>
</div>
```

Nyní vysvětlení k jednotlivým atributům typickým pro použití v kombinaci s jQuery Mobile:

data-theme="a" – jQuery Mobile má vlastnost jednoduchého barevného přizpůsobení ovládacích prvků přiřazením atributu `data-theme`, základní možnosti jsou `a`, `b`, `c`, `d`, `e`. Zkuste

si několik hodnot pro otestování pozměnit. Další témata můžete sami libovolně vytvářet pomocí on-line nástroje s názvem Theme Roller, který je k nalezení na adrese *http://jQuerymobile.com/themeroller/index.php*.

data-iconpos="bottom" – pokud naše horizontální menu bude obsahovat ikony jako v našem případě, je možno definovat jejich výchozí umístění, a to pomocí atributu `data-iconpos`, který má možnosti `left`, `to`, `right` a `bottom`. V našem případě je použita hodnota `bottom`, takže ikona bude umístěna v dolním okraji.

data-icon="refresh" – tento atribut označuje ikonu, která je zobrazena v navigačním tlačítku, jQuery Mobile má sadu přednastavených ikon, které je možno použít přímo.

Lze samozřejmě použít i vlastní ikony, a to buď úpravou souboru s názvem *icons-36-black.png* a *icons-36-white.png*, či úpravou stylopisu. Tento postup je velice dobře popsán na této adrese: *http://jQuerymobile.com/demos/1.1.0/docs/buttons/buttons-icons.html*.

data-position="fixed" – atribut lze použít pouze na `div header` a `footer`, a to v případě, pokud chceme, aby například patička stránky byla fixována k dolnímu okraji displeje, a nikoliv stránky, takže pokud máme na stránce výpis položek v ListView a chceme zobrazovat menu stále při spodním okraji displeje, použijeme zmiňovaný atribut `data-position="fixed"` aplikovaný na `div footer`.

Stránka tedy bude obsahovat `header` včetně titulku stránky, navigační menu, samotný výpis tweetů a patičku.

Stránka je v tuto chvíli bez jakékoliv další funkcionality, ty budeme přidávat v dalších kapitolách.

Pokud máte raději vizuální nástroje pro tvorbu uživatelského UI, je možno tento zdrojový kód ovládacích prvků i jednoduše naklikat pomocí on-line nástroje **Codiqa**, který naleznete na adrese *http://www.codiqa.com/builder/demo*, a výsledný kód si nechat vygenerovat jako HTML dokument ke stažení.

Inicializace aplikace a ověření funkčnosti

Abychom mohli aplikaci reálně používat, je třeba, aby veškerá funkcionalita, kterou naše aplikace bude používat, byla dostupná. To je podmíněno tím, že budou správně načteny veškeré knihovny, které naše aplikace bude používat, a to především tyto knihovny:

- PhoneGap
- jQuery Mobile (včetně CSS stylopisů)
- jQuery
- `Main.js` (vlastní kód aplikace)

První tři knihovny byly zmíněny již dříve, včetně jejich nalinkování do projektu, resp. bylo popsáno přidání mezi tagy `<head></head>` souboru `index.html`. To, co nám ale chybí, je nalinkování souboru s názvem `main.js`.

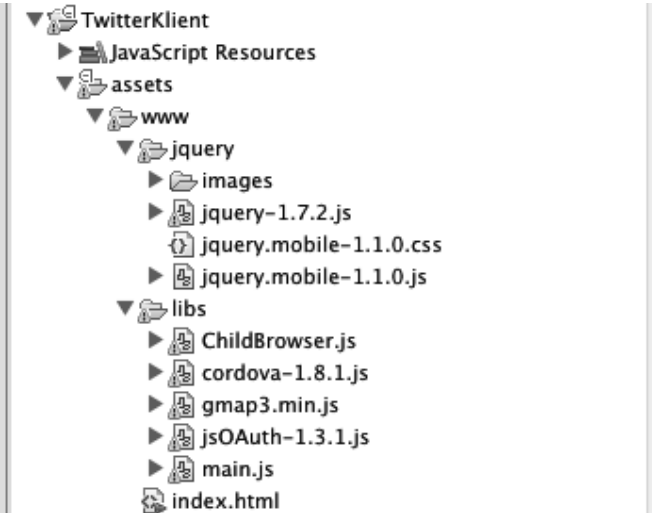
V adresáři `/libs` si tedy založíme soubor s názvem `main.js`, který bude v budoucnu obsahovat celý kód a jádro naší aplikace a tento soubor nalinkujeme opět mezi tagy `<head></head>` do souboru `index.html` přidáním kódu níže.

```
<!-- VLASTNI KOD APLIKACE ZE SLOZKY /LIBS -->
<script type="text/javascript" charset="utf-8" src="libs/main.js"></script>
```

Dále přesuneme soubor `cordova-1.8.1.js` také do adresáře `/libs`, abychom měli všechny související soubory pohromadě.

```
<!-- PHONEGAP KNIHOVNA PRESUNUT Z HLAVNI SLOZKY DO SLOZKY /LIBS -->
<script type="text/javascript" charset="utf-8"
  src="libs/cordova-1.8.1.js">
</script>
```

Adresářová struktura našeho projektu tedy v současné chvíli vypadá jako na níže uvedeném obrázku.



Obrázek 24: Detail souborů projektu

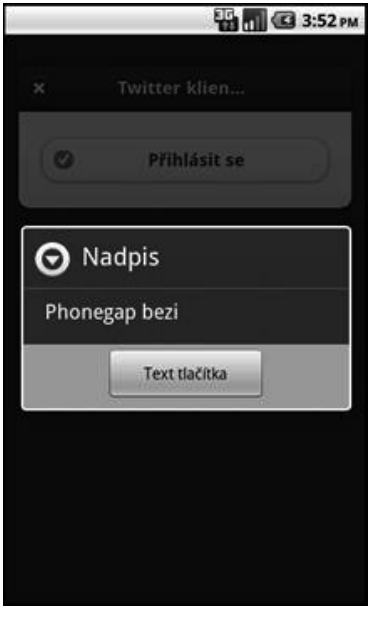
V tomto souboru budeme nadále provádět změny. První změna je ta, že na událost $(document). ready spustíme inicializační funkci `init();`, která nám ověří, zda jsou veškeré funkce, se kterými bude telefon pracovat, k dispozici a také zda je zařízení připojené k Internetu.

Do souboru `main.js` tedy vložíme následující kus zdrojového kódu:

```
//V případě, že je celý dokument včetně knihoven načten, zavoláme funkci //init();
$(document).ready(init());
//V této funkci inicializujeme listener (posluchače)
//pro událost deviceready
//a v jejím CallBacku voláme funkci s názvem onDeviceReady.
function init(){
```

```
  document.addEventListener("deviceready", onDeviceReady, false);
}
//Pokud je PhoneGap načten, spustíme fci onDeviceReady a vypíšeme
//do konzole i na displej uživatele hlášku "PhoneGap bezi"
function onDeviceReady(){
  console.log("PhoneGap bezi");
  showNativeMessage("PhoneGap bezi",false,"Nadpis","Text tlačítka");
//checkSettings();
}
//Funkce pro zobrazení nativního MessageBoxu, jako parametr je použito
//Zpráva, callback, titulek, název tlačítka
function showNativeMessage(message,callback,title,buttonText){
  navigator.notification.alert(
  message, // Zprava
  callback,  // callback
  title,  // titulek
  buttonText // nazev tlacitka
  );
}
```

Pokud nyní aplikaci spustíme v našem Android emulátoru nebo zařízení, měli bychom spatřit výsledek (jako na obrázku níže).



Obrázek 25: Nativní Message Box

Nyní přidáme naší aplikaci jednu další funkcionalitu, a to již zmiňované ověření, zda je telefon připojen k Internetu, a je tak schopen načítat a odesílat nové tweety.

Do našeho souboru `main.js` tedy přidáme následující kus kódu. Do funkce `init()` doplníme následující řádek:

```
function init(){
  document.addEventListener("deviceready", onDeviceReady, false);
//Přidáváme EventListener, který v případě, že je app offline, spustí
v //callbacku fci onOffline
  document.addEventListener("offline", onOffline, false);
}
```

Nakonec souboru přidáme novou funkci, která bude zavolána v případě, že aplikace nebude mít žádné dostupné připojení k Internetu.

```
//Tato funkce je spuštěna EventListenerem offline v případě, ze zařízení
//nemá žádné dostupné připojení k Internetu
function onOffline(){
  showNativeMessage("Není dostupné žádné připojení k internetu",init,
  "Chyba připojení","Zkusit znovu");
}
```

Jedná se v podstatě opět o informativní zprávu pro uživatele o nedostupnosti internetového připojení, s tím rozdílem, že je zde volán callback, a to funkce `init()`, která provede opět celou kontrolu znovu až do chvíle, než se aplikace dostane zpět do on-line režimu.

Pokud na svém PC zkusíte odpojit aktuální připojení, ať již přes WiFi či ethernet, a aplikaci spustíte, měli byste vidět stejný výsledek jako na obrázku níže.

Obrázek 26: Kontrola aktivního internetového připojení

Toto je pouze náhled elektronické knihy. Zakoupení její plné verze je možné v elektronickém obchodě společnosti eReading.