

СРС 1_ЧКВ

Тема: Разработка мобильных приложений на PhoneGap и jQuery Mobile

- Разработка веб-сайтов

PhoneGap — это OpenSource платформа, позволяющая разрабатывать мобильные приложения на HTML, JavaScript и CSS под различные платформы (практически без изменения кода приложения) в их число входят: iOS, Android, Blackberry, WebOS, Symbian и Windows Mobile на подходе. Прелесть его в том, что он не требует навыков разработки под конкретную платформу. Вы пишете свое приложение на JavaScript, используете HTML и CSS для разметки. Вы пишете мобильное приложение как обычный сайт или веб-сервис.

Движок PhoneGap расширяет API браузера и добавляет следующие возможности: доступ к акселометру, доступ к камере (пока только фото), доступ к компасу, доступ к списку контактов, запись и прослушивание аудио файлов, предоставляет доступ к файловой системе, позволяет работать с разными HTML5 хранилищами localStorage, Web SQL и т.п а также позволяет безболезненно обращаться к любому кросс-доменному адресу.

Кроме платформы PhoneGap имеет, пока бесплатный, билдер приложений под все устройства в один клик.

jQuery Mobile

Думаю все знают, но напомним ещё раз. jQuery Mobile — это надстройка над jQuery, позволяющая безо всяких проблем разрабатывать мобильные веб сайты и мобильные веб приложения.

Если вы знаете HTML, JavaScript, CSS, jQuery и испытываете проблемы с Java, Objective-C и другими, но желаете попробовать свои силы в мобильной разработке, то эта статья для вас.

Мы будем создавать мобильное приложение под Android (iOS требует значительно больших усилий для тестового приложения: Mac Only, нужен сертификат разработчика) под названием Tumblr Reader (не хотел создавать ещё один хабраридер).

Установка пакета программ для разработки

Если у вас стоит Eclipse + ADT Plugin, то эту часть можно пропустить. [Оригинальная статья](#). [Подробная статья](#).

1. Скачайте и установите [Eclipse Classic](#)
2. Скачайте и установите [Java Platform \(JDK\)](#)
3. Скачайте и установите [Android SDK](#)

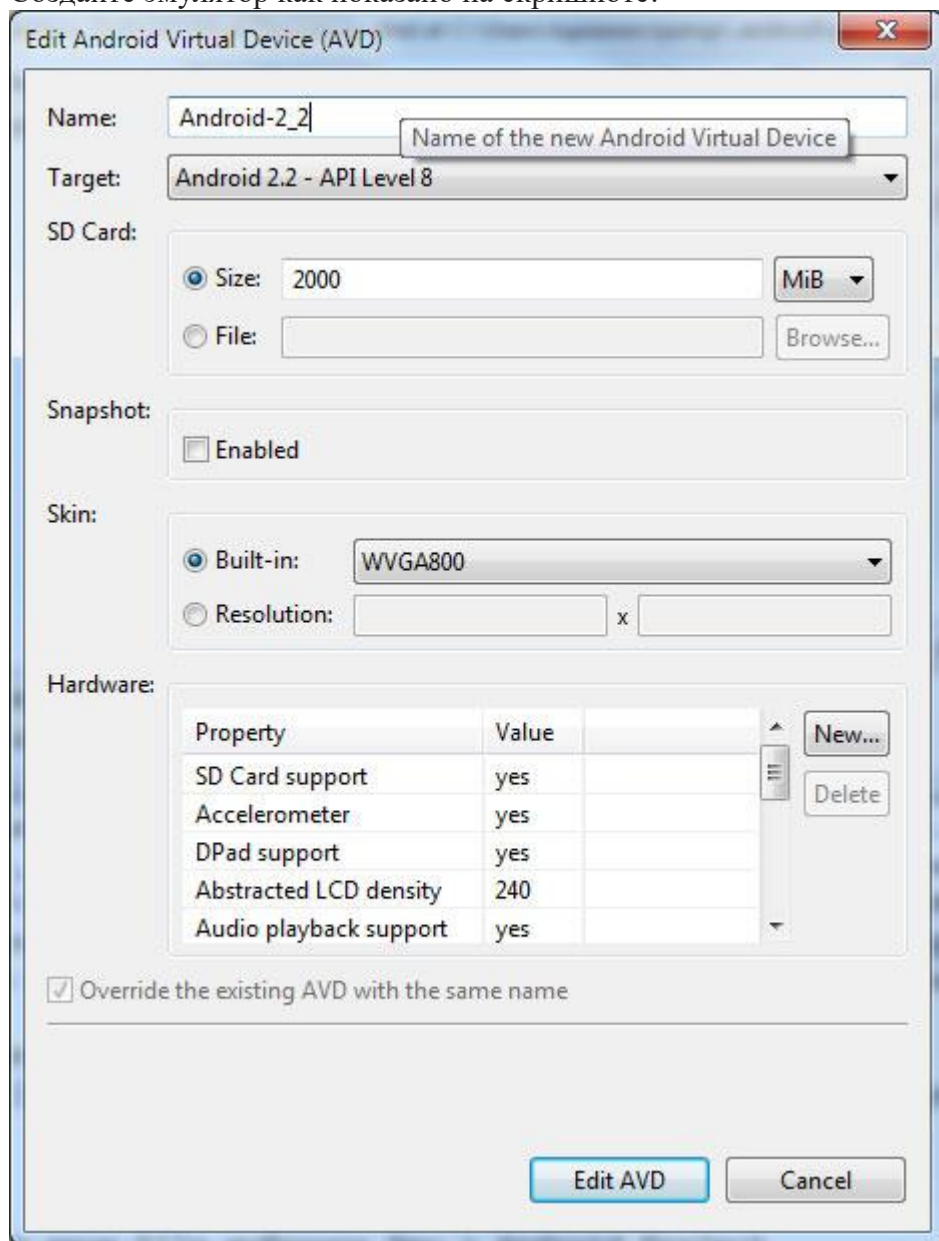
Запустите SDK Manager, выберите все модули и установите их все (займет минут 20)

4. Установите [ADT Plugin](#)
5. [Сконфигурируйте ADT Plugin](#) — вам необходимо указать путь до Android SDK иначе ничего не получится
6. Скачайте последнюю версию [PhoneGap](#) (на момент написания она была 0.9.4)

Создание эмулятора Android

Откройте SDK Manager (он может открыть различные окна, закройте их) выберите Virtual devices, нажмите New...

Создайте эмулятор как показано на скриншоте:

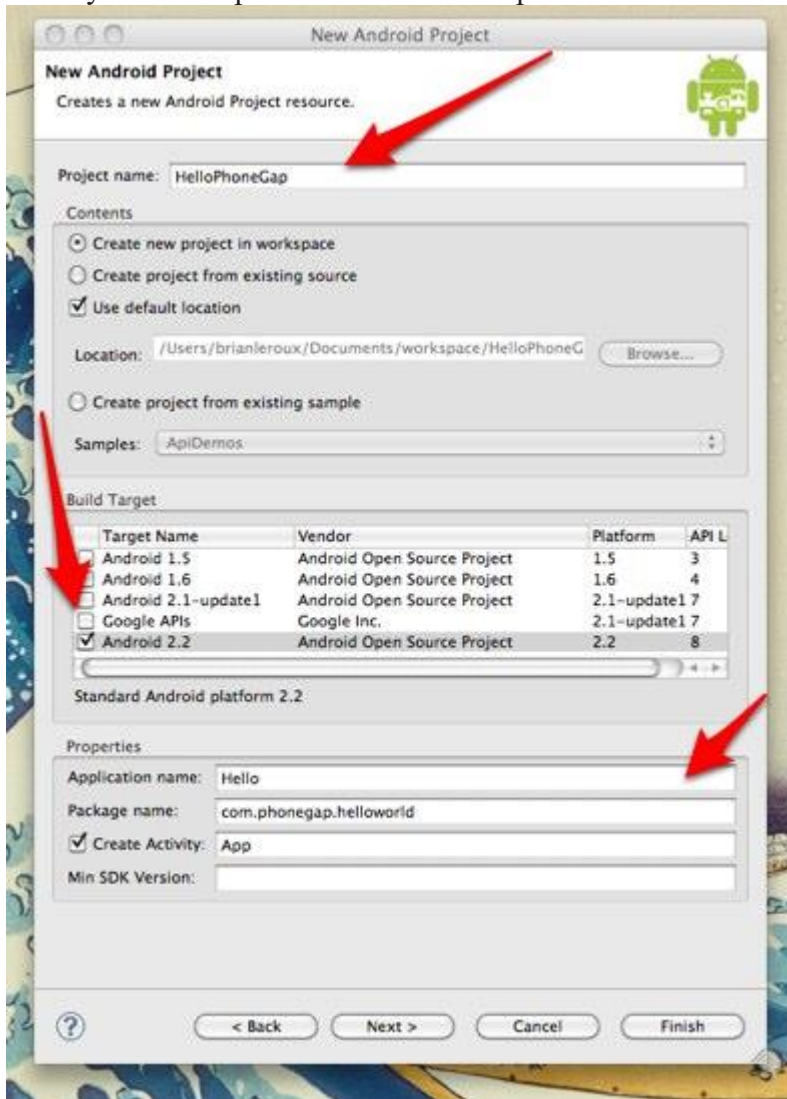


Важно: Необходим именно Android 2.2. Если вы под Windows и в имени пользователя есть не ANSI символы, то вам необходимо после создания виртуалки сделать следующее:

переместите `c:\Users\ВашИмя\.android\avd\Android-2_2.avd` в какую-нибудь другую папку, например в `c:\Android\Android-2_2.avd`, откройте файл `Android-2_2.ini` и измените `path=` на соответствующий.

Создание проекта

1. Запустите Eclipse в меню File выберите New > Android Project



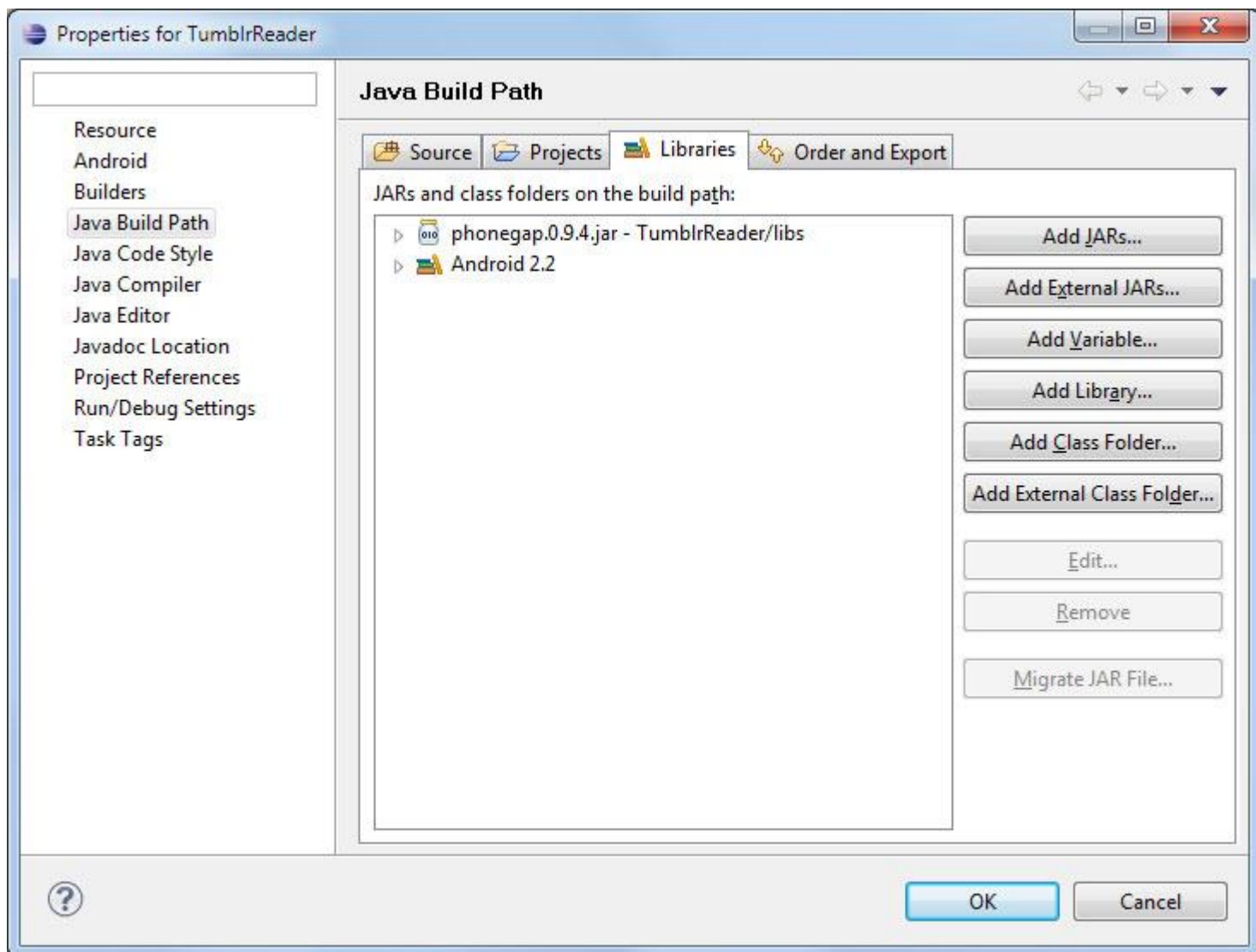
Скриншот с официального tutorials

2. В корневой директории проекта создайте 2 папки /libs и /assets/www

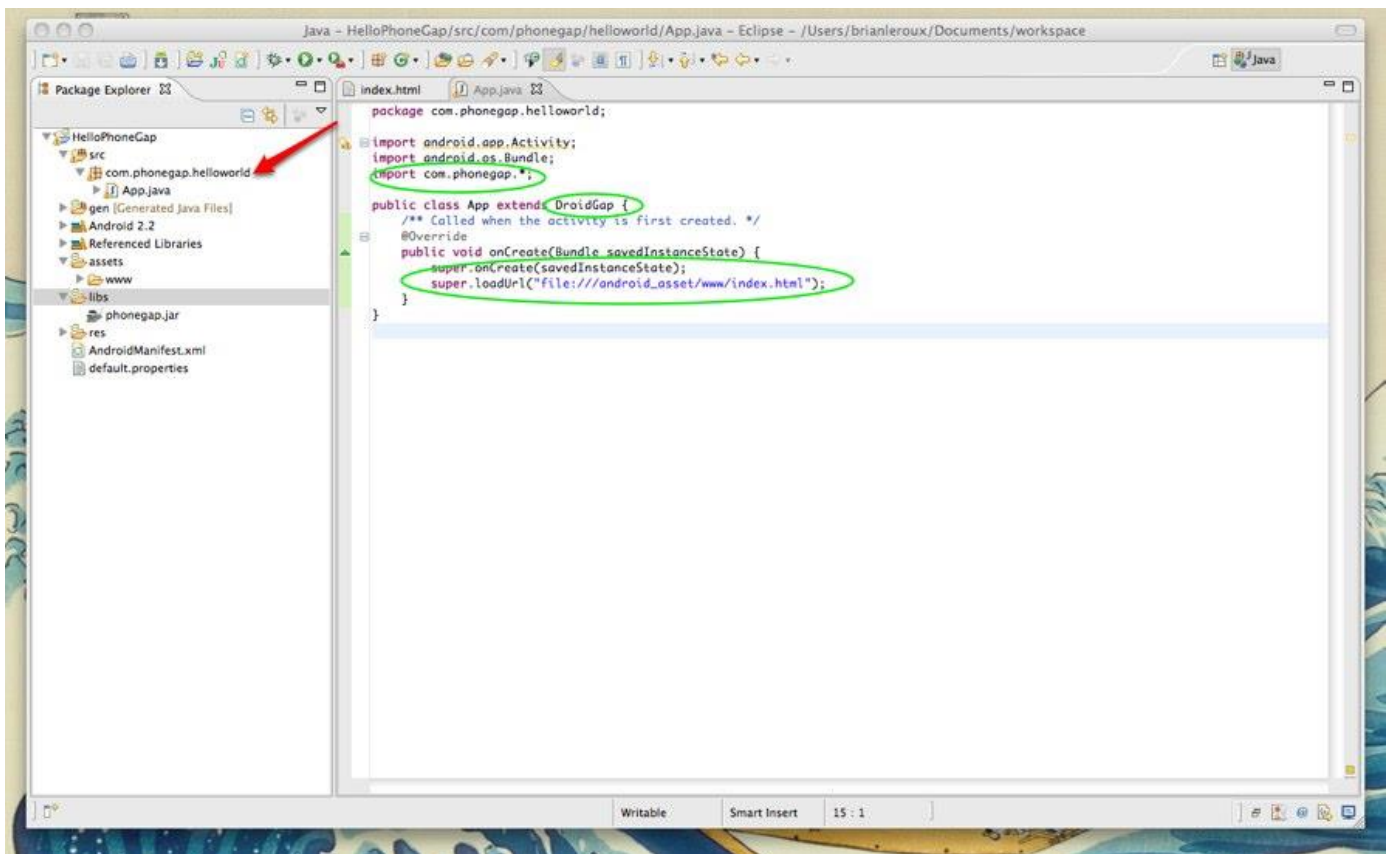
Из архива phonegap-0.9.4 скопируйте phonegap.jar в /libs

Выполните Git чекаут в /assets/www из репозитория: [git://github.com/azproduction/phonegap-tumblr-reader.git](https://github.com/azproduction/phonegap-tumblr-reader.git) или скачайте архивом из github.com/azproduction/phonegap-tumblr-reader/archives/master и распакуйте в /assets/www В репозитории готовое приложение, чтобы вам не пришлось копирастить из статьи.

3. Добавьте phonegap.jar в Build Path для этого нажмите правой кнопкой на корень проекта, выберите Properties, в списке выберите Java Build Path, выберите вкладку Libraries, нажмите Add JARs... и укажите путь до phonegap.jar



4. Откройте файл TumblrReader.java, который лежит в папке src/
 - 4.1. Замените Activity на DroidGap
 - 4.2. Замените setContentView() на super.loadUrl("file:///android_asset/www/index.html");
 - 4.3. Добавьте import com.phonegap.*;



6. Откройте `AndroidManifest.xml`, допишите следующий код (он дает доступ к определенным ресурсам) нам нужен только `android.permission.INTERNET`:

`<supports-screens`

`android:largeScreens="true"`

`android:normalScreens="true"`

`android:smallScreens="true"`

`android:resizeable="true"`

`android:anyDensity="true"`

`>`

`<uses-permission android:name="android.permission.INTERNET" />`

Если вы будете экспериментировать с АПИ, то добавьте все опции:

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<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_C
```

```
OMMANDS" />
```

```
<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" />
```

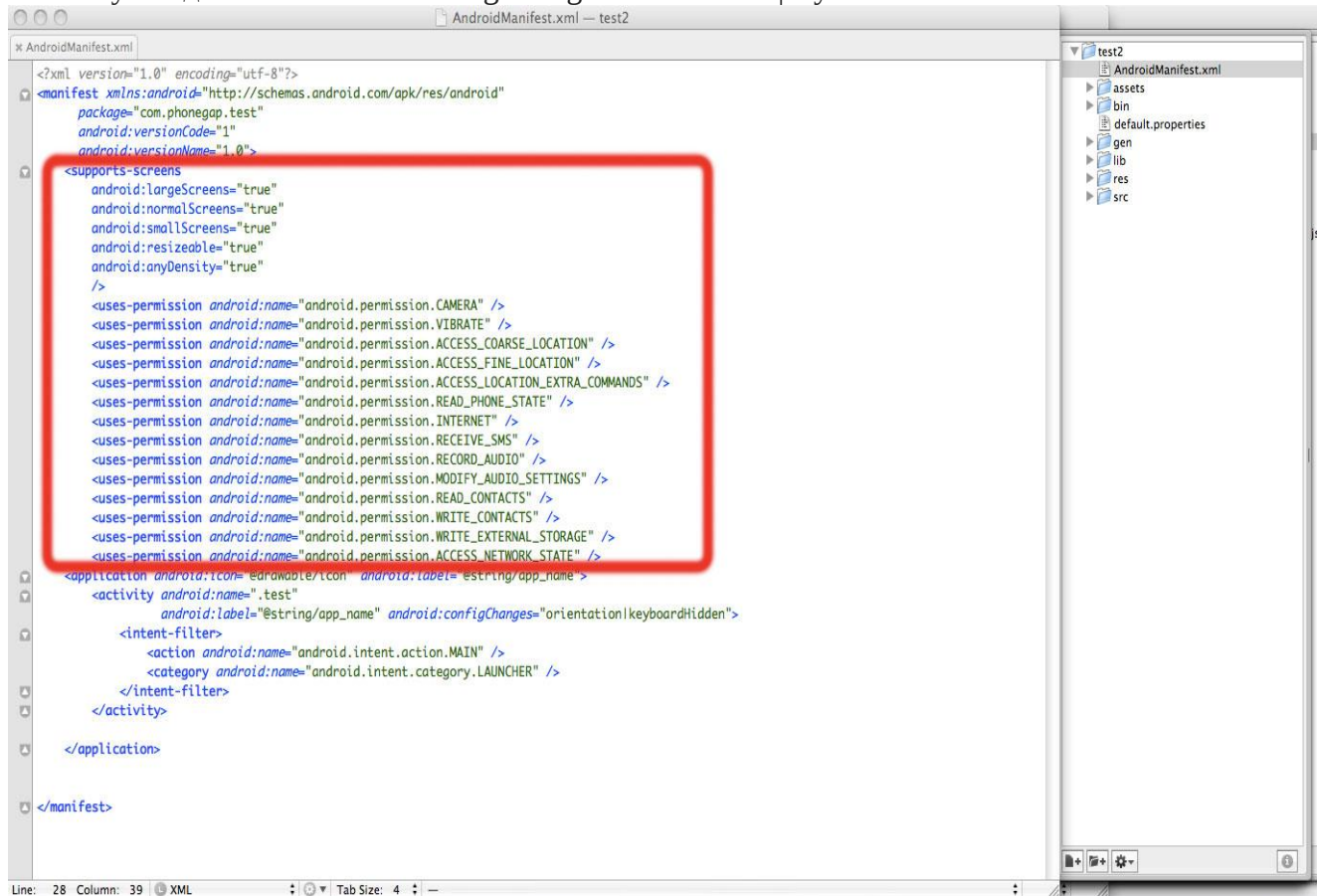
```
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
```

```
/>
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

в activity тег добавьте android:configChanges="orientation|keyboardHidden"



7. Запустите ваше приложение в эмуляторе: Run As > Android Application если вы сделали все правильно, то должен открыться эмулятор с вашим приложением.

Создание Tumblr Reader

Далее в статье описывается поэтапное создание приложения. Теперь Eclipse можно свернуть и открыть ваш любимый редактор. Eclipse понадобится для перезапуска приложения в эмуляторе.

Скелет приложения index.html

Я не останавливаюсь на принципах создания jQuery Mobile страниц. Вы можете прочитать сами. Создаем index.html в папке /assets/www

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Подключаем необходимые скрипты

```
<head>
```

```
<title>Tumblr Reader</title>
```

```
<!-- Стили jQuery Mobile и наши -->
```

```
<link rel="stylesheet" href="libs/jquery.mobile.css" />
```

```
<link rel="stylesheet" href="css/main.css" />
```

```
<!-- jQuery & jQuery Mobile -->
```

```
<script type="text/javascript" charset="utf-8" src="libs/jquery.js"></script>
```

```
<script type="text/javascript" charset="utf-8" src="libs/jquery.mobile.js"></script>
```

```
<!-- Скрипт phonegap -->
```

```
<script type="text/javascript" charset="utf-8" src="libs/phonegap.js"></script>
```

```
<!-- Это наш основной скрипт -->
```

```
<script type="text/javascript" charset="utf-8" src="js/main.js"></script>
```

```
</head>
```

В body добавляем jQuery Mobile страницу — «Список мелких картинок» aka Dashboard

```
<!-- Список мелких картинок -->
```

```
<div data-role="page" id="dashboard">
```

```
<div data-role="header" data-position="fixed">
```

```
<!-- По клику откроет страницу options -->
```

```
<a href="#options" data-icon="gear">Options</a>
```

```
<h1>Dashboard</h1>
```

```
</div>
```

```
<div data-role="content">
```

```
<div class="ui-grid-b" id="dashboard-list">
```

```
<!-- Сюда будут отрисовываться мелкие картинки -->
```

```
</div>
```

```
</div>
```

```
</div>
```

При клике на мелкую картинку будет открываться страница с большой, добавляем страницу «Большая картинка»

```
<!-- Большая картинка -->
```

```
<div data-role="page" id="dialog">
```

```
<div data-role="header">
```

```
<!-- Кнопка Назад -->
```

```
<a data-rel="back" data-icon="back">Back</a>
```

```
<h1>Image</h1>
```

```
<!-- Ссылка на полную версию - по клику откроется в браузере -->
```

```
<!-- about:blank будет заменяться на соответствующую ссылку -->
```

```
<a href="about:blank" data-icon="search" data-theme="b" id="full-image">Full</a>
```

```
</div>
```

```
<div data-role="content">
```

```
<!-- about:blank будет заменяться на путь к большой картинке -->
```

```
<p></p>
```

```
</div>
```

```
</div>
```

По умолчанию картинки будут загружаться с design.tumblr.com (первое, что пришло в голову),

добавим страницу настроек, чтобы имя можно было менять:

```
<!-- Настройки -->
```

```
<div data-role="page" id="options">
```

```
<div data-role="header">
```

```
<h1>Options</h1>
```

```
</div>
```

```
<div data-role="content">
```

```
<div data-role="fieldcontain">
```

```
<label for="name">Tumblr user:</label>
```

```
<!-- Имя Tumblr пользователя -->
```

```
<input type="text" name="name" id="options-name" value="" />
```

```
</div>
```

```
<!-- По клику сохранится имя и откроется "Список мелких картинок" -->
```

```
<a data-role="button">Ok</a>
```

```
</div>
```

```
</div>
```

Полная версия github.com/azproduction/phonegap-tumblr-reader/blob/master/index.html

Скелет готов, добавим немного css (файл css/main.css)

```
#dashboard-list div,
```

```
#dialog p {
```

```
text-align: center;
```

```
}
```

Логика приложения js/main.js

Сперва изучим [Tumblr API](#) в разделе JSON output немного описан формат JSON: по ссылке

```
http://%user%.tumblr.com/api/read/json
```

нам придет JavaScript вот такого формата:

```
var tumblr_api_read = { /* тут какие-то данные ... */ }
```

Жаль, что не чистый JSON(P), но у нас есть jQuery.getScript, которая все разрулит за нас. Больше ничего о формате неизвестно. По исходнику стало понятно, что tumblr_api_read.tumblelog.title это заголовок блога, а tumblr_api_read.posts это массив постов формата:

```
[{
```

```
'type': 'photo',
```

```
'photo-url-1280': 'http://...',
```

```
'photo-url-250': 'http://...',
```

```
'photo-url-75': 'http://...',
```

```
/* и ещё что-то не нужное для нас */
```

```
}, ...]
```

Мы будем использовать только эти поля, поэтому остальные не описываю.

Рендер для отрисовки списка картинок

Каждая картинка будет вида:

```
<div class="ui-block-a">
```

```
<a href="#dialog">
```

```

```

```
</a>
```

```
</div>
```

Код рендера:

```
/**
```

```
 * Отрисовывает тumbnailсы
```

```
 *
```

```
 * @param {Object} posts
```

```
 *
```

```
 * @returns {String}
```

```
 */
```

```
var render = function (posts) {
```

```
var view = "",
```

```
// Используем правильный индекс т.к.
```

```
// Берем не все посты
```

```
index = 0;
```

```
// Собираем
```

```
posts.forEach(function(post){
```

```
// Нужны только фотографии
```

```
if (post.type !== 'photo') {
```

```
return;
```

```
}
```

```
view +=
```

```
'<div class="ui-block-' + suffix[index % 3] + ">' +
```

```
// При клике автоматом откроется страница dialog
```

```
'<a href="#dialog">' +
```

```
'<img data-full-src="" + post['photo-url-1280'] + "" data-big-src="" + post['photo-url-2
```

```
50'] + "" src="" + post['photo-url-75'] + "" />' +
```

```
'</a>' +
```

```
'</div>';
```

```
index++;
```

```
});
```

```
return view;
```

```
};
```

Загрузчик данных с Tumblr API

Загрузчик дергает URL

```
http://%user%.tumblr.com/api/read/json
```

и получает соответствующие данные, используя данные мы отрисовываем список картинок.

```
/**
```

```
 * Загружает и отрисовывает контент
```

```
 *
```

```
 * @param {String} [user='tumblr']
```

```
 */
```

```
var load = function (user) {
```

```
  user = user || 'tumblr';
```

```
  // Включаем крутилку
```

```
  $.mobile.pageLoading();
```

```
  // Добавляем на всякий случай
```

```
  var timeoutId = window.setTimeout(function () {
```

```
    $.mobile.pageLoading(true);
```

```
  }, 3000);
```

```
  // Загружаем
```

```
  $.getScript('http://' + user + '.tumblr.com/api/read/json', function (data, status) {
```

```
    // Если все хорошо
```

```
    if (status === 'success'){
```

```
      // Меняем заголовок
```

```
      $dashboard.find('h1').text(tumblr_api_read.tumblelog.title);
```

```
      // Рендерим контент
```

```
$('#dashboard-list').html(render(tumblr_api_read.posts));
```

```
} else {
```

```
// Что-то не так
```

```
alert('Error occurred while loading ' + user);
```

```
}
```

```
// Останавливаем крутилку
```

```
$.mobile.pageLoading(true);
```

```
// Убиваем таймер
```

```
window.clearTimeout(timeoutId);
```

```
});
```

```
};
```

Инициализатор приложения

Он инициализирует имя пользователя из localStorage, выполняет предварительную загрузку и делает другие операции по мелочи.

```
var init = function () {
```

```
// Если 1 раз инициализировался, не вызываем повторно
```

```
if (init.called) {
```

```
return;
```

```
}
```

```
init.called = true;
```

```
// Инициализация блоков
```

```
$dashboard = $('#dashboard');
```

```
$dashboardList = $('#dashboard-list');
```

```
$dialogImg = $('#dialog').find('img');
```

```
$fullImg = $('#full-image');
```

```
$options = $('#options');
```

```
$optionsName = $('#options-name');
```

```
// Клик на любой тумбнайл
```

```
$dashboardList.delegate('a', 'click', function () {
```

```
    var $img = $(this).find('img');
```

```
// Меняем опции диалога
```

```
$dialogImg.attr('src', $img.data('big-src'));
```

```
$fullImg.attr('href', $img.data('full-src'));
```

```
});
```

```
// Загружаем или берем значение по умолчанию
```

```
currentUser = window.localStorage.getItem("tumblr-reader-user") || 'design';
```

```
// Закидываем в опции
```

```
$optionsName.val(currentUser);
```

```
// Нажатие кнопки ОК в опциях
```

```
$options.find('a').click(function () {
```

```
    var newUser = $optionsName.val();
```

```
// Если пользователь поменялся
```

```
    if (currentUser !== newUser) {
```

```
// Меняем текущего пользователя
```

```
        currentUser = newUser;
```

```
// Сохраняем опции
```

```
        window.localStorage.setItem("tumblr-reader-user", currentUser);
```

```
// Загружаем данные с тумблера
```

```
load(currentUser);
```

```
}
```

```
// Переходим на главную
```

```
$.mobile.changePage('#dashboard');
```

```
});
```

```
// Первичная загрузка
```

```
load(currentUser);
```

```
};
```

```
init.called = false;
```

```
// Девайс готов, этого события нет в браузере
```

```
document.addEventListener("deviceready", init, true);
```

```
// Оставляем для отладки в обычном браузере
```

```
$(init);
```

Исходник main.js: github.com/azproduction/phonegap-tumblr-reader/blob/master/js/main.js

Теперь наше приложение полностью готово и его можно запустить в Эмуляторе Run As > Android Application, если вы сделали мелкую правку и не хотите пересобирать приложение просто откройте index.html в любом WebKit браузере.

Собираем приложение под все платформы в PhoneGap Build

1. Вам необходимо отправить заявку на странице build.phonegap.com (мне ответ пришел через 5 минут)
2. На ваш почтовый адрес придет письмо с beta code используйте его в build.phonegap.com/people/sign_up
3. Как только вы авторизовались зайдите на build.phonegap.com/apps
4. Выберите опцию upload an archive or index.html file
5. Упакуйте содержимое /assets/www в .zip и отправьте архив, вместо архива можно использовать мой git репозиторий
6. Ждем Create, ждем немного пока PhoneGap Build собирает приложение под все платформы
Теперь ваше приложение доступно для скачивания build.phonegap.com/apps/9296 (моя сборка)

Ссылки

1. Если вам лень собирать и что-либо качать вы можете посмотреть [приложение онлайн](#)
2. Скачать готовое приложение под Android, webOS, Symbian можно [тут](#) (нужна регистрация в PhoneGap Build) iOS требует сертификат, Blackberry требует особых имен файлов поэтому их нет.
3. Скачать без регистрации [Android 1](#) (Сборка PhoneGap Build требует лишних прав), [Android 2](#) (ручная сборка требует только android.permission.INTERNET), [webOS](#), [Symbian](#)
4. Репозиторий Tumblr Reader на [GitHub](#)

Документация

1. [Документация PhoneGap](#)
2. [Вики PhoneGap: Android Eclipse Quickstart](#)
3. [API Tumblr](#)
4. [jQuery Mobile Docs](#)

Я продемонстрировал простое приложение, которое может работать и как веб-страница, вы можете попробовать создать что-то более сложное, использующее API телефона. Предложения, пожелания, критика приветствуется!

CPC_2

<https://www.ibm.com/developerworks/ru/library/wa-mobappdev1/>