Mobile App Documentation

By Flora Liao, University of Toronto

April 2014

Table of Contents

Introduction

The prototype was done in Phonegap, which allows developers to create mobile apps using web technologies such as Javascript, HTML and CSS. The major advantage of Phonegap is that it supports all major platforms like iOS, Android, Windows, and Blackberry. The framework is simple to navigate. Follow the instructions below to start:

› Register Adobe PhoneGap at https://build.phonegap.com/plans/free-adobeid using ROM's GitHub account or create an Adobe ID

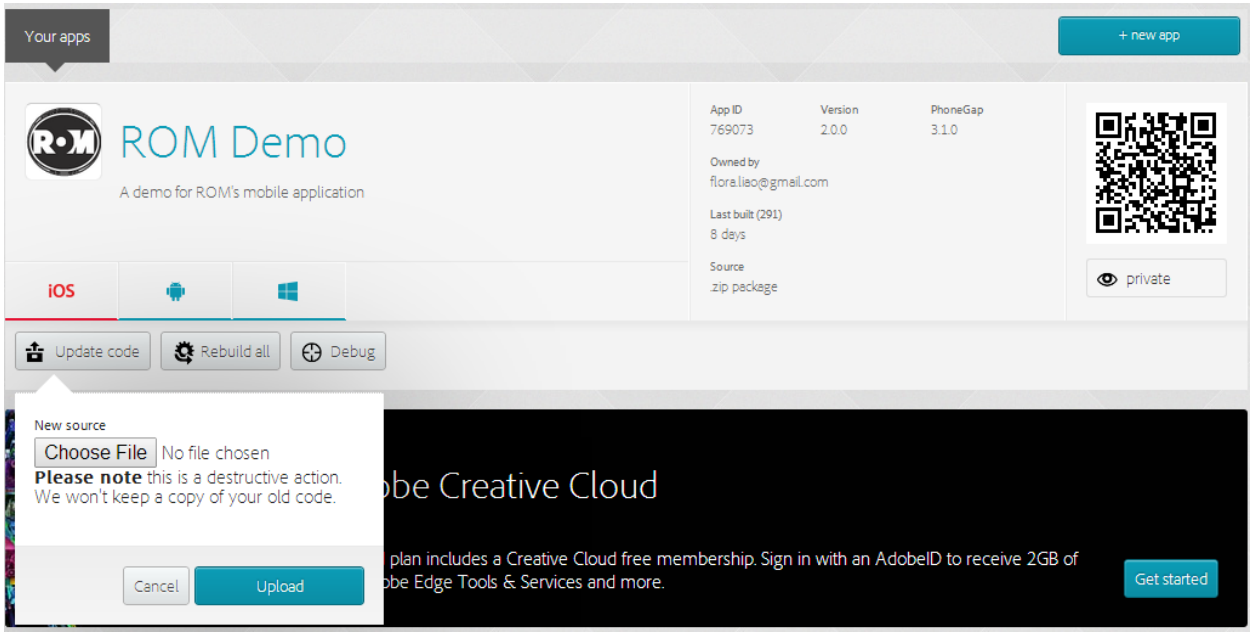
› Sign in and click on the blue button n*ew app* as shown in the figure below

› Navigate to *private* tab, upload the **www.zip** file given in the project ROM/assets

› Make Changes: unzip **www.zip**, modify and save files

› Update code: Save the **www** folder back to ZIP format and upload to the framework

› Download app: Install **QR Code Scanner** on your device and scan the QR code in the framework shown below and install the app



Note: Since PhoneGap Build uses Apple's standard development process to build applications, you will need to sign up for their developer program to build iOS applications on PhoneGap Build. You will also need a Mac to configure your certificate and provisioning profile.

Please see the following link for more details about iOS signing:

http://docs.build.phonegap.com/en_US/3.1.0/signing_signing-ios.md.html

Map & Link to Wikipedia in Modal Window

The icon opens up an image modal window, showing the direction to the object

The icon opens up an iframe modal window, showing the Wikipedia page of the object. Third party library **TinyBox** is used for the modal window.

Please see the following link for more technical details:

http://www.scriptiny.com/2011/03/javascript-modal-windows/

Social Sharing

Social Sharing allows users to share text, image and a link of the object on the ROM website via the native sharing widget on the device. Facebook and Twitter are the focus of the requirement. Note that Facebook on Android does not allow developers to pre-fill text, but it allows sharing image and a link. The official Phonegap plugin **SocialSharing** is used for the requirement.

Please see the following link for more technical details:

https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin/tree/phonegap-2.x/

Bookmark

The feature is not completed because it's not a priority based on the time frame given.

Bookmark allows users to save their favorite objects locally on the app. The code is left there in case ROM wants to further explore the feature.

Swipe to Next Stop, Table of Contents & Skip Stops

PagingSlider is an extension used for the swipe gesture on each stop. Use the HTML templates in tour1.html and tour2.html for new tours.

Please see the following link for more technical details:

https://github.com/linslin/pagingSlider

Data collection

As soon as the user enters the family tour, a timer is used in the back end to calculate the total time the user spends on each stop. When the user swipes to the next stop or skips stops through table of contents, a query is inserted in the database with the following information:

The table has three columns:

› user_id: The unique id of the user at the tour

› stop_id: The stop of the user was at. The digit before the decimal represents the tour number and the digit after the decimal represents the stop number. E.g. 1.3 is the third stop of the first tour

› duration: The time the user spends at the stop in seconds

user_id	stop_id	duration
80012	1.1	9
80012	1.3	8

Figure 1. Tour data

To elaborate more on the technical side, Ajax call sends the information to the server using XMLHttpRequest. The server is hosted on CloudBees, which allows the server to be available for update to the database. CloudBees also provides database with free 5MB and 3,600 maximum queries/ hour. The server side uses JDBC to insert queries in the table. The code is integrated in indexPageSlider.js and pagingSlider.js from **PagingSlider**.

Please see the following link for more details on CloudBees:

http://www.cloudbees.com/

Heidi SQL was used during the demo to view the database.

› Download the installer at http://www.heidisql.com/download.php

› Set up *New Session in Root Folder*

› Fill up the following information for the database I set up:

›› **Hostname/ IP**: us-cdbr-cb-east-01.cleardb.net

›› **User**: bbcf0740c873eb

›› **Password**: d74c51da

› Open and select to *tour* on the left side, and go to *Data* tab on the top to see tour data as shown in Figure 1.

Pinch zoom on maps

The pinch zoom feature is implemented using iscroll5, which is a Javascript scroller that provides various user interactions.

Please see the following link for more technical details:

http://cubiq.org/iscroll-5