Sony Ericsson

Getting Started Guide

WEB SDK

Sony Ericsson

Sony Ericsson Developer World

At www.sonyericsson.com/developer, developers can find the latest technical documentation and development tools such as phone White papers, Developers guidelines for different technologies, Getting started tutorials, SDKs (Software Development Kits) and tool plugins. The Web site also features news articles, moderated discussion forums offering free technical support and a Wiki community sharing expertise and code examples.

For more information about these professional services, go to the Sony Ericsson Developer World Web site.

Trademarks and acknowledgements

Xperia is a trademark or registered trademark of Sony Ericsson Mobile Communications AB.

Google and Android are trademarks or registered trademarks of Google Corporation in the United States and/or other countries.

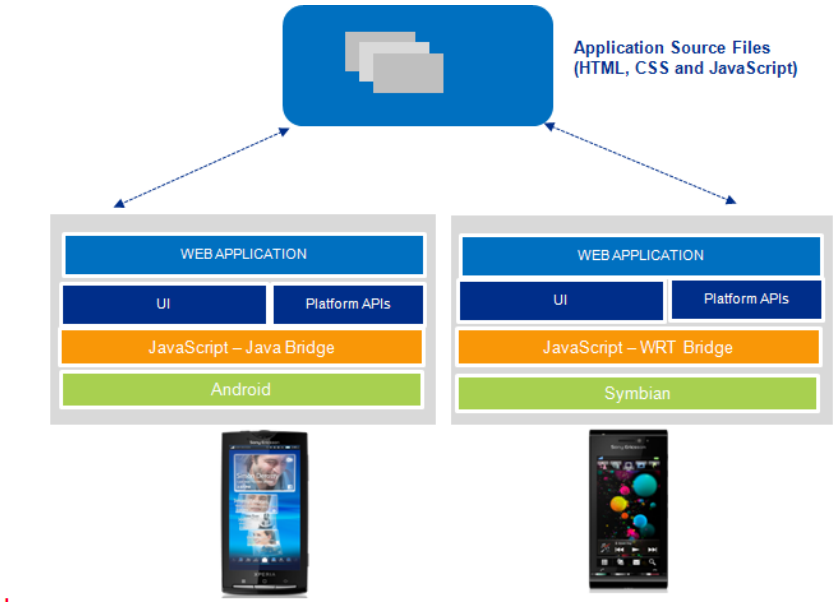
Nokia and Symbian are trademarks or registered trademarks of Nokia Corporation in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

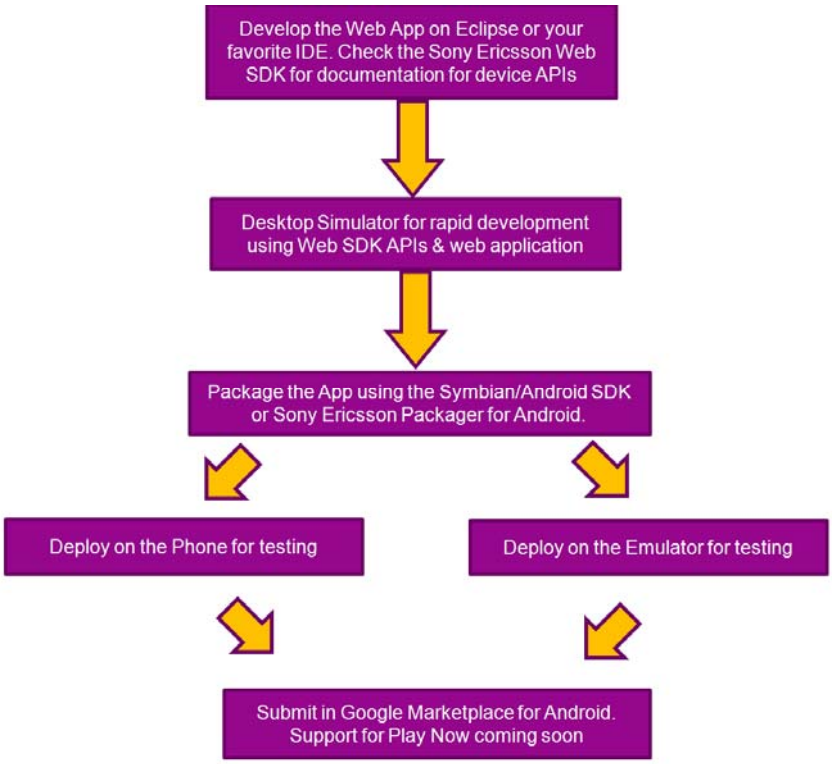
Table of Contents

Introduction

The Web SDK enables developers to produce applications which work consistently independent of the underlying platform. It initially supports Android (e.g. XPERIA™ X10) and Symbian (e.g. Satio™) platforms. The Web SDK architecture is as follows –



The following steps needs to be performed for building a Web Application for Xperia X10 or Satio.

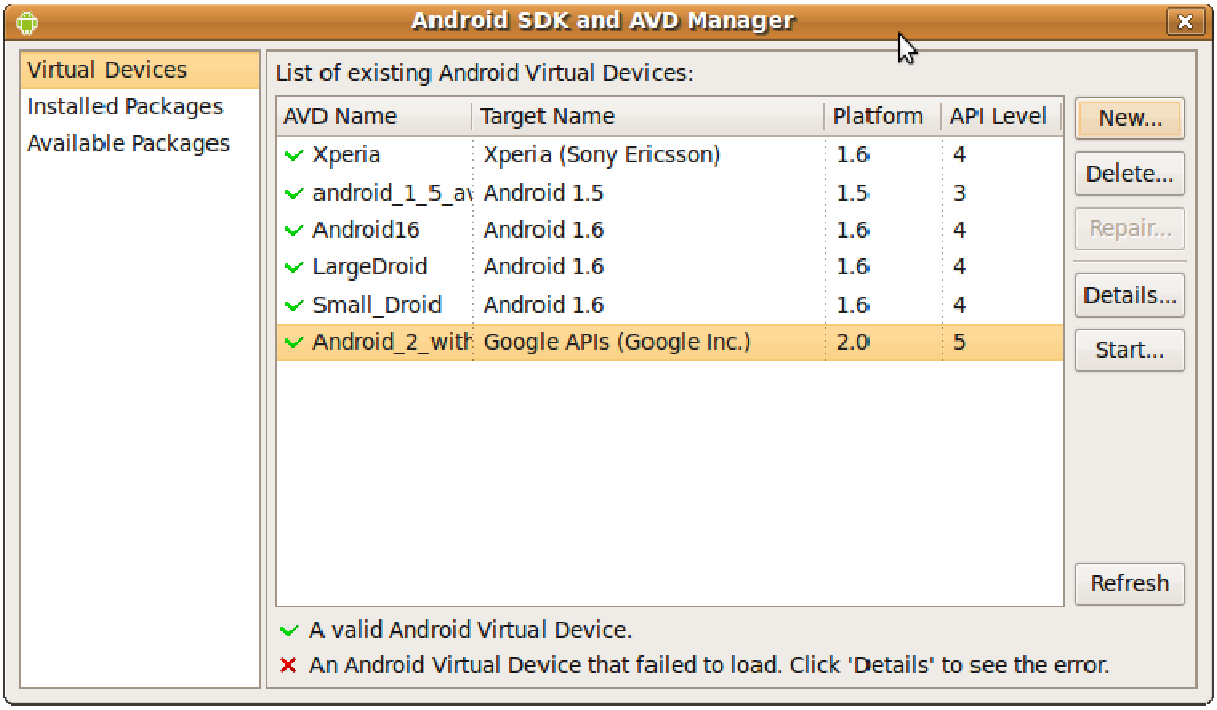


Depending on the knowledge level of the developer, Sony Ericsson Web SDK provides the tools needed for rapid development and deployment of the application. The Web SDK has the following components –

1. Sony Ericsson PhoneGap Simulator – runs the application and simulates phone features for testing
2. Phone Gap Cross Platform Framework – Enables rapid development by utilizing the Phone Gap APIs
3. Packager – packages the application to deploy in sales channels (Play Now Arena support coming soon)
4. Emulator – run the application on a full emulation of the target phone (Xperia X10 and Satio)

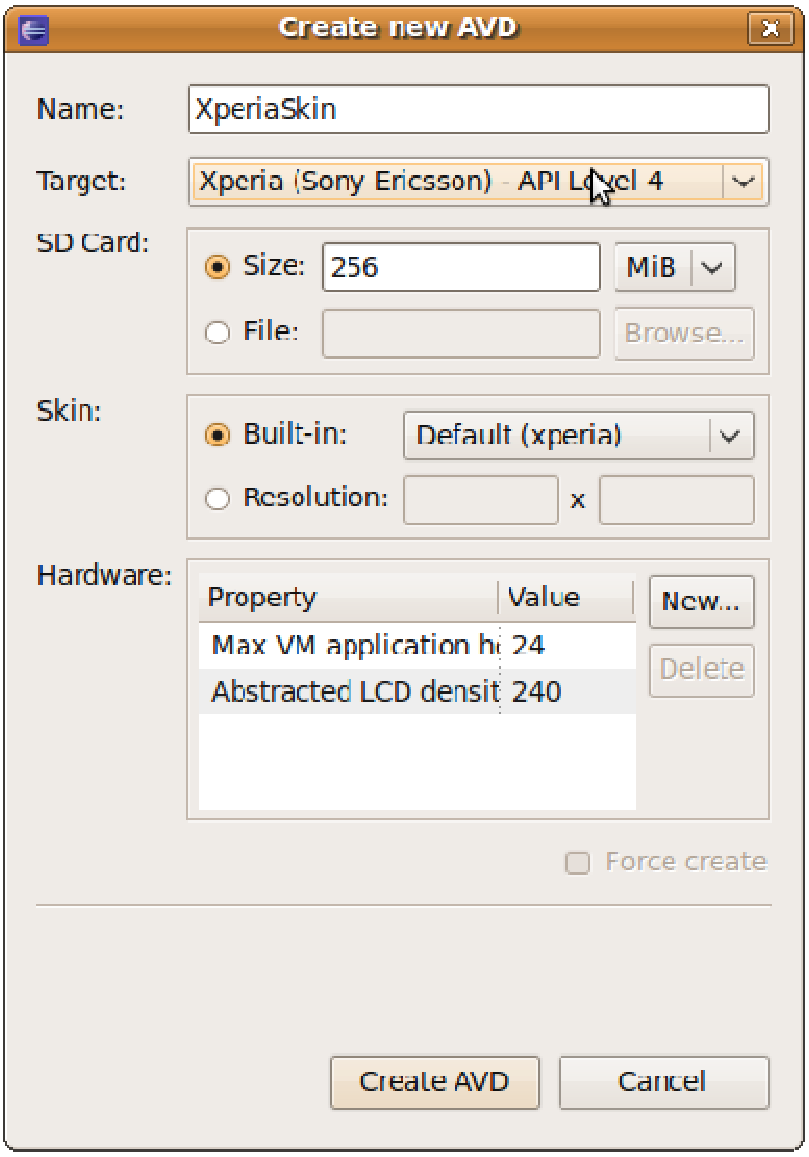
Installing X10 Skin

1. Go to C:/Program Files/Sony Ericsson /Web SDK/Xperia X10 Skin folder

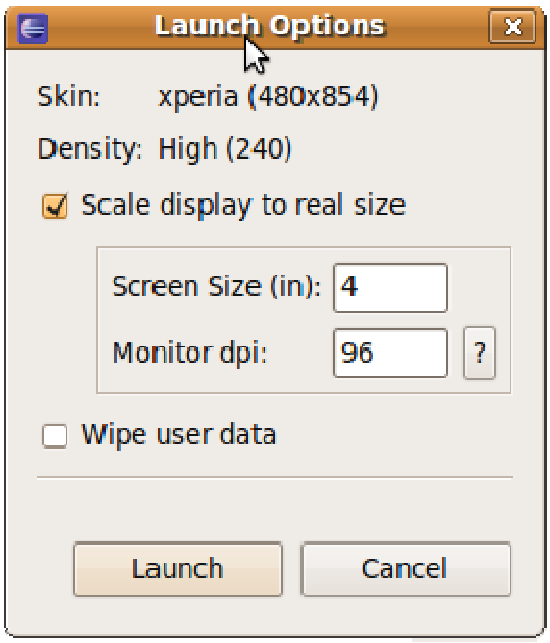


2. Click New, and fill out the fields. Note: The name of an Android Virtual Device can't contain any spaces, dashes or periods. Select Xperia as the target device. Once you are done, click "Create AVD".

3. Start the AVD you have created. This AVD requires a high resolution monitor, so if you do not have one, it is a good idea to scale the size down when starting it.



4. Start the AVD you have created. This AVD requires a high resolution monitor, so if you do not have one, it is a good idea to scale the size down when starting it.



Getting Started with Android Web SDK for X10 in Eclipse

For building the web application with Android SDK, perform the following steps –

1) Install the Java SE Development Kit – JDK from http://java.sun.com/javase/downloads/index.jsp. This is a standard Windows installer files, follow the on screen instructions for installation.

2) Install the Eclipse IDE for Java Developers from http://www.eclipse.org/downloads/ This a ZIP archive file, extract the folder “eclipse” into your “Program Files” folder.

3) Install the Android 1.6 SDK from http://developer.android.com/sdk/index.html. This a ZIP archive file, extract the folder “android-sdk–windows-1.6.zip” into your “Program Files” folder. Install the ADT Plug-in for Eclipse as described here : http://developer.android.com/sdk/android-1.6.html

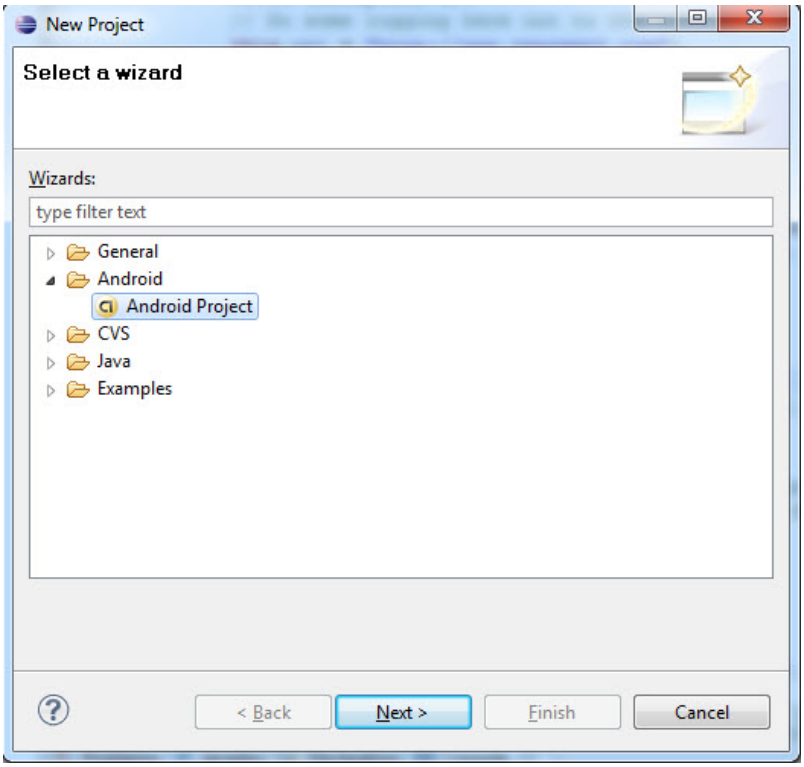
4) To verify your installation of Eclipse, Android SDK, ADT, and Java, you can build a sample “Hello World” application using the instructions located here: http://developer.android.com/guide/tutorials/hello-world.html. At a very minimum, it is good to at least know the basics of Android development including the understanding of Activities, Views, Strings.xml, etc.

   Once you are all set with Eclipse and being able to run Helloworld, you can close that project from Eclipse by right clicking on the project folder and selecting "Close Project", don't worry this will just collapse it, but will be available for future use.

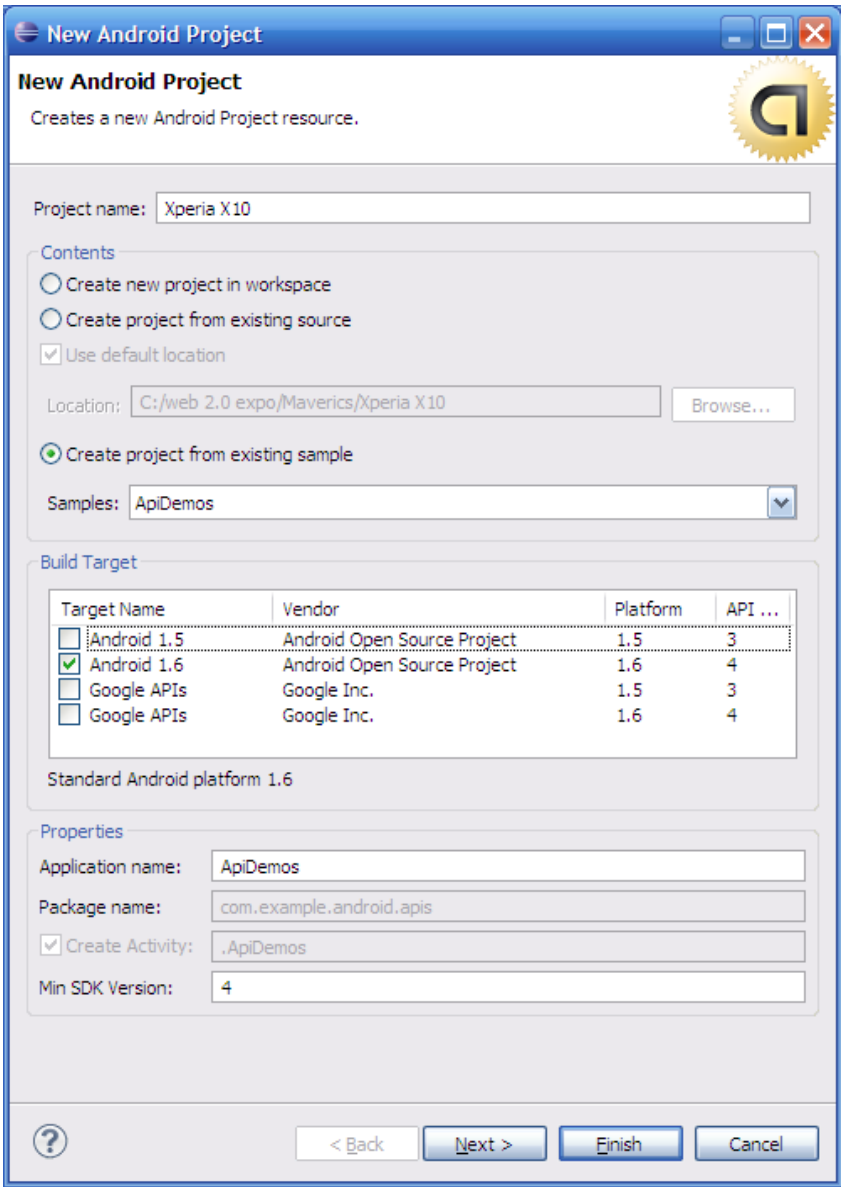
5) Now download the latest Sony Ericsson Web SDK from Sony Ericsson Developer World here

6) Now load the sample project into Eclipse as a project.

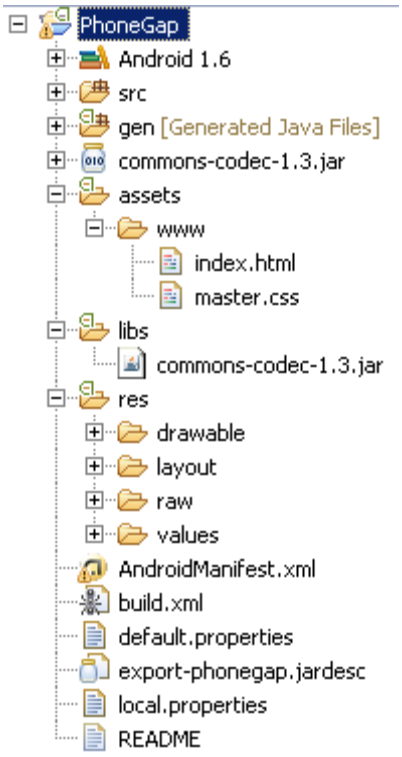
7) From Eclipse, go to File -> New-> Project which opens the New Project Dialog, and click Android Project.



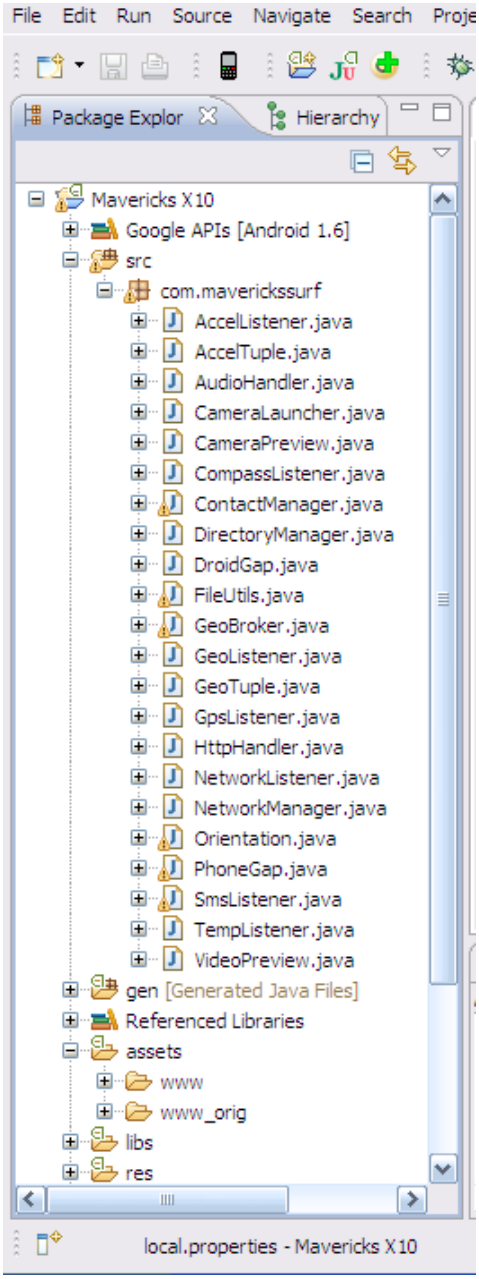
7) Now Provide the Project Name, Create project from existing source. Browse to where you unzipped the PhoneGap source: Navigate to the Android Folder, click OK. Ensure that Android 1.6, is selected then click Finish.



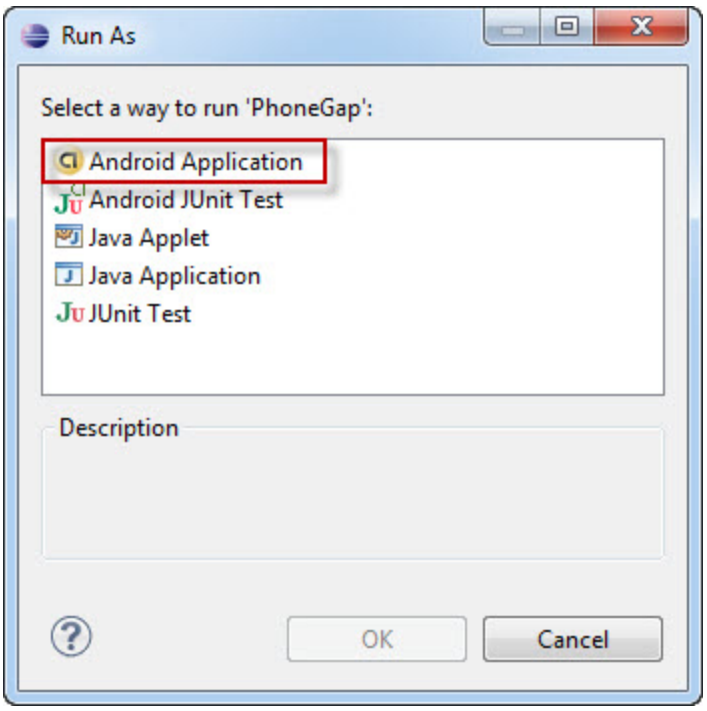
8) Now, in Eclipse the PhoneGap Project will be present.



Here is the directory for the Mavericks application available in the SDK.



10) Click the Run button in Eclipse, it is the button that looks like "Play". When project is run, there is the dialog box to Run As.., choose Android Application. Just select the project folder and click run.



When the project, the emulator launches and sees a title: "Phone Gap"

Getting Started (WRT on Sony Ericsson)

1) Read the S60 5th Edition Release notes and install the prerequisite tools (Active Perl 5.6.1, Java Runtime 1.6).

2) Download and install the S60 5th Edition SDK.

3) Install the Sony Ericsson Satio skin for the S60 5th edition emulator (skin/semc_extension_1.0.0.msi). Note that this emulator only runs on Windows PC.

   Note: PhoneGap for Symbian has no native codebase, and requires no building. This may seem unusual to a developer who has used PhoneGap for one of the other platforms (Android), for which native code must be built around the web files (i.e. html/css/js files built into a package with Java or C). This is because PhoneGap for Symbian uses the WebRuntime

technology to package the web files, and thus all one needs is a web-app and the Symbian version of Phonegap.js.

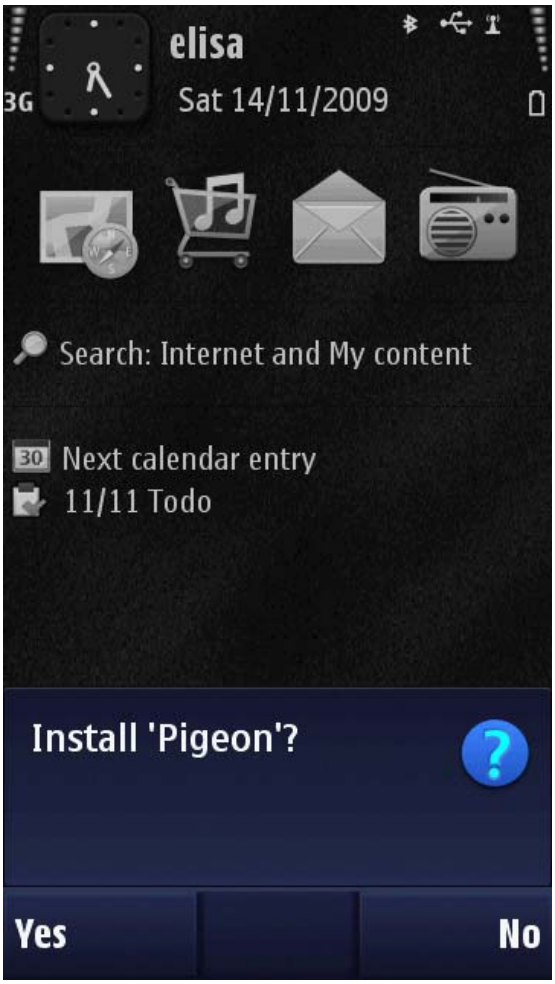
Running the Demo Apps on the Satio Emulator

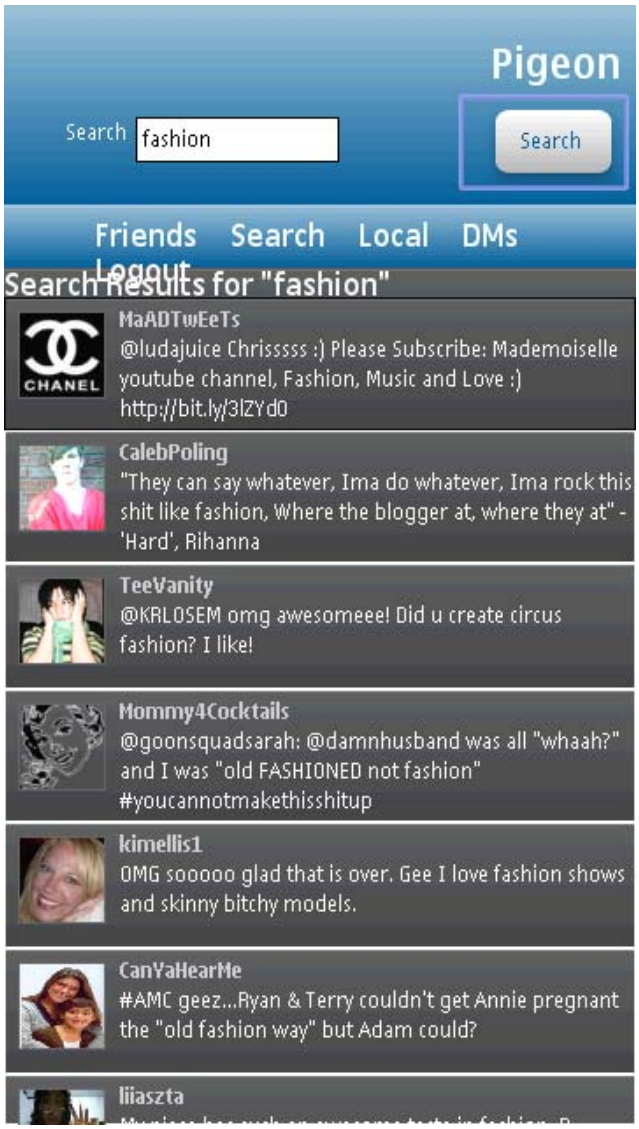
Run the emulator from Start -> Programs -> S60 Developer Tools -> 5th Edition SDK -> v1.0 -> Emulator

Install the Pigeon demo application in the Emulator: File -> Open, and select /samples/Pigeon/Pigeon.wgz. The emulator will then prompt you to install the application. It will be in Applications on the emulator device.

Running the Demo Apps on the Satio Device

Install the Pigeon demo application to the device: send the same Pigeon.wgz file as above to your device via Bluetooth, email, usb, etc. Symbian will recognize the file type and install it. You can also try the PhoneGap API demo app at /samples/PhoneGapDemo/symbian.wrt.demo.app.wgz. This file is a more complete demonstration of the various device functionalities exposed by the PhoneGap API.





Develop your Own Satio Web Application

The sample application source code for Pigeon resides at /samples/Pigeon/www/. These files are just web technologies (html, css, js, etc), save for info.plist. Info.plist is an xml file which tells Symbian about the application. Properties are self-explanatory ... refer to /samples/Pigeon/www/info.plist for an example.

As the application is developed primarily on web technologies, design and functionality can be tested on the fly in a browser. Once you begin to implement Sony Ericsson Phone Gap APIs however, you will need to run them on the emulator or the device to test the device functions.

Other than info.plist, the only thing that will be new to a web developer in these apps is the PhoneGap API, which allows the developer to interact with the device. To make these apis available to your application, simply include phonegap.js, or phonegap-min.js, located in the /libs folder. These APIs should be consistent across all platforms. However note that this project is a work in progress and minor tweaks across platforms are still required. In particular, developers should be aware that each device uses different webkit builds, and different resolutions, so html, css and images may show up differently on different devices.

Refer to the PhoneGap API Documentation for the device functionalities that are exposed (such as Geolocation, Vibration, Contacts, and others). Once you have your web files, phonegap.js, and info.plist complete (or ready to be tested at least) and in a www folder, simply compress the folder into a zip file, and rename its extension from .zip to .wgz. Then, install the file to the emulator or the device, as above.