The Discussion of Cross-platform Mobile Application Development Based on Phone Gap Method

Limei Cui

Qujing Normal University, Qujing Yunnan 655011

Email:756612367@qq.com

Keywords: Phone Gap; JavaScript; Cross-platform; Android; Mobile Application

Abstract. With Phone Gap method, developers can use Web technical which correspond to the W3C standard developing native mobile application. According to call the API of Phone Gap, Web applications can communicate with the SDK API of the mobile platform which using by the developers, and then, they can use the native functions of the mobile platform. Phone Gap provides a method to develop plugins and import plugins, this method makes developers develop their own plugins or import new plugins to meet their special demands. According to the contrast testing, the mobile applications using Phone Gap method have a better performance than the applications using Web View method.

Phone Gap Technical Introduction

Phone Gap is an open source framework currently developed and maintained by the United States Adobe company. Using Phone Gap can be used HTML, CSS, JavaScript and HTML5+CSS3 prepared by the APP Web to build a cross platform for mobile applications. The main prospects is by acting as a bridge between JavaScript and native code with phone gap, not only to web application project conveniently packaged for local applications, but also by other local characteristics of the phone gap plug-in called Android devices, such as camera, geographic information, local network settings, and developers can according to the specific hardware needs and business requirements to develop phone gap plug-ins for their own equipment.

Technical Principle and Implementation Process of Phone gap

Cross Platform Principle of Phone Gap. Using phone gap technology and developers by using CSS and HTML to design the UI and control the program logic; at the same time, use JavaScript to call the API and the corresponding Platform SDK to interact, to activate the phone's GPS, motion, gravity sensors and other functions. Phone Gap prepared using the program will be packaged and installed on the phone, when the application is loaded into the phone's browser control to resolve the implementation of [4].

Phone gap framework for developers to provide a series of API, the API can be invoked in the JavaScript code; in the actual development process, through these API calls, phone gap program can interact with the developers use the Mobile Platform SDK API, and the actual call of the platform of local functions. After the development is completed, the user uses the application, the developer uses HTML technology Web, CSS and JavaScript files will be in the form of "mirror image" is packaged and released to the user's mobile platform. In actual operation, the Web browser embedded in the platform is analyzed and implemented, so as to realize the function of a development, multi platform operation. Phone Gap currently supports almost all mobile applications, such as iOS, Android, Phone Windows, blackberry and WebOS, etc..

Phone Gap Adopt JavaScript Call Java Process. The Phone Gap technology discussed in this paper will take the Android platform as an example, and based on the Phone Gap 1.6, 2.3.6 Android version to explore and test. Below through a simple Hello World program description in the Android platform based on Phone Gap development of the application of the principle of operation. The specific development process will be introduced in the next section.

The following JavaScript code to achieve through the Phone Gap call Android device on the vibration notification function, when the program is running, the user clicks on the main program of the "shock 2 seconds" button, the device will continue to vibrate for 2 seconds. And the realization of this function, only need to use the same as the preparation of Web applications, in the HTML page to add the following line of JavaScript code. And the introduction of phonegap.js and phonegap.jar two class library support.

```
<script type="text/JavaScript" charset="utf-8">
// vibrate for 2 seconds.
Function vibrate () {navigator. notification. Vibrate (2000);}
</script>
<input type="button" value=" Vibrate for 2 seconds. " on click="vibrate ()" />
```

Consider the user to click on the "2 seconds" button on the main program; the JavaScript code is how to call the API Android. When the user clicks on the button, the Notification. prototype. vibrate method in the phonegap.js will receive the call, and then call the Phone Gap. exec method. As shown in the following code section:

```
Notification.prototype.vibrate = function (mills) {Phone Gap. exec(null, null, "Notification", "vibrate", [mills]);};
```

Phone Gap. Exec method is called, it will call prompt (stringify JSON. (args), "+JSON." stringify gap: ([service, action, callback ID, true]) method to complete the communication between JavaScript and Java. At this point, the Web View component of the Android system will attempt to pop up a window. Using Web Chrome Client provided API Android can intercept this action Web View. Specific to the Phone Gap 1.6 is the inheritance of the com. Phone gap. Cordova Chrome Client on Js Prompt class in the Web Chrome Client method.

Execution of plugin Manager. exec (service, action, callbackId, message, async,) methods in the on Js Prompt method. At this point, Plug Manager will be based on the received parameters, will be distributed to a specific command Plugin, that is, the plug-in. About Phone Gap plug-in, will be described in the next section. In this case, the received plugin is Notification; the received action is vibrate; the parameter is 2000 Ms. Thus, Phone Gap call this. vibrate in the Notification (args. get Long (0)) method, notify the device vibration 2000 Ms.

In the notification class vibrate method does not return a value, but in other phone gap plugins, common success and error two basic return value, to return a JSON object form, in order to inform the user, program execution state. And this return value can be captured and output on the main program page. Therefore, the summary of the JavaScript Phone Gap call Java process, as shown below.

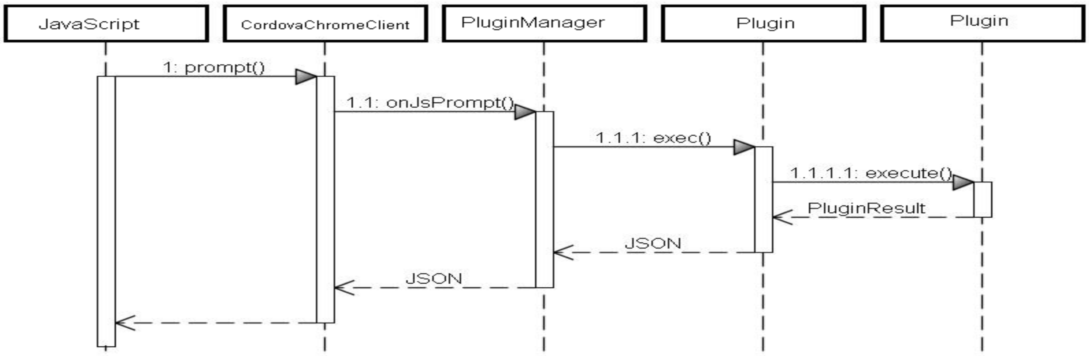

Figure 1. The communication process between JavaScript and Phone Gap in Java

Phone Gap Adopt Java Call JavaScript Process. Phone Gap to achieve a callback server, the server is responsible for the callback JavaScript code; the server has a JavaScript code queue, in the src/com/phone gap/CallbackServer.java file:

```
/**
* The list of JavaScript statements to be sent to JavaScript.
```

```
*/
Private Linked List javascript;
```

Servers store to callback JavaScript code for JavaScript client to retrieve, here the Java side is server-side and client-side JavaScript is the client, server-side impossible client requests, so phone gap realizes two services model, a is the polling, a is XHR asynchronous callback, is also the Ajax model. In the Phone Gap source directory, the CallbackServer.java src/com/phone gap/ file is the callback server code.

Provide Callback Server above two model of a relatively simple principle of polling mode. When in use, a callback JavaScript list of preservation of callback server server will, every time the client JavaScript will ask the server, whether there is a JavaScript callback needs, if there is specific invocation. And based on XHR is mechanism with Ajax, JavaScript initiated an asynchronous request, the server will in return data before keep this connection. When the return data in place, server to client requests to return data, and then close the connection. The client receives the returned data for processing. The client JavaScript's related code is as follows:

```
Phone Gap.JS Call back = function () {
xmlhttp. open("GET",
"http://127.0.0.1:"+PhoneGap.JSCallbackPort+"/"+PhoneGap.JSCallbackToken , true);
xmlhttp. send ();
}
```

This is the code for the XHR model, the client XHR using the JavaScript request server to get the JavaScript code, the callback.

```
Phone Gap. JSCallbackPolling = function () {
var msg = prompt("", "gap_poll:");
if (msg) {
setTimeout (function() {
try {
var t = eval (""+msg);
}
catch (e) {
console.log ("JSCallbackPolling: Message from Server: " + msg);
console.log ("JSCallbackPolling Error: "+e);
}
}, 1);
setTimeout (Phone Gap.JS Call back Polling, 1);
}
else {
set Time out (Phone Gap.JS Call back Polling, Phone Gap.JS Call back Polling Period);
}
}
```

This is a polling mode, you can see the client every Phone Gap. JS Callback Polling Period period of time, on the request of a server (through the prompt ("", "gap poll:"); [8].

Development Process.1. Start Eclipse, and then in the menu "File" select "New > Project Android". In the project root directory, create two new directories: /libs/assets/www

2. Copy the phonegap.js (from the Android after decompression of the Phone Gap directory, will be extracted with the version number of the JS file name change to phonegap.js) to /assets/www.

Copy phonegap.jar (from the Phone Gap extracted from the Android directory) to /libs. Right click the /libs folder to find Paths Build / > Configure Build Paths. And then add the phonegap.jar to the project in the Libraries tab page. Copy the entire directory of XML (from Android to extract the Phone Gap directory, including a plugins.xml) to /res.

3. Minor adjustments to the main Java file in the SRC Eclipse folder:

Change the inheritance of class from Droid Gap to Activity

Replace super. loadUrl () set Content View ("file:///android_asset/www/index.html"); Add to

```
import com. Phone gap.*;
```

Remove import android. app. Activity;

4. Right click AndroidManifest.xml Also choose Open With > Text Editor。Copy the following permission settings to version Name: (in real development, open the program permissions as needed)

5. In the "/assets/www" directory, create a new file "index.html", and paste the following code:

```
<html>
<head>
<title>Notification Example</title>
<script type="text/javascript" charset="utf-8" src="phonegap.js"></script>
<script type="text/javascript" charset="utf-8">
// Vibrate for 2 seconds
function vibrate () {
navigator. notification. Vibrate (2000);
}
</script>
</head>
<body>
<input type="button" value=" Vibrate for 2 seconds" on click="vibrate ()" />
</body>
</html>
```

Deployment and Operation. 1. Right click the project node to select As Run, and then click Application Android.

2. Eclipse will ask you to choose a suitable AVD (Android virtual machine), if not set, you need to create an AVD or an actual Android device connected to the computer.

3. Confirm that the device has checked the debugging USB option, and the device is connected to the computer. (Settings > Application > Development)

4. Right click the project node to select the As Run, click Application Android, wait a moment can be in the AVD or the actual Android devices to see the running effect.

Phone Gap Plug in Development

Phone Gap Plug in. All API of Phone Gap are called and developed in the form of plug-in. Due to the difference in the function of the hardware, such as smart phones and smart tv. So Phone Gap comes with the plug-in does not meet the requirements of the developer for different hardware devices. But Phone Gap provides a robust, convenient plug-in development and introduction, we can for their own special needs to introduce new plug-ins or develop their own plug-ins.

Phone Gap Plug-In Based on the Operation Principle of the Android Platform. In this paper, in the chapter has been introduced based on Android platform phone gap apps running process, it can be said that each phone gap API is a plug-in, call the process also is mentioned above, the JavaScript to invoke a java process.

After the program runs, the page request is submitted to the on JsPrompt method to execute the plugin Manager. exec (service, action, callback ID, message, async) methods. Plug Manager will be based on the receipt of the parameters, the command will be distributed to a specific Plugin, and at this time you need to configure the project is located in the XML../res/xml/plugins.xml file, the declaration of the name of the Java class. In order to load the plug-in, and to the plug-in to a specific command, after the end of the implementation of the command plug-in, if there is a return value, then return. The return method as described in section 2.3.

Phone Gap Performance Comparison Test of Plug-In and Web View Mode

Test Purpose. Before using phone gap, most of the web application to are based on local application of Android platform comes with Android Web View. By comparing the Web View native support Java and JavaScript communication mode and supported by the phone gap plug-in communication can provide a reference for developers and using phone gap technology to provide theoretical basis. It is well known that in the mobile platform, system resources is very tense, how under the given system resources to expand program performance and efficiency are all mobile applications must pay attention to the problem, this test also focuses on the comparative use different techniques when resource cost, performance and efficiency.

Test tool: the use of SDK Monkey comes with the test tool Monkey, Android is a command line tool in SDK Android, you can run in the simulator or the actual device. It sends the pseudo random user event flow (such as the key input, touch screen input, gesture input, etc.), to realize the pressure test on the application being developed. Monkey test is a rapid and effective method to test the stability and robustness of software [7].

Test Case. In order to ensure the accuracy of the test, using the Web View method and the Phone Gap plug-in mode, respectively, developed two simple adjustment system sound size and display the small application.

The Web View, to use the HTML + JavaScript development events page calls to the Android system function is Android Web View component of add java script interface method. This method capture HTML page events and calls the corresponding Android native methods will ultimately result in a JavaScript call to return.

The use of Phone Gap, the need to introduce the Phone Gap dependent libraries, and then development. Phone Gap in the application will be loaded with the required plug-ins, and then according to the different pages of the corresponding event call Android native methods. Calls for the capture of the page prompt event, passing the request parameters and return parameters. Different from the direct use of Web View, the use of Phone Gap can be packaged in a good way to Java and JavaScript communication, code logic is clear, easy to develop and modify.

Test Result. The use of Monkey to test the use of the 10000 cases of the injection, observe its performance and response as shown in the following table.

Through the table of test results, we can see that using phone gap plug-in Java and JavaScript to call each other by 10000 times required time 208465ms, memory 9.0MB, with respect to the Web View 180802ms and 7.3MB memory footprint and its efficiency is clearly to be slightly lower. However, taking into account that Phone Gap is loaded with all of its core plug-ins in the initialization, it is necessary to streamline this part of the comparison test.

Improved test Results. When the application is loaded by the system, the Phone Gap is automatically initialized to load all the core plug-ins, but most of these plug-ins are useless for our program. Delete many of the Phone Gap core plug-ins, such as GPS, network, compass, etc., which are independent of the test cases, can reduce the system resources occupied by Phone Gap in theory. After deleting the plug-in to be tested again, it can be found that the improved Phone Gap test case for 10000 times the random event injection time consuming 198214ms, after the test is completed in memory 5.3MB. Although the time consuming is still larger than the test case based on Web View, but the consumption of memory is significantly reduced. It can be seen that, based on the Phone Gap test case after optimizing the streamlining, efficiency greatly improved.

Phone gap Technical Summary

Phone gap advantage is obvious, cross platform, easy to use, especially currently using JavaScript and HTML5+CSS3 technology of the developers to, the existing web app quickly ported to a variety of popular mobile application platform. More importantly, through the Phone Gap plug-in mode, you can make the complex business into the local code execution, not only improve efficiency, but also enhance the reliability of the code.

References

[1] Andrew Lunny. Phone Gap Binger's Guide [M]. Birmingham B3 2PB, UK. Published by Packt Publishing Ltd. 2011.9.21-22

[2] Thomas Myer. Wrox Beginning Phone Gap [M]. Indianapolis. John Wiley & Sons, Inc.2011.11.

[3] Exploration on the framework of cross platform mobile application development – Phone Gap http://www.ibm.com/developerworks/ cn/open source/os-cn-phone gap/

[4] Analysis of how to use the Phone Gap plug-in http://mobile.51cto.com/others-290644.htm

[5] Gap Phone development two: the development of a Gap Phone plug-in http://gteam-yu.iteye.com/blog/1358707

[6] Android automatic test Monkey http://www.cnblogs.com/yyangblog/archive/2011/03/10/ 1980068 .html

[7] Phoen gap source code analysis, plug-in mechanism, Java and JS code with detailed intermodulation.http://www.qhm123.com/2012/01/28/phonegap-source-code-analyzing-java-js-plugin-mechanism.html

[8] Yuan Qi, cross platform embedded development environment research [J], computer and telecommunications, 2008 11

[9] Li Chunhu, cross platform software design and application based on [D] Qt, University of Electronic Science and technology, 2011

[10] Zhao Shenghai, the research and application of design pattern in embedded software design [D], Sichuan University, 2006

[11] Zhong Wen, network management system of University Alumni Network [D], South China University of Technology, 2012