



A Sample PhoneGap Application Using SUP

This document summarizes the creation of a PhoneGap application on android platform which uses SUP server to fetch the data. This document also describes the basics of PhoneGap from the environment setup, application creation to the application deployment through 'PhoneGap Build'.



Kamlendra Singh Chauhan
Developer Associate, Mobiliser, SAP Mobile Platform India

TABLE OF CONTENTS

INTRODUCTION	3
PREREQUISITES AND SETUP.....	3
Requirements for development.....	3
Set environment variables.....	3
SUP server configurations.....	3
APPLICATION CREATION AND DEPLOYMENT	3
REFERENCES	7

INTRODUCTION

PhoneGap is an open source mobile development platform that enables programmers to build applications for mobile devices using JavaScript, HTML5, and CSS3. The resulting applications are neither purely native nor truly web based. The advantage is no platform dependency which allows us to use same application, created on one platform to be used on other platform as well.

This document describes creation of a basic PhoneGap application which fetches data from gateway using SUP server.

PREREQUISITES AND SETUP

Requirements for development

- Java JDK
- Android SDK
- Eclipse, Eclipse ADT plugin
- Android Platform and Components
- PhoneGap Framework

Set environment variables

The following paths should be set in your account's PATH variable:

- your_system_path/jdk/bin
- your_system_path/android-sdk/tools

Apart from these, you will need to set the following variables also:

- JAVA_HOME – path of your JDK directory
- ANDROID_HOME – path to your android SDK directory.

For more information on above points please [click here](#).

SUP server configurations

Login to SUP server, create the settings with proper application, security configuration (here we are using basic authentication), domain and whitelist proper backend URL for fetching data.

For more information on how to work with SUP server and configurations please [click here](#).

APPLICATION CREATION AND DEPLOYMENT

Once the environment and eclipse setup is done, we can start with the code.

An XMLHttpRequest object is being used for the request/response. After setting the necessary headers the request would be sent out to the SUP server with proper authentication and in turn to the gateway.

1. In the assets/www folder of the workspace, there will already be a file called phonegap.js. In that folder create a file called index.html with the following code:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Cordova</title>
<script type="text/javascript" src="datajs-1.0.1.min.js"></script>
```

```
<script type="text/javascript" charset="utf-8" src="cordova-1.9.0.js"></script>
<script type="text/javascript" charset="utf-8">

    // Wait for Cordova to load
    document.addEventListener("deviceready", onDeviceReady, false);

    // Cordova is ready
    function onDeviceReady() {
    }
    // process the confirmation dialog result
    function onConfirm(buttonIndex) {
    }
    // Show a custom alert dialog
    function showConfirm() {
        var request = new XMLHttpRequest();
        var text;
        var i;
        request.open("GET", "URL", true);
        request.setRequestHeader("Authorization","Base 64 encoded
        username:password");
        request.setRequestHeader("Content-Type");
        request.setRequestHeader("security Configuration");
        // Optional, In case the security configuration which we want
        use, is set as default in SCC.
        request.onreadystatechange = function() {
            if (request.readyState == 4) {
                if (request.status == 200 || request.status == 0) {
                    // -> request.responseText <- is a result
                    navigator.notification.alert(
                        'The Response String is ' + request.responseText, // message
                        onConfirm, // callback to invoke with index of
button pressed
                        'PHONEGAP DEMO', // title
                        'OK' // buttonLabels
                    );
                    text = request.responseText; // print the response text
                }
            }
        }
        request.send();
    }
}
</script>
</head>
<body>
<h1>Sample Data </h1>

<button type="button" onclick="showConfirm()">Try it</button>
</body>
</html>
```

Note:

The URL should be like

http://<Host>:<Port>/<Application ID>/<Collection/entry>

Ex.: http://a.b.c.d:8000/com.sap.app/Collection ('ID')/\$value

2. For deployment on Simulator right click the project and go to **Run As > Android Application**. Eclipse will ask you to select an appropriate AVD. If there isn't one, then you'll need to create it.
3. For making it work on multiple devices, 'PhoneGap Build' can be used. Login and Upload the app in any of the two forms; either a zip file of the project or connect it with git repository (by entering the git repository path).
As soon as we will provide the details, the PhoneGap build will start the compilation of the project and if everything is in place, compilation will be successful. As shown in the "Figure 1" some icons are blue and some are red. The red one shows the compilation failure and blue ones successful compilation and generation of the corresponding executable for the platform. Just click on any blue icon to download the file.

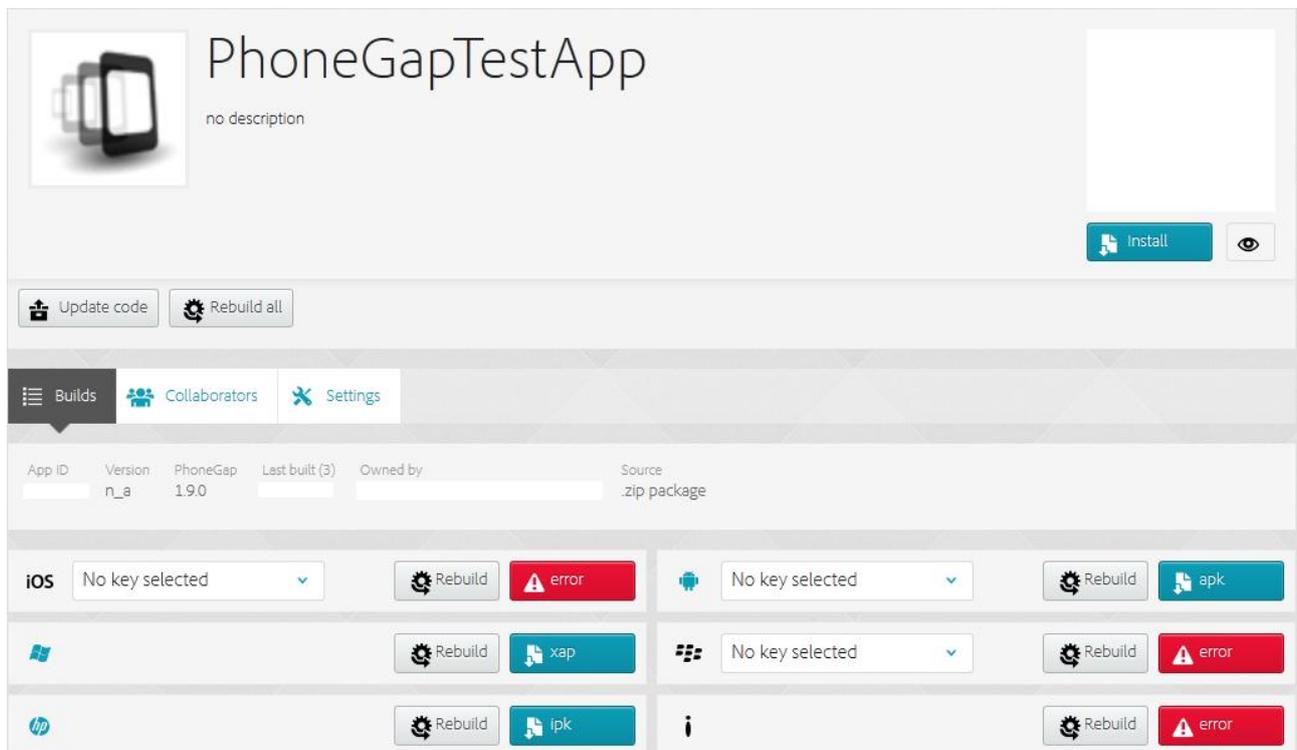


Figure 1

Click on any of the red icons. It shows what the problem with the compilation is.

Example: For iOS the provisioning profile is a must along with .p12 certificate. As soon as everything is in place the corresponding .ipa file will be available to download. Check out 'Figure 2' for this.

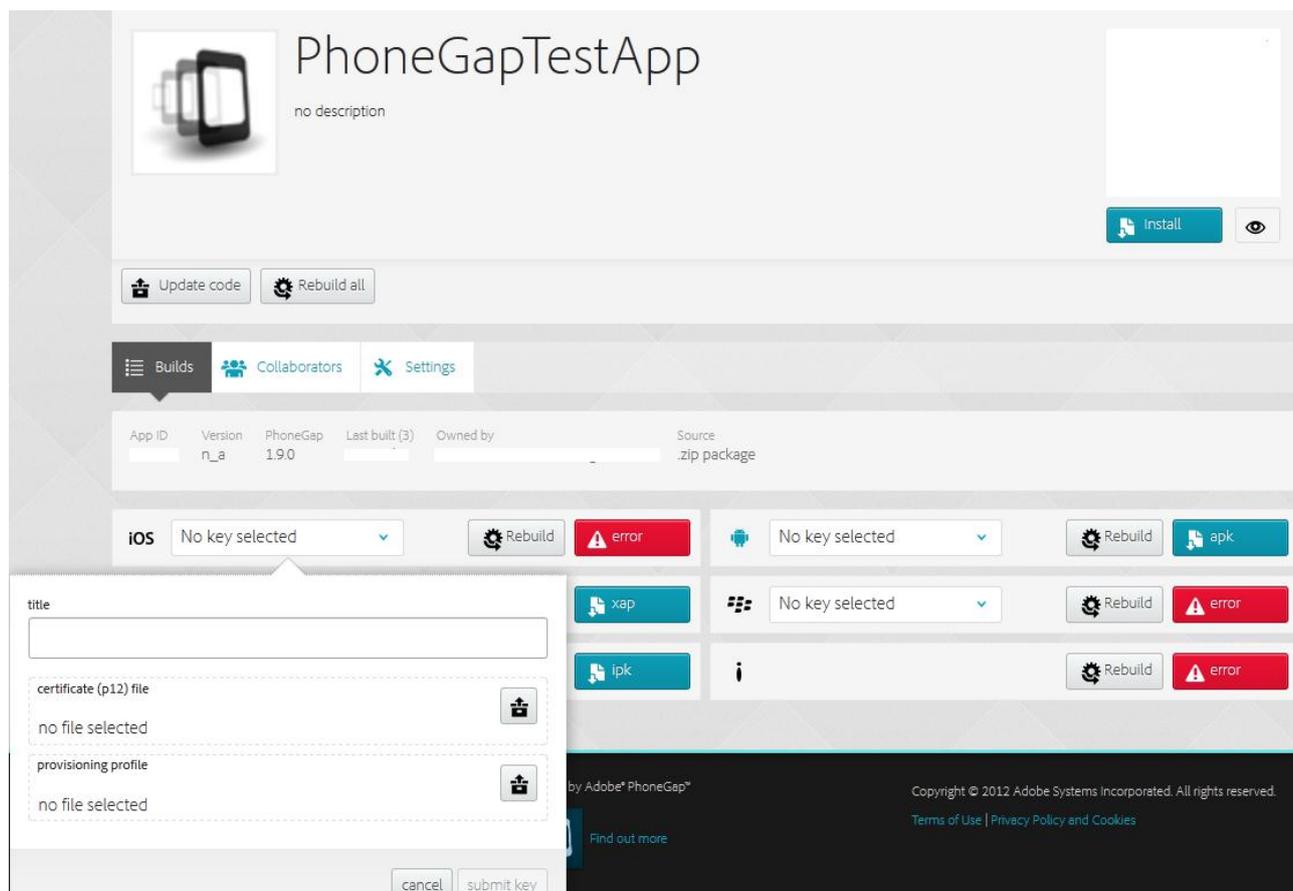


Figure 2

Note: Please follow the steps given on the reference [3] for creating the provisioning profile and other necessary certificates.

4. After downloading the executable file on the Device, install it and run.

REFERENCES

- [1].For development environment setup [Click here](#).
- [2].To access and know more about Phone Gap Build Tool [Click here](#).
- [3].For iOS platform [Click here](#) to know about provisioning profile and certificates.
- [4]. [Beginners Guide for Online Mobile Application Development using SUP2.1-ODP](#)

© 2012 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

