



# Robillion API

July 6th, 2011

*Getting started with  
Robillion for PhoneGap  
Android Plugin*



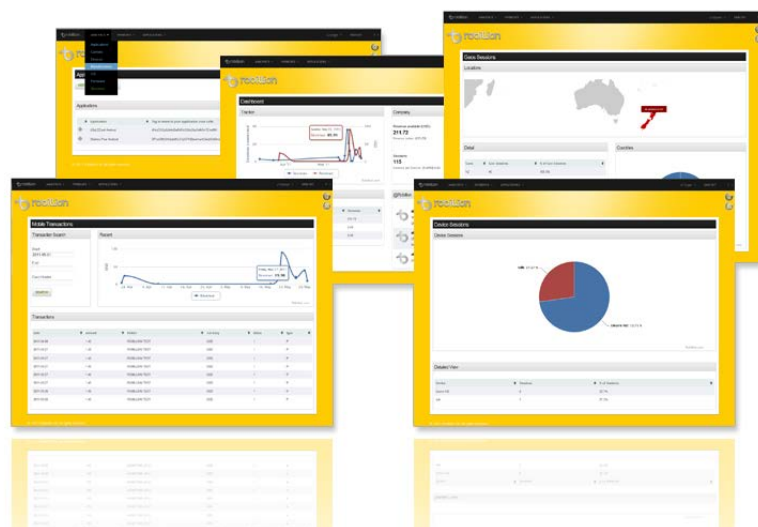
HTML5



CSS3



JavaScript



This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Robillion Ltd on the issues discussed as of the date of publication. Because Robillion must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Robillion, and Robillion cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. **ROBILLION MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.**

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Robillion Ltd.

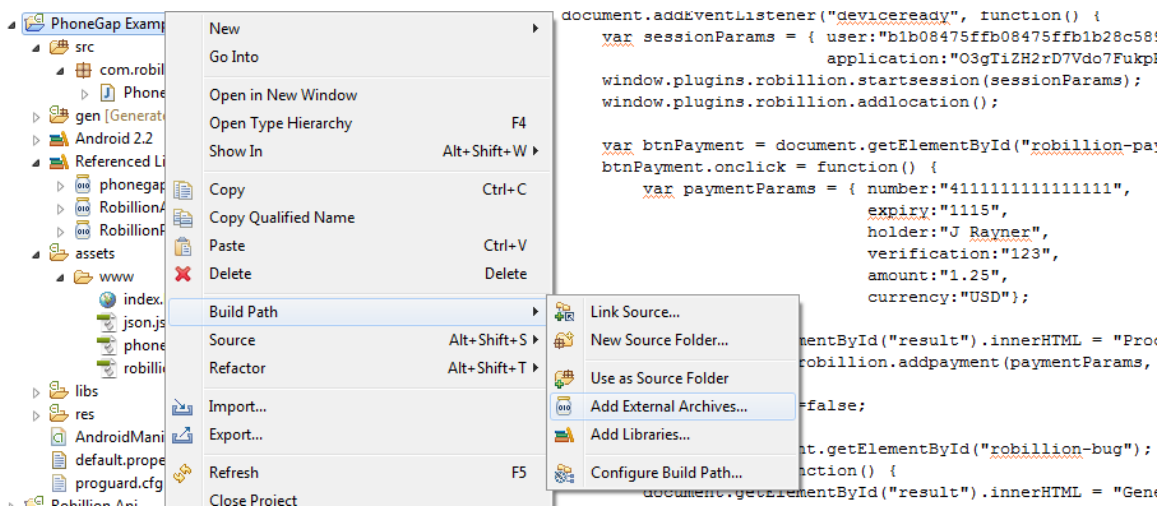
Robillion may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Robillion, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

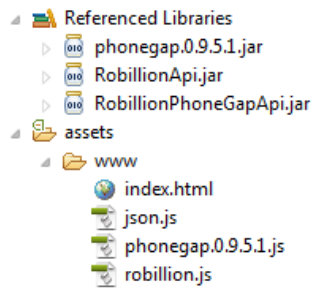
## Adding Robillion API

To add Robillion analytic and/or payment capabilities to your PhoneGap application, you will need to add both the RobillionApi.jar and the RobillionPhoneGapApi.jar to your application's classpath.

- If you're using Eclipse, modify your Java Build Path, and choose Add External Archives...



Next you need to add the Robillion.js file to your assets folder. Typically you may have created a WWW subfolder



## Adding permissions to AndroidManifest.xml

In order for Robillion to send analytic information or request payment processing the internet permission must be set for your project. Optionally, permissions can be set for location analytics.

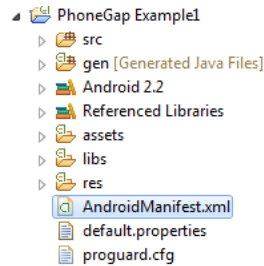
```
<uses-permission android:name="android.permission.INTERNET" />
```

Optional, location...

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

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

In eclipse simple double click on the AndroidManifest.xml document in your project tree.



Then add the appropriate permission entries to your xml,

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.robillion.api.phonegap.example1"
    android:versionCode="1"
    android:versionName="1.0">

    <supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:resizeable="true"
        android:anyDensity="true"
    />

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

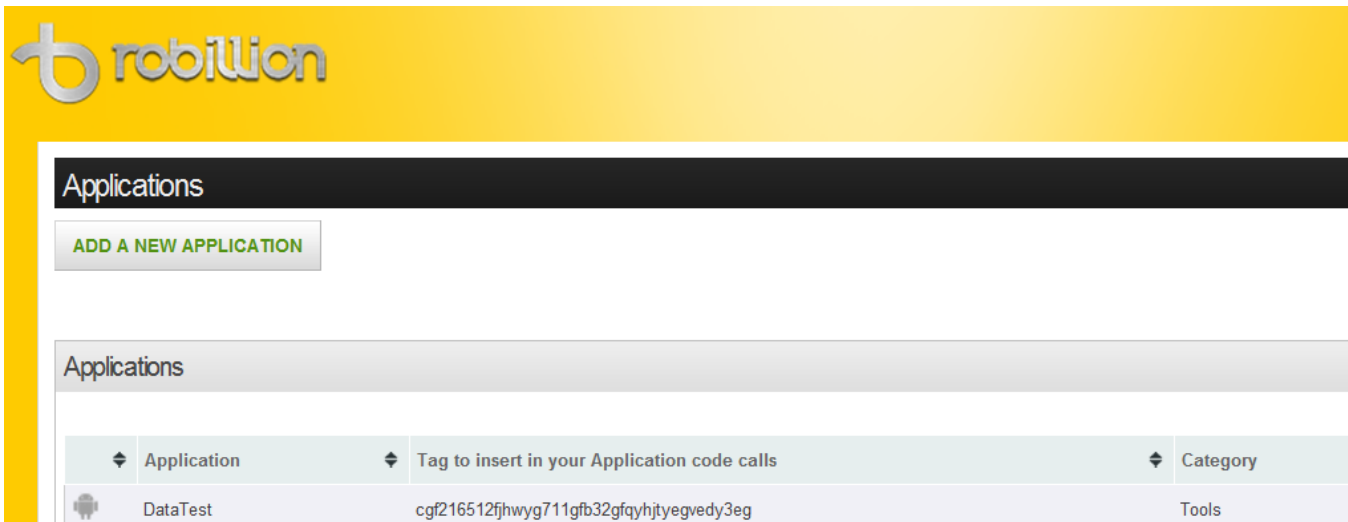
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMM" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission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

## Adding StartSession

Next we want to add the basic call to start our Robillion session. Firstly we need a few important variables to define who we are and to give us access to Robillion.

```
var sessionParams = { user:"b1b08475ffb08475ffb1b28c5890100b52161f97",
                      application:"O3gTiZH2rD7Vdo7FukpK6dOatRPVE7jMGB7kzBgT"};
```

Add the two RobillionAgent session string variables. USER is the 40 character tag you were given when you first joined Robillion, for your company. The APPLICATION is the tag your application is given if you open your dashboard and navigate to the manage applications you will see this tag. Every application has a unique 40 character application tag, used by the Robillion service to open a user session for a particular application.

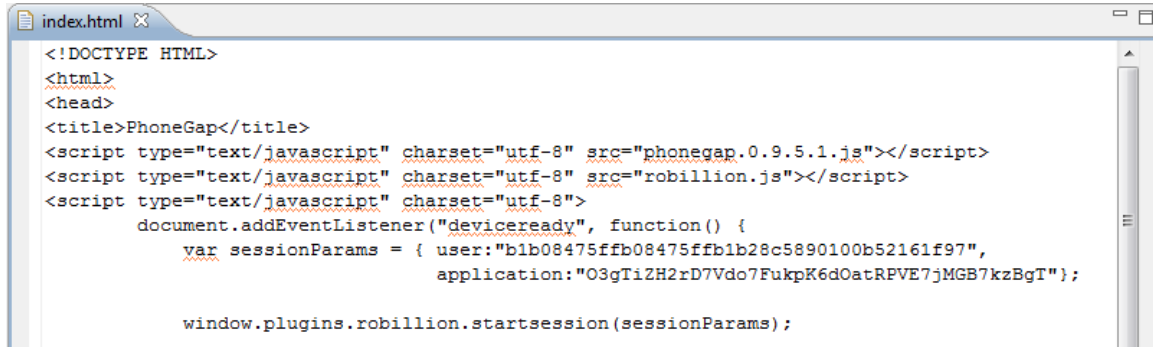


The screenshot shows the Robillion dashboard interface. At the top left is the Robillion logo. Below it is a header bar with the word "Applications" and a button labeled "ADD A NEW APPLICATION". The main content area is titled "Applications" and contains a table with the following data:

Application	Tag to insert in your Application code calls	Category
DataTest	cgf216512fjhwyg711gfb32gfqyhjtyegvedy3eg	Tools

Next, in your html you need to add StartSession call. We recommend you add these as early as possible to your main/first form. Robillion uses a singleton for session scope, this helps removes false session starts from multiple activities being called with StartSession included.

example StartSession method,



```
index.html x
<!DOCTYPE HTML>
<html>
<head>
<title>PhoneGap</title>
<script type="text/javascript" charset="utf-8" src="phonegap.0.9.5.1.js"></script>
<script type="text/javascript" charset="utf-8" src="robillion.js"></script>
<script type="text/javascript" charset="utf-8">
  document.addEventListener("deviceready", function() {
    var sessionParams = { user:"b1b08475ffb08475ffb1b28c5890100b52161f97",
                          application:"O3gTiZH2rD7Vdo7FukpK6dOatRPVE7jMGB7kzBgT"};

    window.plugins.robillion.startsession(sessionParams);
  });
</script>
</head>
</html>
```

## Adding Location

The more a developer or marketer knows about their market the better the software company can target new features, promotions and support. Robillion has a number of extra analytics on the way, this current build includes Location recording.

Simple add one extra line `window.plugins.robillion.addlocation()`, after your `startsession` call and your done.

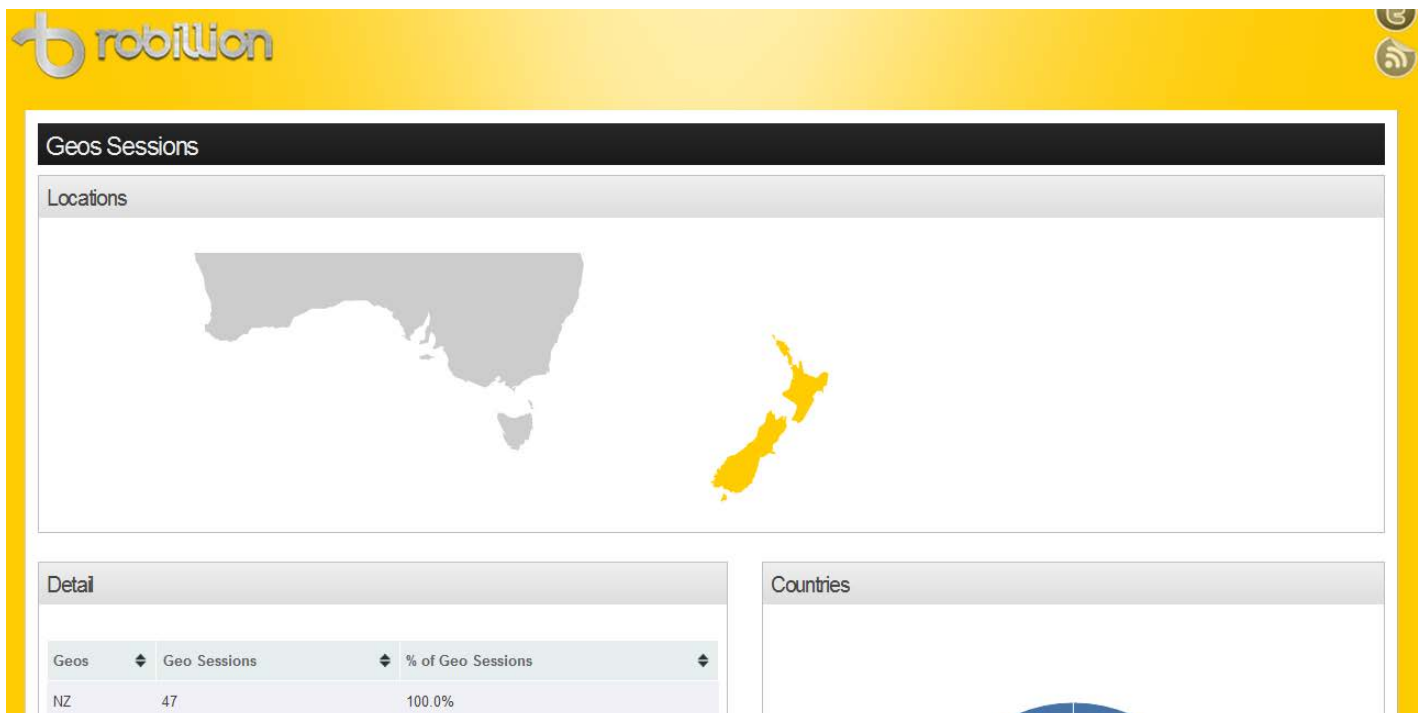
```

index.html X
<!DOCTYPE HTML>
<html>
<head>
<title>PhoneGap</title>
<script type="text/javascript" charset="utf-8" src="phonegap.0.9.5.1.js"></script>
<script type="text/javascript" charset="utf-8" src="robillion.js"></script>
<script type="text/javascript" charset="utf-8">
  document.addEventListener("deviceready", function() {
    var sessionParams = { user:"b1b08475ffb08475ffb1b28c5890100b52161f97",
      application:"O3gTiZH2rD7Vdo7FukpK6dOatRPVE7jMGB7kzBgT"};

    window.plugins.robillion.startsession(sessionParams);
    window.plugins.robillion.addlocation();
  });

```

You will now see location analytics in your dashboard,



## Adding Payments

Due to the asynchronous nature of a credit card/payment systems, adding transactions requires a little more in-depth operation handling. Luckily Robillion and PhoneGap have added easy to use functions to encapsulate this for you, but still ensure you can skin the complete payment system in keeping with your particular app/service.

For a minimal solution, as more complex UI's is out of scope for this document, but available in the open source examples on our developer resources page.

```
var paymentParams = { number:"4111111111111111",
                      expiry:"1115",
                      holder:"J Rayner",
                      verification:"123",
                      amount:"1.25",
                      currency:"USD"};

document.getElementById("result").innerHTML = "Processing...";
window.plugins.robillion.addpayment(paymentParams, onPaymentFail, onPaymentSuccess);
```

Currency 3 character string must be from the following list, or Robillion will default to US dollars (USD).

ISO currency code - NZD, AUD, FJD, USD etc.

CAD	Canadian Dollar
CHF	Swiss Franc
EUR	Euro
FRF	French Franc
GBP	United Kingdom Pound
HKD	Hong Kong Dollar
JPY	Japanese Yen
NZD	New Zealand Dollar
SGD	Singapore Dollar
USD	United States Dollar
ZAR	Rand
AUD	Australian Dollar
WST	Samoan Tala
VUV	Vanuatu Vatu

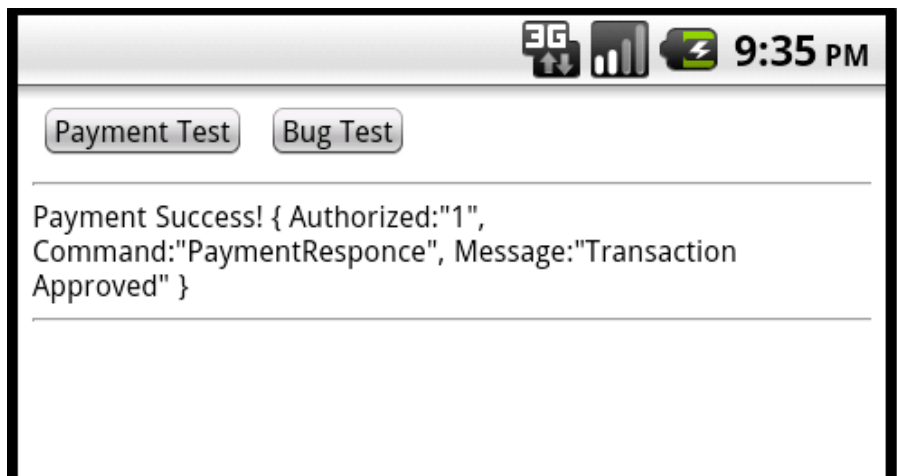


TOP	Tongan Pa'anga
SBD	Solomon Islands Dollar
PNG	Papua New Guinea Kina
MYR	Malaysian Ringgit
KWD	Kuwaiti Dinar
FJD	Fiji Dollar

It may take a short period for Transactions to process. Robillion provides easy to use JS callback mechanisms that execute when the transaction results are known. The returned messages are in JSON allowing for ease of processing.

```
function onPaymentSuccess(r) {
    document.getElementById("result").innerHTML = "Payment Success!\n" + r;
}

function onPaymentFail(r) {
    document.getElementById("result").innerHTML = "Payment Failed!\n" + r;
}
```



## Adding Issue Management

It is often a huge hassle to get errors from your apps/customers to flow through to your testing, support and developers. The Robillion plugin makes this easy. One line of code in your exception catches sends issue tracking info to your issue/project management system.

Currently JIRA Project Management by Atlassian and Email are support. We are actively working with project management companies to build connectors. Contact Robillion support to get your mechanism included.

```
var btnBug = document.getElementById("robillion-bug");
btnBug.onclick = function() {
  document.getElementById("result").innerHTML = "Generating Bug...";
  try {
    blah;
  } catch (e) {
    logError(e);
  }
}
btnBug.disabled=false;
```

Adding a quick logging function to your js, sends all the relevant details plus all of the Robillion goodness, like device, carrier, OS, firmware details automatically to your management suite.

```
function logError(e) {
  var msg = "Error name: " + e.name + "\nError message: " + e.message;
  var func = arguments.callee.caller;
  if (func) {
    msg = msg + "\nError code: " + func.toString();
  }

  var bugParams = { project:"TST",
                  type:"1",
                  title:"Mobile Error",
                  message:msg };

  window.plugins.robillion.addbug(bugParams);
}
```

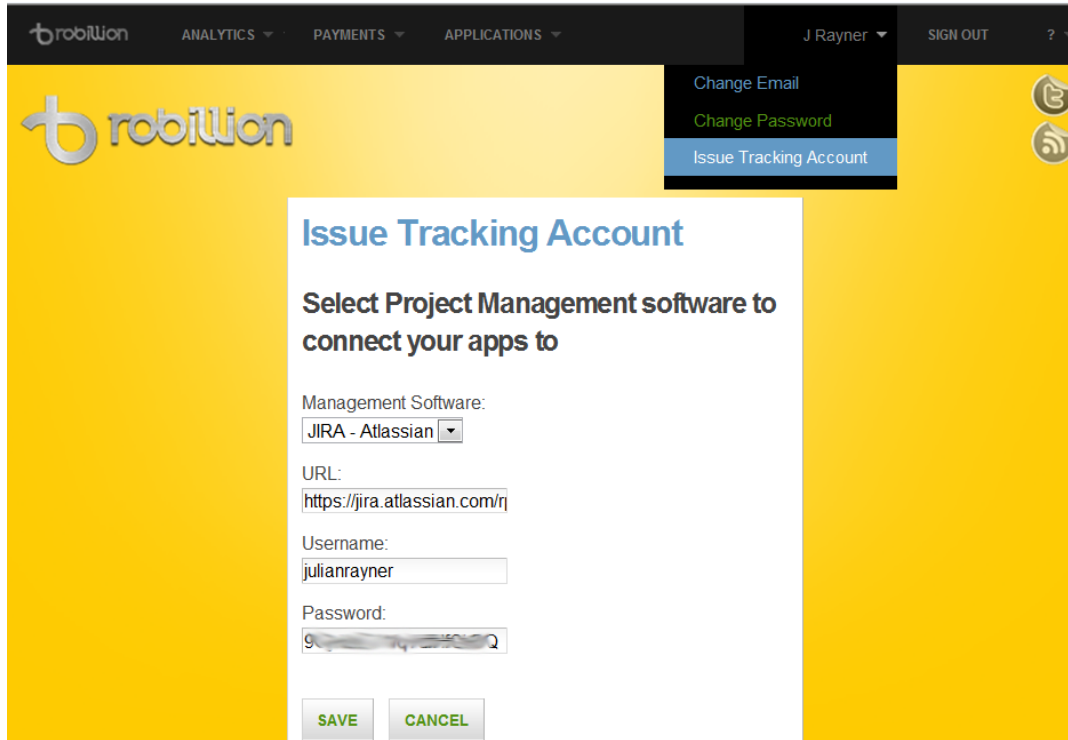
The error, plus the offending code is presented directly as a bug ticket,

The screenshot shows a JIRA issue ticket titled "Mobile Error" under the project "A Test Project / TST-28532". The ticket includes a description of the error, the offending code, and device information. The description section is expanded, showing the error details and the JavaScript code that caused the error. The activity section is currently empty, indicating no comments or work logs have been added yet.

**Issue Details:**

- Project: A Test Project / TST-28532
- Issue Title: Mobile Error
- Buttons: Edit, Assign, Comment, More Actions, Resolve Issue, Close Issue, Workflow
- Description:
  - Error name: ReferenceError
  - Error message: Can't find variable: blah
  - Error code: function () { document.getElementById("result").innerHTML = "Generating Bug..."; try { blah; } catch (e) { logError(e); } }
  - manufacturer: unknown
  - device: sdk
  - carrier: Android
  - os: Android
  - rom: generic/sdk/generic/2.2/FRF91/43546:eng/test-keys
- Activity: All, Comments, Work Log, History, Activity
- Activity Content: There are no comments yet on this issue.

You do need to add the connector details, for errors to automatically flow through. Simply login to Robillion and set your preferred Issue/Project management suite



The screenshot shows the Robillion web interface. At the top, there is a navigation bar with the Robillion logo, menu items for ANALYTICS, PAYMENTS, and APPLICATIONS, and a user profile for J Rayner with a SIGN OUT option. A dropdown menu is open, showing options for Change Email, Change Password, and Issue Tracking Account (which is highlighted). The main content area features a yellow background with the Robillion logo and a central white modal window titled 'Issue Tracking Account'. The modal contains the instruction 'Select Project Management software to connect your apps to' and a form with the following fields: 'Management Software' (a dropdown menu set to 'JIRA - Atlassian'), 'URL' (a text input field containing 'https://jira.atlassian.com/'), 'Username' (a text input field containing 'julianrayner'), and 'Password' (a password input field). At the bottom of the modal are 'SAVE' and 'CANCEL' buttons.