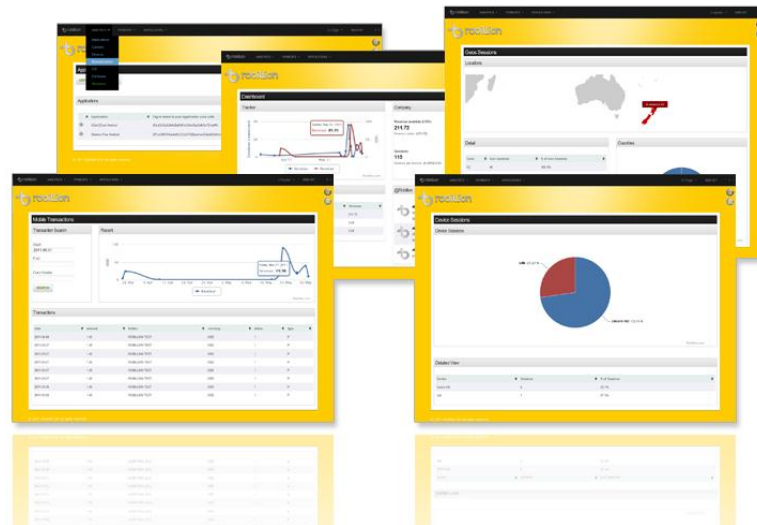


Robillion API

June 21, 2011

Getting started with Robillion for Windows Phone

Julian Rayner, Architect, Robillion Ltd



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.**

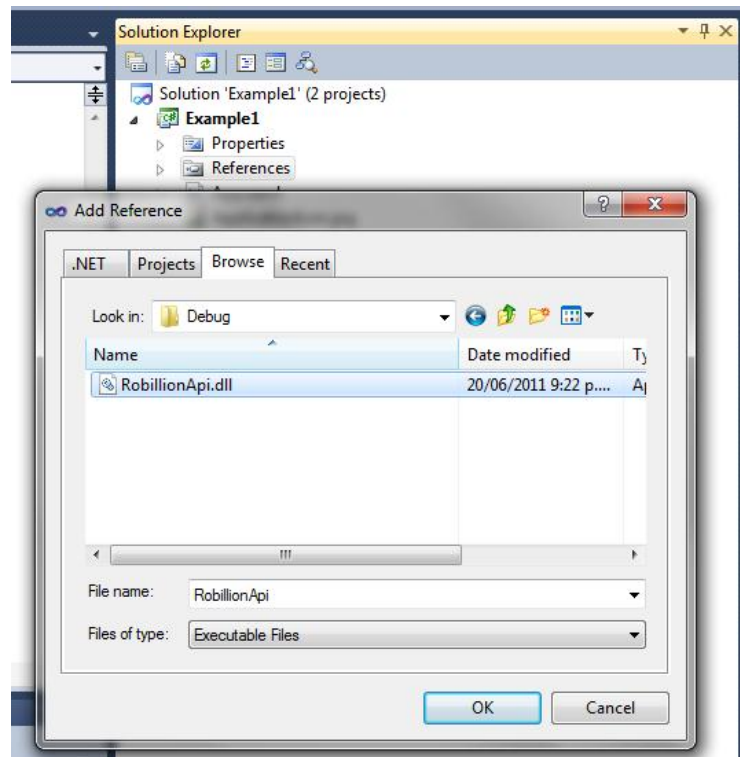
Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording,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

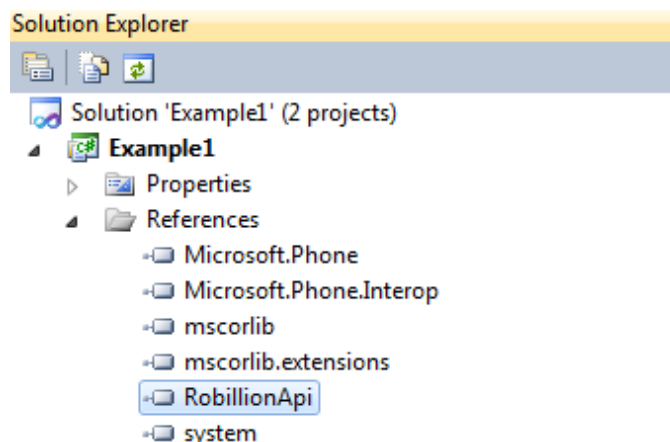
Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended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

Adding Robillion API

To add Robillion analytic and/or payment capabilities to your application, you will need to add the RobillionApi.dll to your application: You can easily do this by right clicking on References in your project in the Solution Explorer. Select 'Add Reference', the select the Browse tab and locate the Robillion dll..

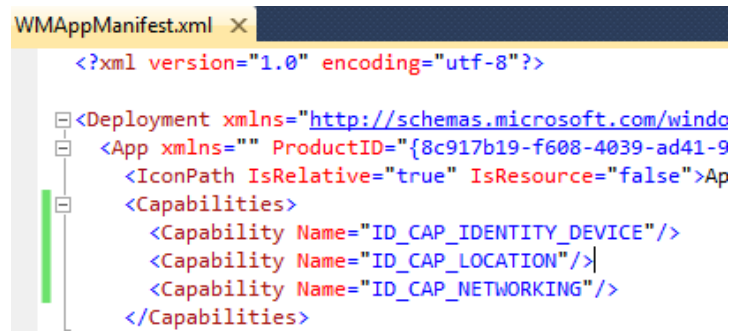


Resulting in the api being part of your references,



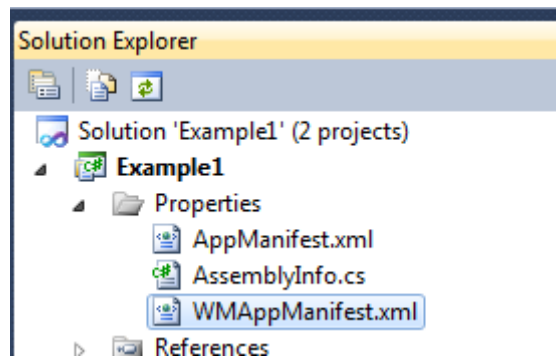
Adding permissions to WMAAppManifest.xml

In order for Robillion to send analytic information or request payment processing the internet and other permissions must be set for your project.



```
WMAAppManifest.xml x
<?xml version="1.0" encoding="utf-8"?>
<Deployment xmlns="http://schemas.microsoft.com/windo
<App xmlns="" ProductID="{8c917b19-f608-4039-ad41-9
  <IconPath IsRelative="true" IsResource="false">Ap
  <Capabilities>
    <Capability Name="ID_CAP_IDENTITY_DEVICE"/>
    <Capability Name="ID_CAP_LOCATION"/>
    <Capability Name="ID_CAP_NETWORKING"/>
  </Capabilities>
```

You will find the manifest file here



Adding onStartSession and onEndSession

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

```

MainPage.xaml.cs
Example1.MainPage
using Robillion.Api;

namespace Example1
{
    public partial class MainPage : PhoneApplicationPage
    {
        public const string ROBILLION_USER = "b1b08475ffb08475ffb1b28c58901e"
        public const string ROBILLION_APPLICATION = "uR0w1gkBNwW3kIvI3oJSD5S7xkGNRF"
    }
}

```

Add the RobillionAgent instance, and the two string variables. ROBILLION_USER is the 40 character tag you were given when you first joined Robillion, for your company. The ROBILLION_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.

Application	Tag to insert in your Application code calls	Category
DataTest	cgf216512fjhwj_	Tools
Rim_Example	0ldHQlzKxTjB72ebfkbNd7M	Games - Casual
iPhone_Example	07TOqrXtD4BHpLQRcuHdcQXN	Games - Arcade & /
WP7_Example1	uR0w1gkBNwW3kIvI3oJSD5S7xkGNRF	Travel

Next, in your constructor or early in your application startup you should add calls to Robillion's `onStartSession`. Likewise, as late as possible in your application process add a call to `onEndSession`. You can call either method multiple times, as Robillion will simply update session length based on final close method call.

```
// Constructor
public MainPage()
{
    InitializeComponent();

    RobillionAgent robillion = RobillionAgent.GetInstance;
    robillion.onStartSession(ROBILLION_USER, ROBILLION_APPLICATION);

    robillion.onEndSession();
}
}
```

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 `robillion.addLocation()`, to your app after the `onStartSession` call and your done.

```

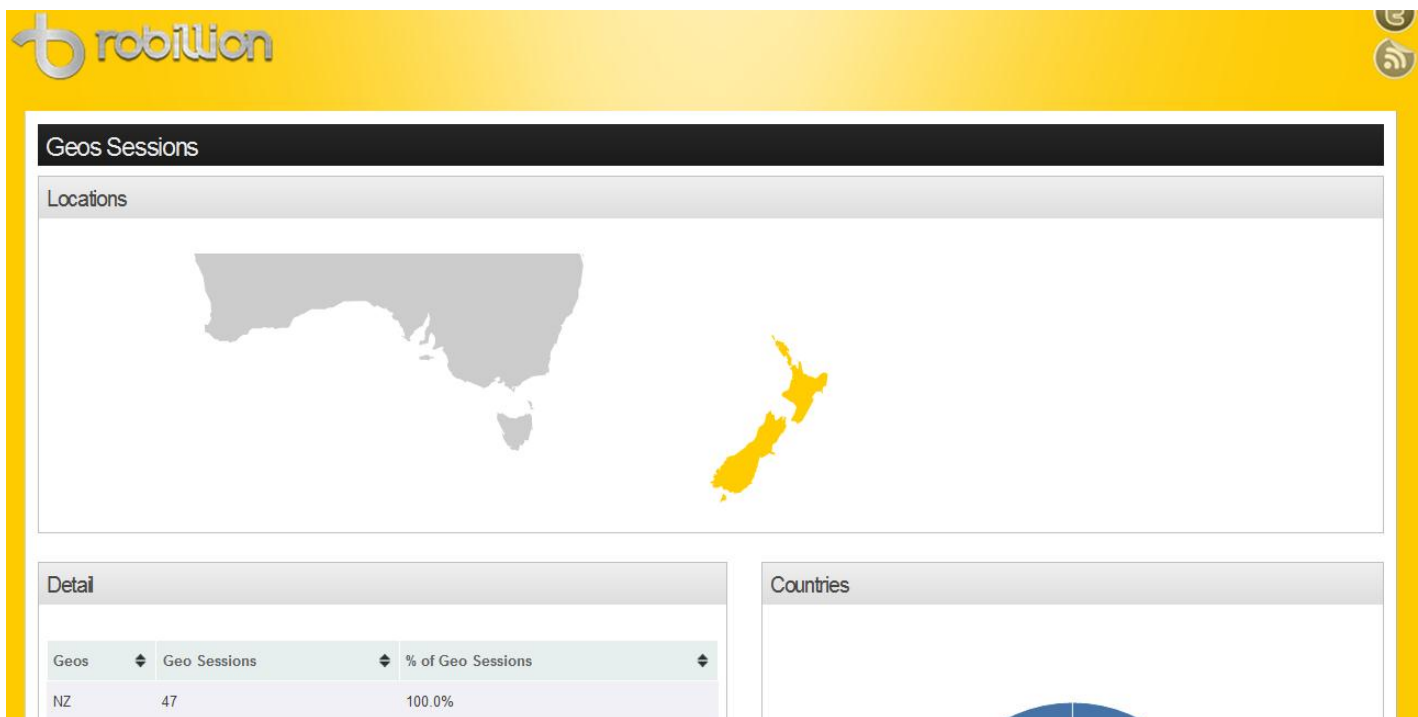
// Constructor
public MainPage()
{
    InitializeComponent();

    RobillionAgent robillion = RobillionAgent.getInstance;
    robillion.onStartSession(ROBILLION_USER, ROBILLION_APPLICATION);
    robillion.onAddLocation();

    robillion.onEndSession();
}

```

You will now see location analytics in your dashboard,



Adding Payments

Due to the asynchronous nature of a credit card/payment system. Adding payments requires a little more in-depth operation handling. Luckily we 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 UIs is out of scope for this document, but available in the open source examples on our developer resources page.

```

// Constructor
public MainPage()
{
    InitializeComponent();

    RobillionAgent robillion = RobillionAgent.GetInstance();
    robillion.onStartSession(ROBILLION_USER, ROBILLION_APPLICATION);
    robillion.onAddLocation();

    robillion.onPayment("WP SEVEN",           // Card Holder
                       "4111111111111111", // Card Number
                       "22.12",            // Transaction amount
                       "USD",              // Transaction Currency
                       "1115",             // Card Expiry
                       "123");             // Cvc2

    robillion.onEndSession();
}

```

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



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 framework makes this easy. One line of code in your exception catches sends issue tracking info to your issue/project management system.

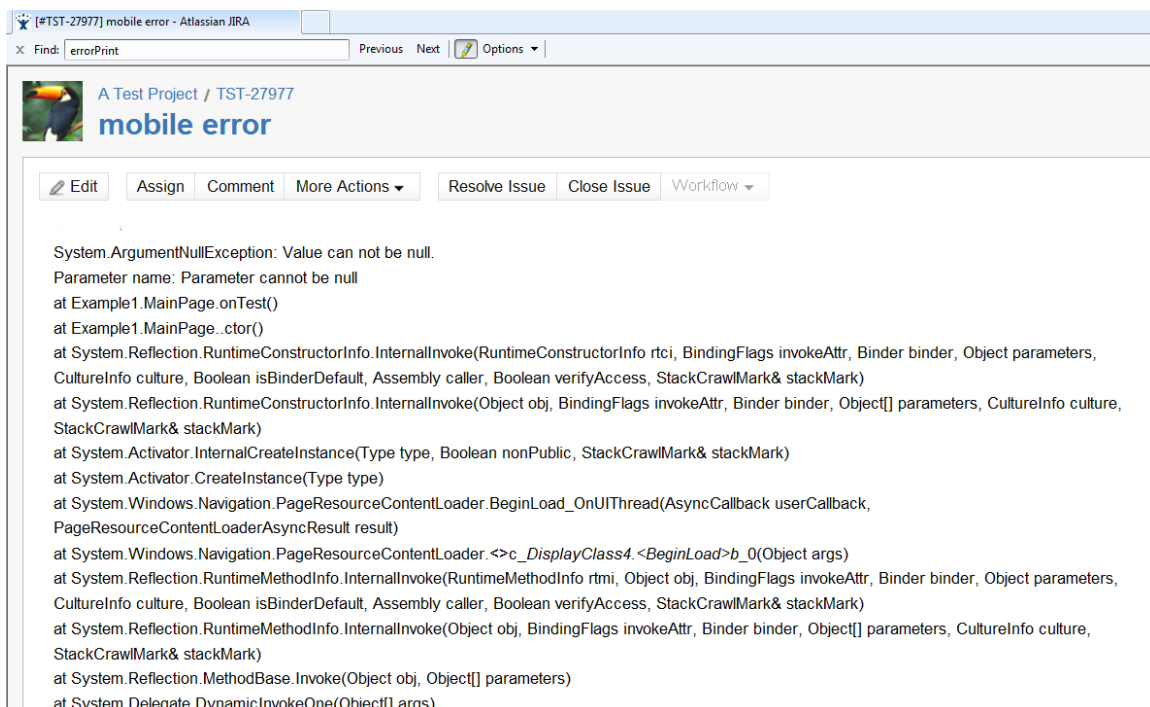
Currently JIRA Project Management by Atlassian and Email are support. We are actively working with project management companies to build connectors. Contact Robillion support to get your mechanism included.

```

public void onTest()
{
    try
    {
        throw new ArgumentNullException("Parameter cannot be null");
    }
    catch (Exception e)
    {
        RobillionAgent robillion = RobillionAgent.GetInstance;
        robillion.onBug("TST",
            "1", "mobile error",
            e.ToString());
    }
}

```

Adding the onBug method from Robillion sends the error automatically to your project management system,



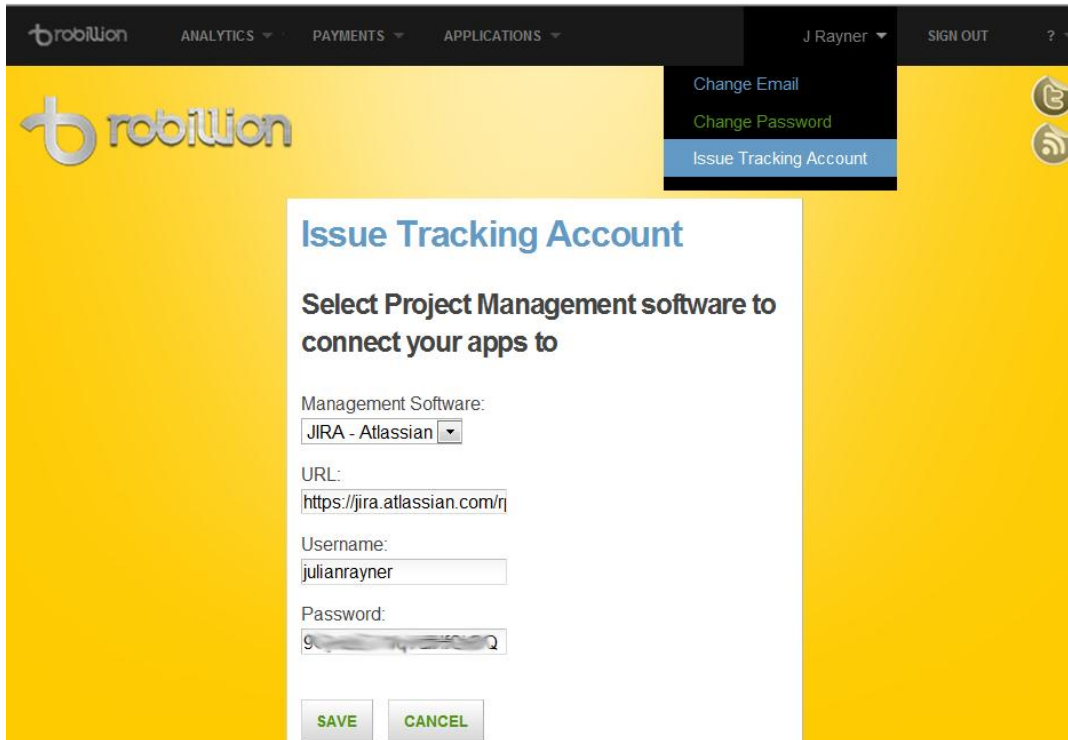
The screenshot shows a JIRA issue page for "mobile error" in the project "A Test Project / TST-27977". The issue is currently open. The stack trace is as follows:

```

System.ArgumentNullException: Value can not be null.
Parameter name: Parameter cannot be null
at Example1.MainPage.onTest()
at Example1.MainPage..ctor()
at System.Reflection.RuntimeConstructorInfo.InternalInvoke(RuntimeConstructorInfo rcti, BindingFlags invokeAttr, Binder binder, Object parameters, CultureInfo culture, Boolean isBinderDefault, Assembly caller, Boolean verifyAccess, StackCrawlMark& stackMark)
at System.Reflection.RuntimeConstructorInfo.InternalInvoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture, StackCrawlMark& stackMark)
at System.Activator.InternalCreateInstance(Type type, Boolean nonPublic, StackCrawlMark& stackMark)
at System.Activator.CreateInstance(Type type)
at System.Windows.Navigation.PageResourceContentLoader.BeginLoad_OnUIThread(AsyncCallback userCallback, PageResourceContentLoaderAsyncResult result)
at System.Windows.Navigation.PageResourceContentLoader.<>c__DisplayClass4.<BeginLoad>b_0(Object args)
at System.Reflection.RuntimeMethodInfo.InternalInvoke(RuntimeMethodInfo rmi, Object obj, BindingFlags invokeAttr, Binder binder, Object parameters, CultureInfo culture, Boolean isBinderDefault, Assembly caller, Boolean verifyAccess, StackCrawlMark& stackMark)
at System.Reflection.RuntimeMethodInfo.InternalInvoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture, StackCrawlMark& stackMark)
at System.Reflection.MethodBase.Invoke(Object obj, Object[] parameters)
at System.Delegate.DynamicInvokeOne(Object[] args)

```

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 application 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". This modal window prompts the user to "Select Project Management software to connect your apps to". It contains the following fields: "Management Software:" with a dropdown menu set to "JIRA - Atlassian"; "URL:" with a text input field containing "https://jira.atlassian.com/r"; "Username:" with a text input field containing "julianrayner"; and "Password:" with a masked text input field. At the bottom of the modal, there are "SAVE" and "CANCEL" buttons.