PTC®

PTC® IoT Academic Program

Getting Started with Android Smartphones and ThingWorx

How to send and visualize data from your Android phone using ThingWorx

Revision #	Date	ThingWorx Revision	Changes	Owner
1.0	18-05-15	5.1.0.433		Veronica MIHAI
2.0	10.05.15		Updated based on Andy's comments	Veronica MIHAI
3.0	12.05.15		Updated based on Andy's comments	Veronica MIHAI
4.0	16.05.15		Added two exercises and DeAnna's exercise	Veronica MIHAI
5.0	24.05.15		Made small changes to text	Veronica MIHAI
6.0	14.07.15		Updated the alerts and subscription part	Veronica MIHAI
7.0 - Current	16.07.15		Add part about connecting multiple devices	Veronica MIHAI

In this tutorial you will use the sensors on your Android phone, send data to your ThingWorx hosted instance and then display the data in a pre-built ThingWorx Mashup. The purpose of this project is to let you quickly connect a sensor or set of sensors to a ThingWorx Mash Up and then display that data. This project also removes the need to deal will electronics or programing. Later you can make modifications such as adding additional sensors, modifying the mash up, creating an alert or using the phone and its sensors to make something smart and connected.

This tutorial involves five steps. Three are for set up.

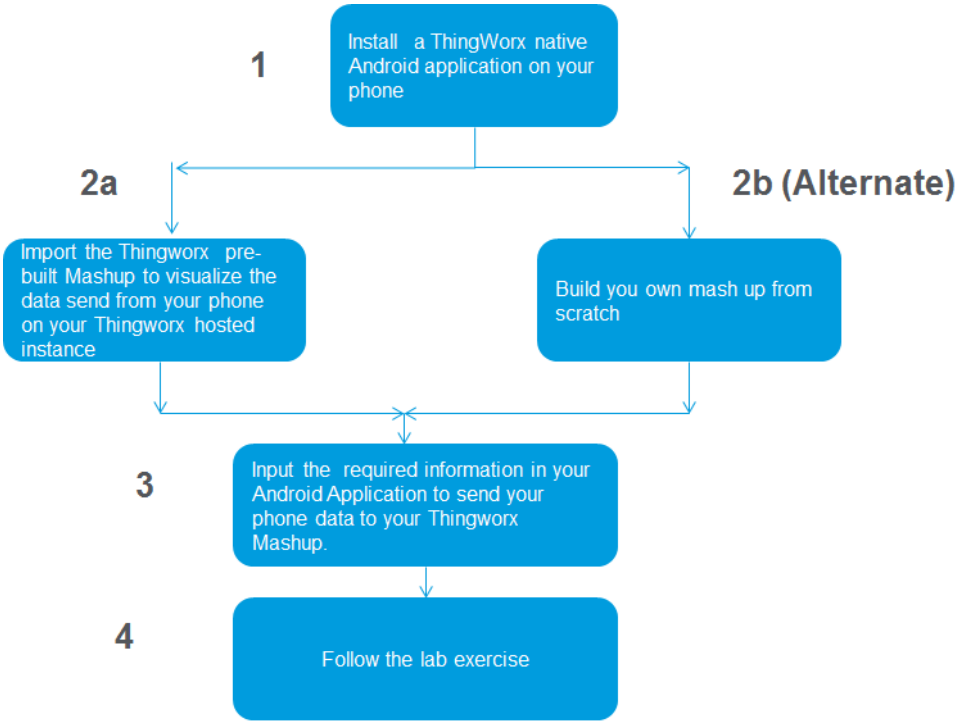
1. Install the ThingWorx native Android application on your phone. The application was written using the Java Android SDK and you can view the code **here**[1]. This application sends data about your Phone's Type, IMEI[2], GPS Location, CPU Usage, Accelerometer data (whether the phone was shaken or not), Available RAM and Total RAM to your ThingWorx hosted instance.
2. Import the ThingWorx pre-built Mash Up to your hosted instance. The Mash Up allows you to visualize the data sent from your phone.

[1] This is an Android Studio Project. To import it on your Android Studio IDE, first extract the project from the archive and aterwards on your Android Studio IDE go to File → Import Project. Locate your project location and click OK.

[2] IMEI stands for International Mobile Equipment Identity number and it is the unique ID number assigned to nearly every modern mobile phone sold.The IMEI is a 14-digit string that includes information on the origin, model, and serial number of the mobile device

3. Input some information to the Android Application on your phone so that it can send your phone's sensor data to the ThingWorx Mashup on your hosted instance.
4. Use the data that you collect to bring value. In the first exercise we will determine the quality of a bike ride.
5. Modify the Mash Up so that you learn a little about ThingWorx. In the second and third exercise you will add an alert when a threshold is reached and a Google map to visualize the location of the phone.

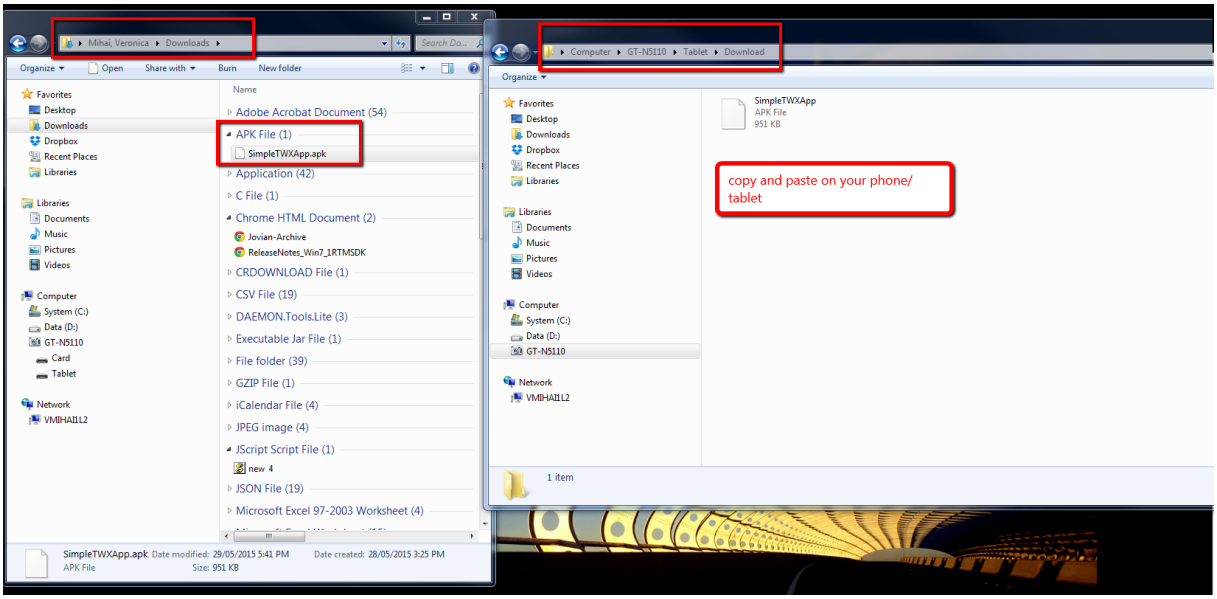
STEP SUMMARY



STEP 1

To install the ThingWorx Android App on your phone you can download and install the file directly on your phone/tablet by visiting this **link**. The downloaded files will appear in your Downloads folder. Go to that location and click the downloaded „ GettingStartedWithThingWorx_10_07_15.apk" file.

Another way to do this is to download it to your computer and then transfer it to your phone (also known as ‚sideloading', i.e. installing applications from sources other than Google Store). If you first download the file to your computer you need to transfer it to your phone by connecting your phone to your PC, using your phone's data cable (usually a micro-USB cable). This will make your phone appear as a Removable Storage device in the My Computer window. Click on this folder on your Computer and then copy and paste the .apk file on your phone to a location of your choice.

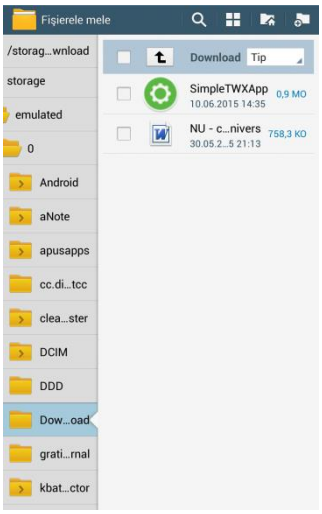


You may need to modify your settings to allow app installations from unknown sources. Here is a detailed tutorial on how this can be done.

To install the application on your phone, follow these steps:

- Disconnect your phone from your computer
- On your phone go to the location to which you copied and pasted the .apk file that you previously uploaded

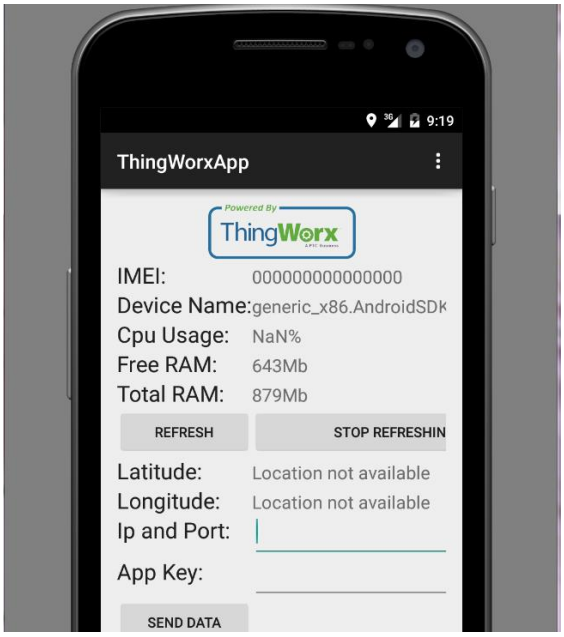
-



Click it. You will be asked to allow permissions for installation. Choose install.

- After installation you will find the Simple ThingWorx App in all applications on your phone. Run the application.

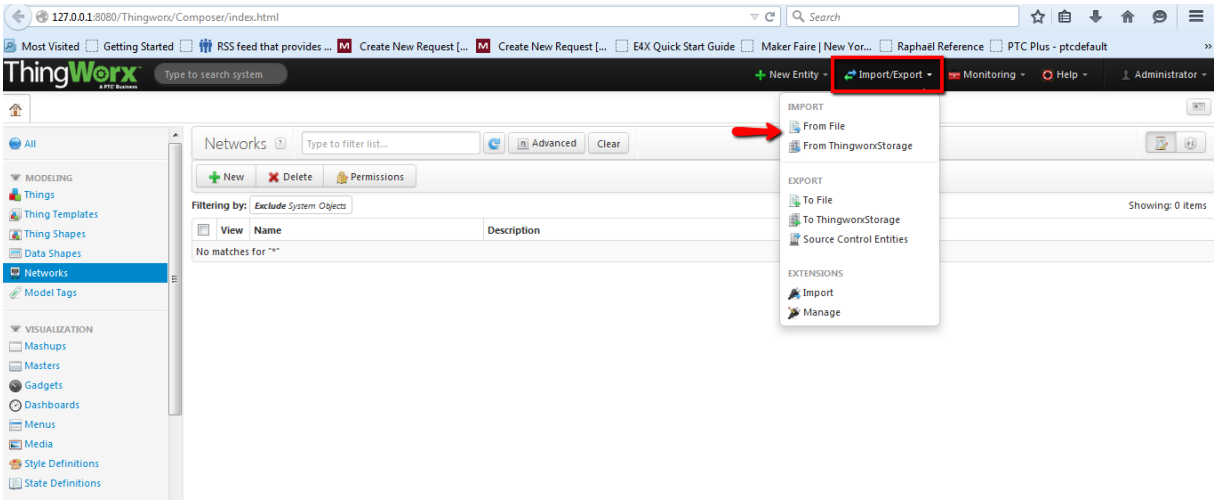
Open it to visualize the user interface. The application should look like the following;



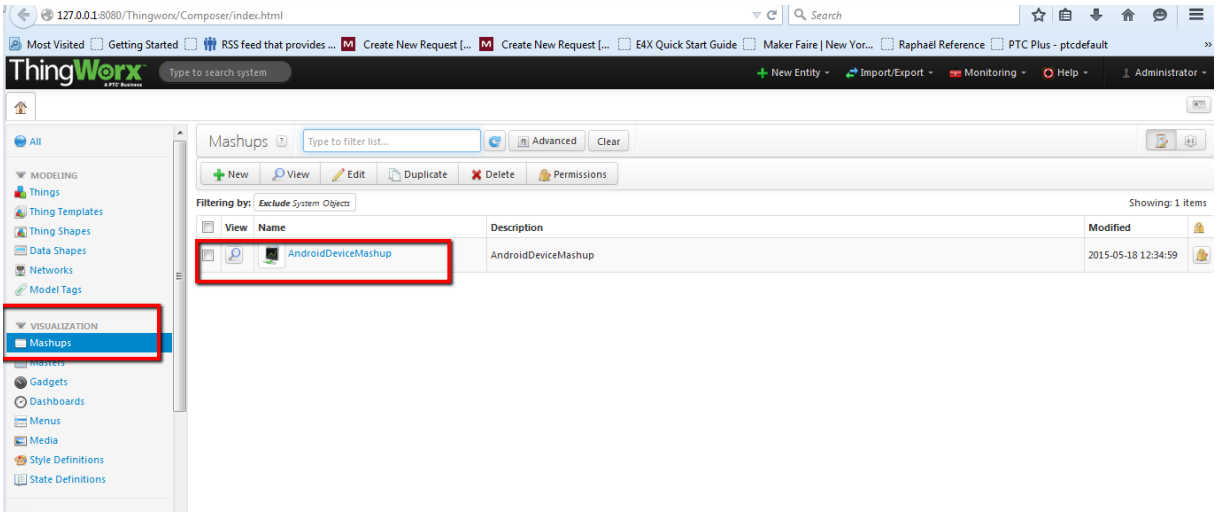
STEP 2

Go back to your computer's web browser, and download the Entities for the pre-built Android Mashup located **here**. The files will appear in your computer's downloads folder. Unzip the archive.

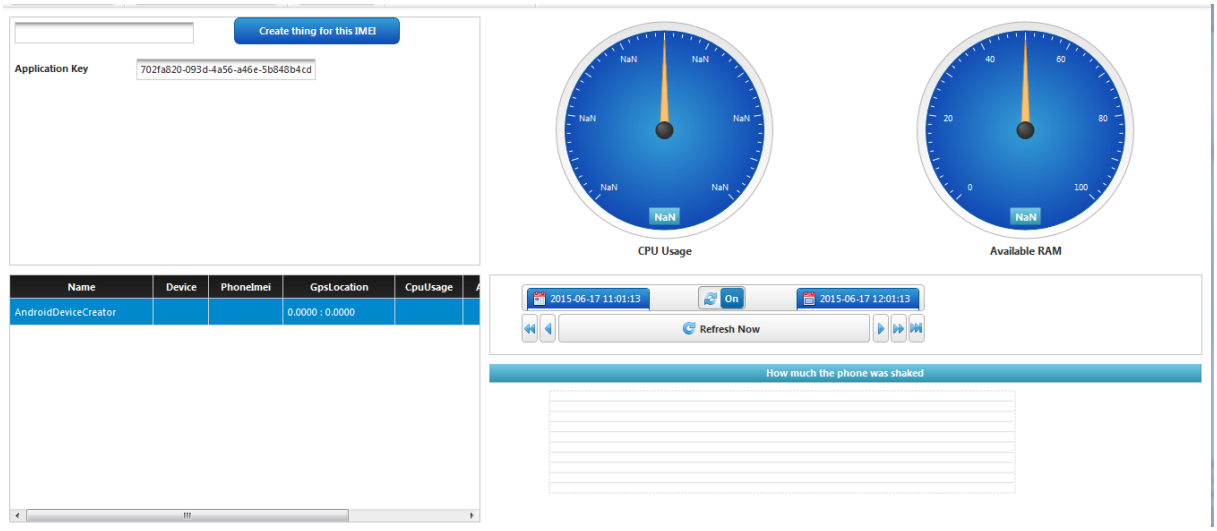
Next go to your ThingWorx hosted instance Composer and click the Import/Export Dropdown.



Select Import from file. Choose the Android SimpleTWXAndroidApp_Entities.xml file that you previously downloaded. The pre-built Mashup is now available under Mashups. You will need to have administrator rights to be able to import the files (If you don't have admin rights, you can ask your professor to do this import. The import only needs to be done once)

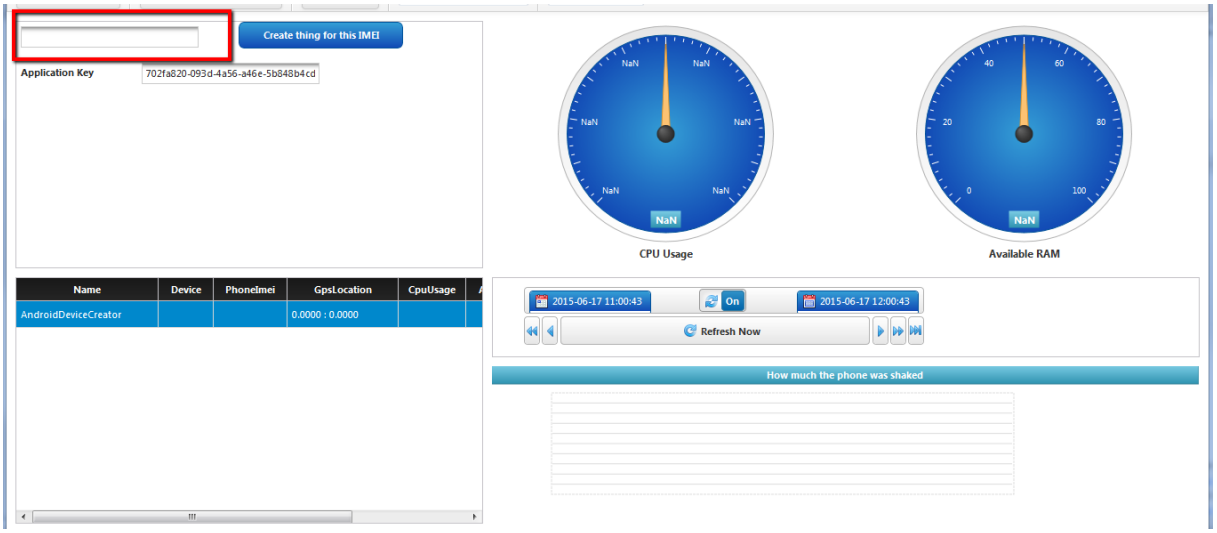


Open it and click View Mashup to visualize it.

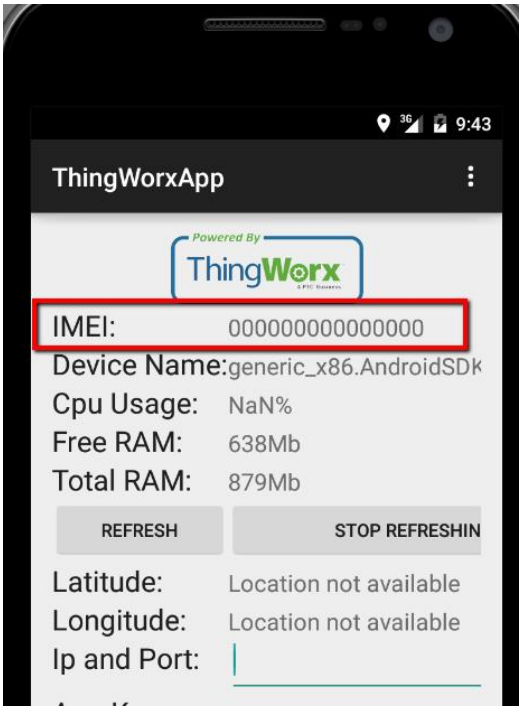


STEP 3

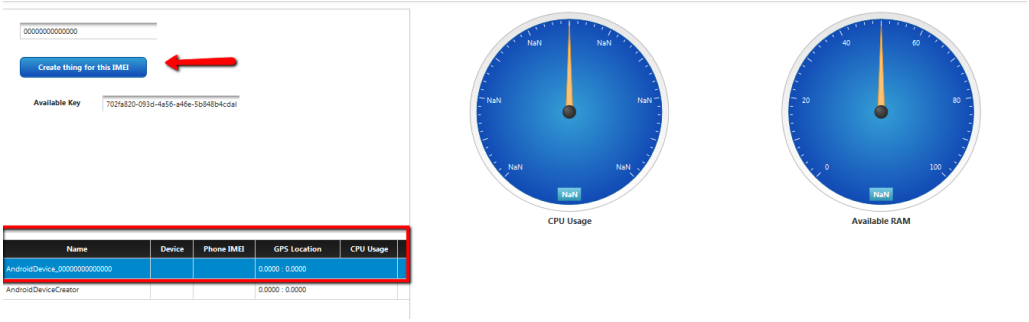
- Enter your phone's IMEI in the input field of your Mashup.



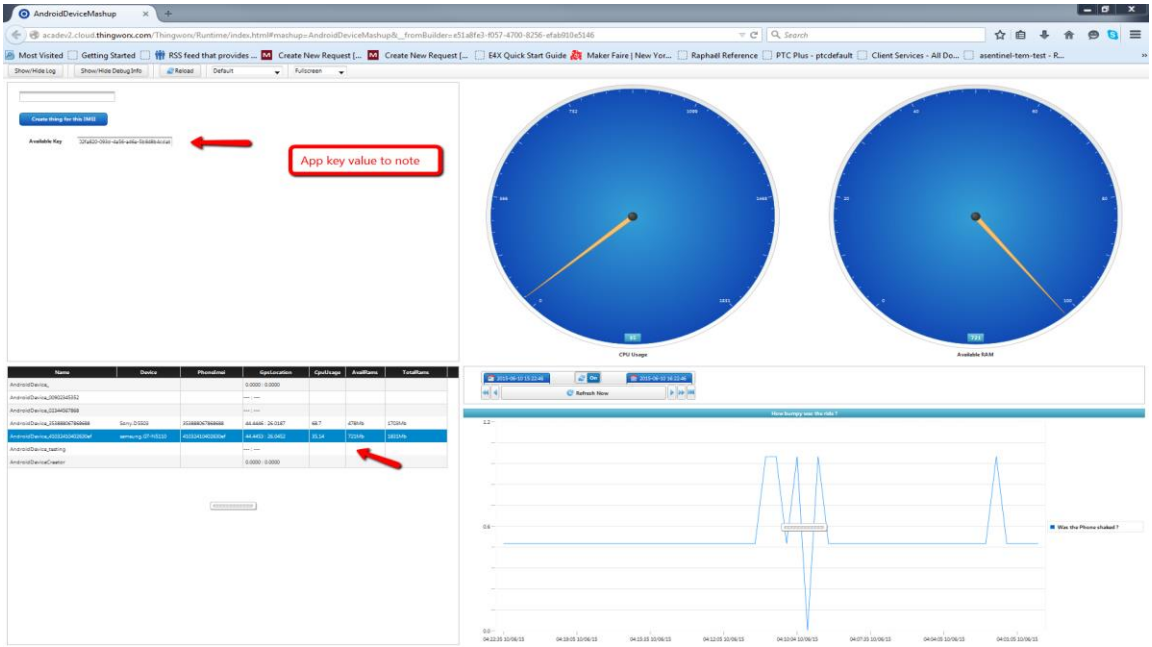
To locate your phone's IMEI, go to the ThingWorx Application on your phone. It's the first field from the user interface.



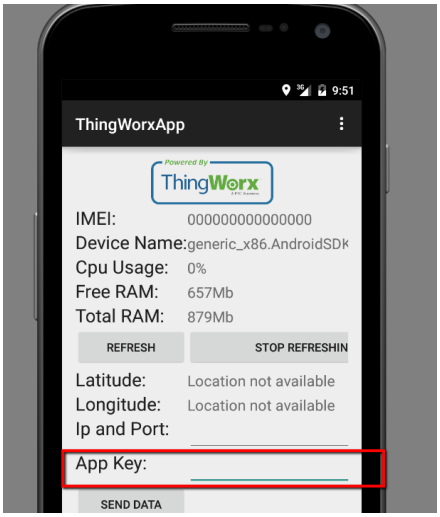
- Next, click the Create thing for this IMEI button that is pointed to in the picture below. This creates a „Thing" on the ThingWorx platform that will model the phone from which the data is sent to ThingWorx. The binding of data is done based on your phone's IMEI value. The IP address that you fill in on your phone app tells the phone where to send data to. The creation of your Phone Model Thing will be reflected in the grid at the bottom of the mashup.



All the information concerning your device is currently empty. To populate these fields with data, take note of the Application Key shown in your Mashup Application Key field on a piece of paper.

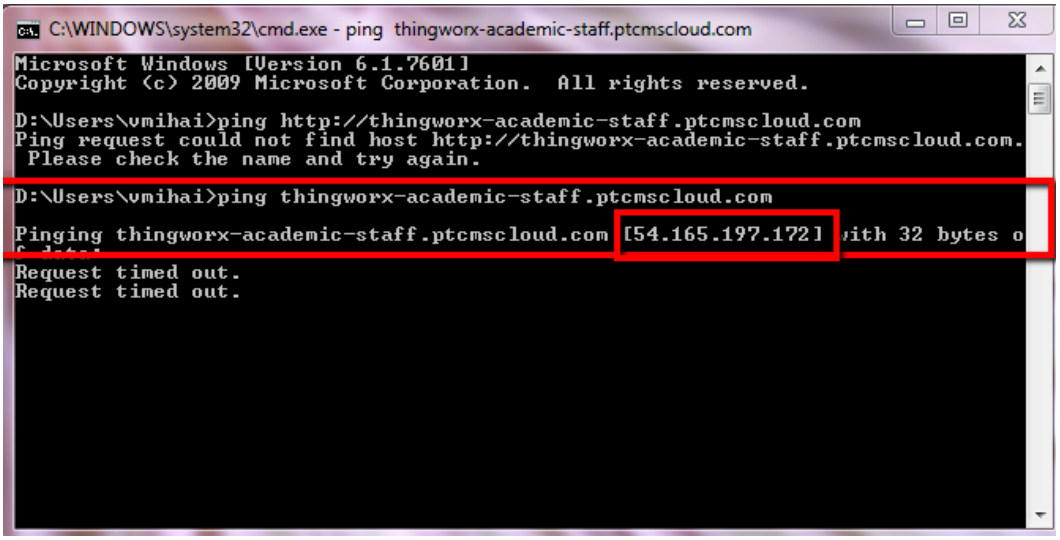


Next, go back to the ThingWorx App on your phone and enter the value for this app key in the corresponding field. The Application key is a security code that lets your phone talk to the ThingWorx Mash Up. To make this process easier, you could copy and paste the App Key Value from the ThingWorx Mashup into an email and send it to yourself so that you can copy and paste it in your Android App.



You will also need to enter your ThingWorx hosted Instance IP address on which it is listening for incoming data. Put it in this format x.x.x.x . Replace x with the values for your instance IP.

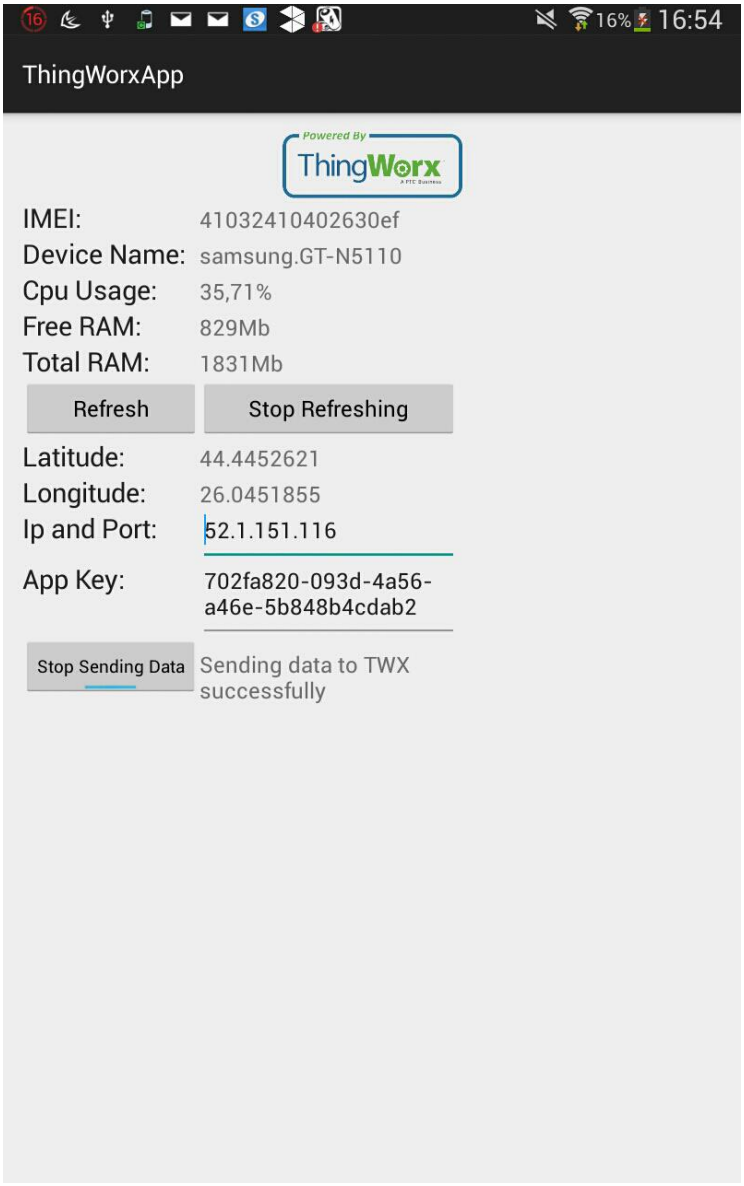
To find your hosted instance IP, go back to the computer you use to access your hosted instance, open a command prompt and ping the name of your ThingWorx instance. The IP address will be shown next to the instance name. Ignore the Request timed out reply for your ping. These may occur due to network restrictions on your network side or on the hosted instance side.

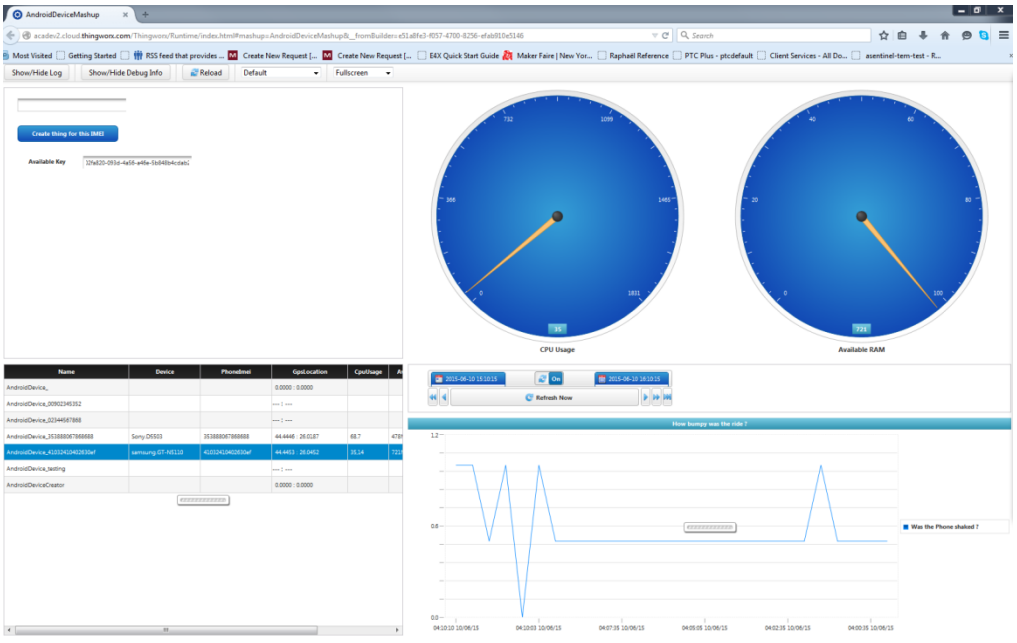


Now, all that is left to do is to press the SEND DATA Button on your TWX Mobile App and view the changes in your ThingWorx Mashup. This will send data to ThingWorx once every 3 seconds. To stop sending data to ThingWorx press the Stop Sending Data button on your Android App. The application will also stop sending data when you close it on your phone.

If you want to connect multiple phones to this Mashup, simply repeat the process described above :

1. Get your phone's IMEI from your Android App and create a Thing with this IMEI in the pre-built Mashup.
2. Enter the App Key and your ThingWorx instance IP on your Android Phone App.
3. Press send data to populate the fields in your Mashup with data.


16%
16:54
ThingWorxApp
Powered By
ThingWorx
IMEI:
41032410402630ef
Device Name:
samsung.GT-N5110
Cpu Usage:
35,71%
Free RAM:
829Mb
Total RAM:
1831Mb
Refresh
Stop Refreshing
Latitude:
44.4452621
Longitude:
26.0451855
Ip and Port:
52.1.151.116
App Key:
702fa820-093d-4a56-a46e-5b848b4cdab2
Stop Sending Data
Sending data to TWX successfully



The values plotted on the Chart range from -1 to 1 and are directly proportional to the intensity with which your phone was shaken. The value -1 means that your phone is currently not connected, the value 0 is used to indicate that no shaking of the phone is taking place and 1 is the maximum value for the intensity with which your phone was shaken. Other intermediate values are 0.5, 0.7, and 0.9. These are the only values that the threshold takes.

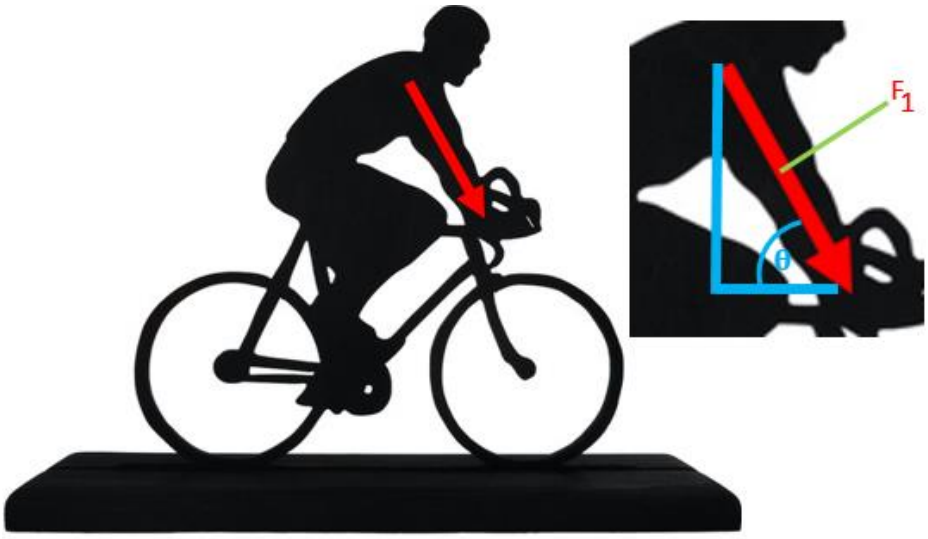
STEP 4

Determining the Quality of a Bicycle Ride:

Exercise 1

Problem:

We know that riding a bike on rough terrain can cause vibrations which result in bodily pain for the rider. In the overview, three contact points were discussed: hands, feet and buttocks. One of the more common problems that riders have is incorrect seat position. Seat position is directly related to the amount of pressure transferred from the hands onto the handle bars. See the image below:



$$F_1 = m_{rider} * g * \sin(\theta)$$

The force from the riders arms onto the handlebar increases with increasing θ.

Testing Procedure:

First, move the handle bars 1 inch forward, this will decrease θ which will also decrease the force which the arms and hands have to take on.

Second, use an arm strap like the one below to secure your cell phone on one of your arms (aka the contact point). Ride your route as usual while collecting data in ThingWorx. Use the Analytical questions to achieve desired results.

Analytical Questions:

1. What threshold for vibrations is comfortable for your arms? Ride the same route several times with increasing handle bar position (1 inch increments) until a desired level of comfort is reached.
2. How will you alert yourself that a route is uncomfortable?

STEP 5

Adding an Alert to your ThingWorx Application

Exercise2

In this exercise you will learn how to set an alert in ThingWorx which will send you an email when a vibration threshold has been reached.

Go to the AndroidDeviceTemplate and open the Subscriptions section. Press the pencil icon on subscription for the ShakeTreshold Source Property.

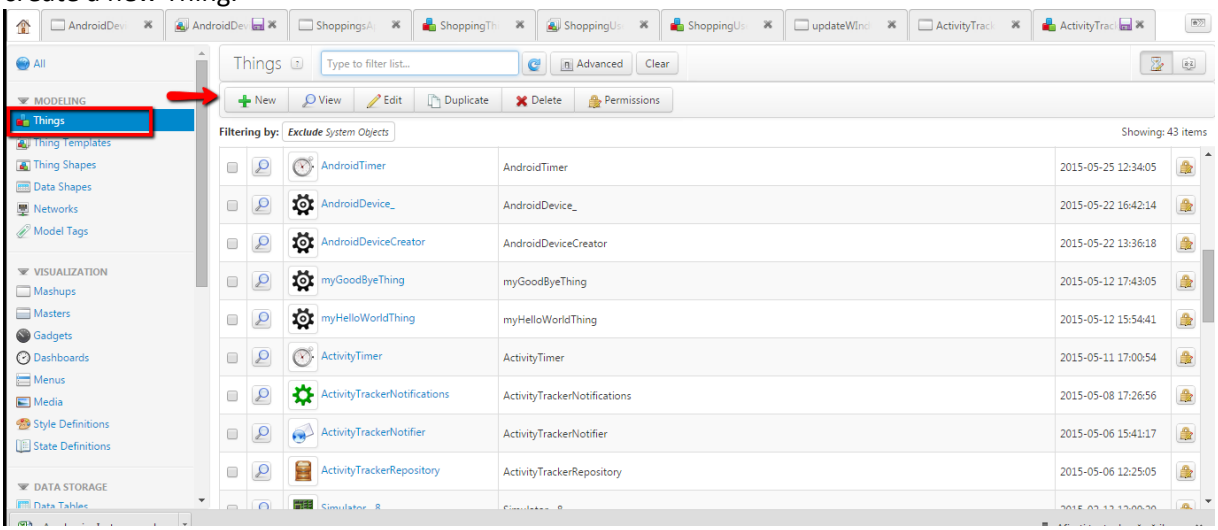
This subscription adds the value of the ShakeTreshold property in a Stream each time the value of this property changes. By storing the data in a stream you can have a history of when and how much your phone was shaken; you can later choose to plot it on a graph.

To send yourself an email when a certain value for the threshold is reached you need add a piece of code in the second subscription of the AndroidDeviceTemplate.

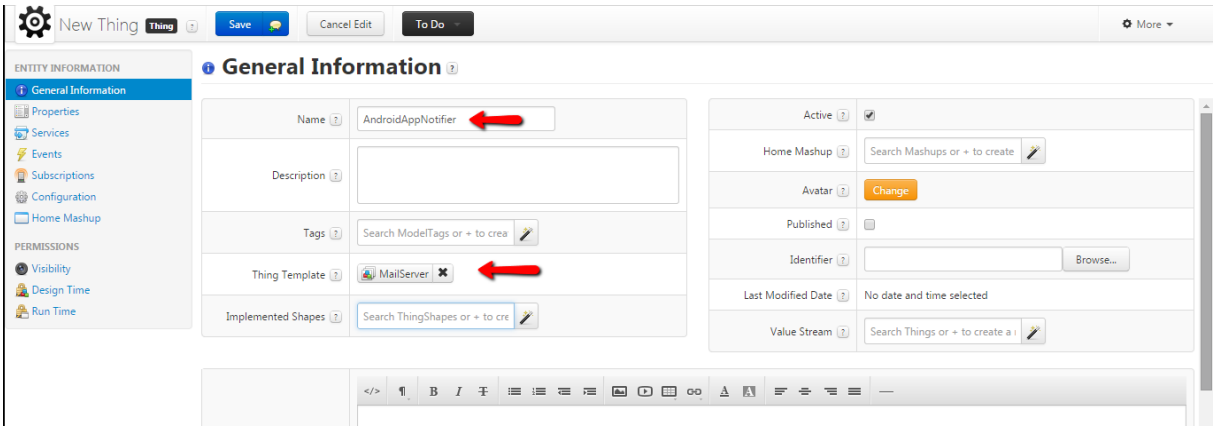
But before doing this, you need to have an Email Notifier Thing based on a MailServer Template. This Thing based on the MailServer Template will give you access to a SendMessage service that you will be able to use in the subscription to your ShakeThreshold alert. See glossary of terms to learn more about Things and Templates.

Follow the next steps to create and configure one on your hosted instance:

- Go to the Things section on your ThingWorx Composer and press the green plus button to create a new Thing.

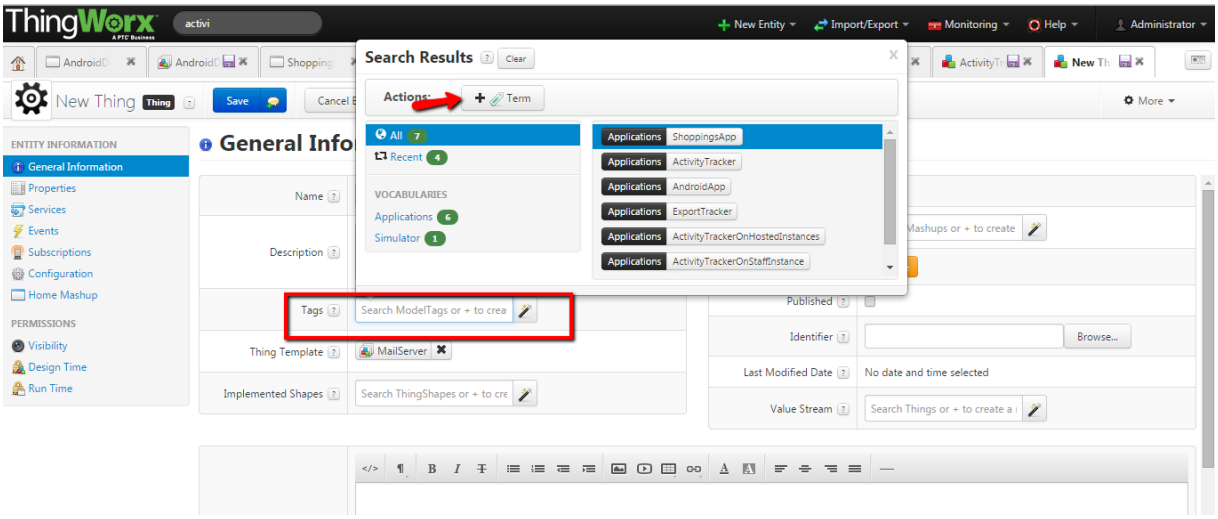


- Name it AndroidAppNotifier and select MailServer [3]as Template.

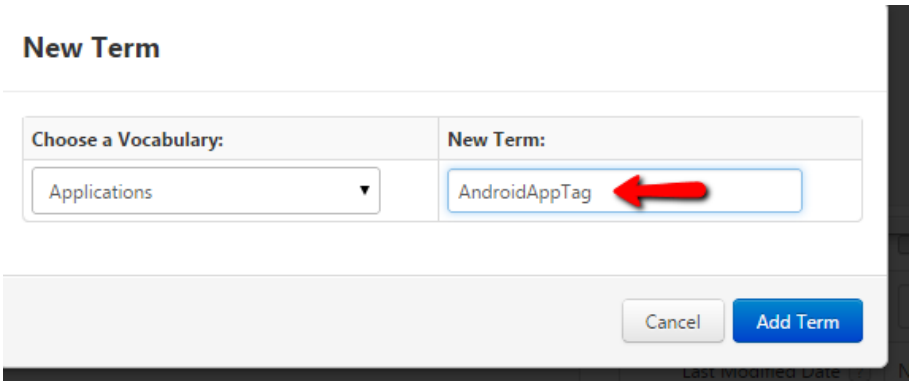


- You can also choose to tag it if you plan on exporting it later. If you don't have a tag already created, you can create one by pressing the wand on the Tags field and clicking on +Term.

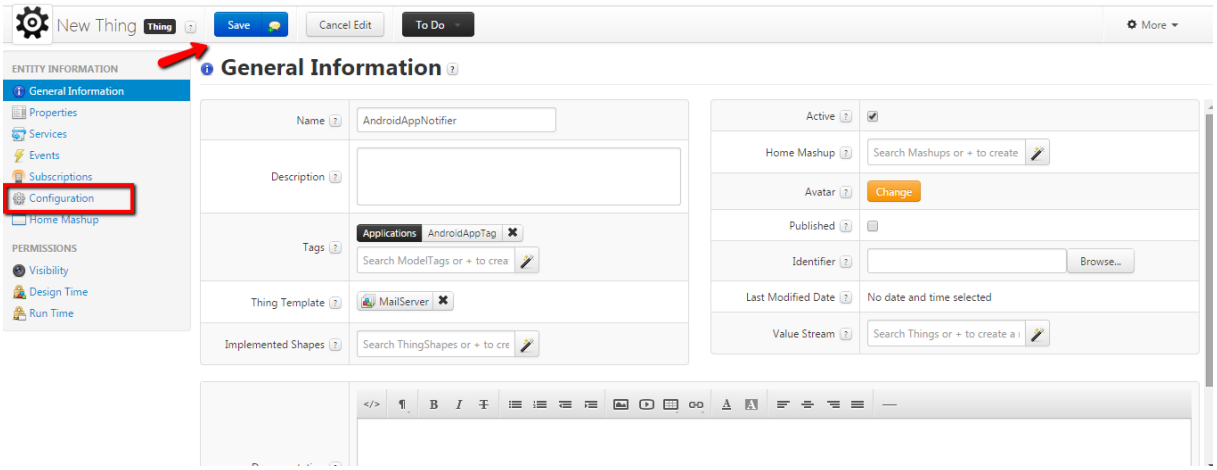
[3] If you cannot find a MailServer Template ask someone with administrator rights to import it as an extension. You can download the extension package from here. You may need to ask for permission to download them by filling in this form. To import it, go to Import/Export on your ThingWorx Composer, the Extensions Section and choose Import.



Give it a name and click Add Term.



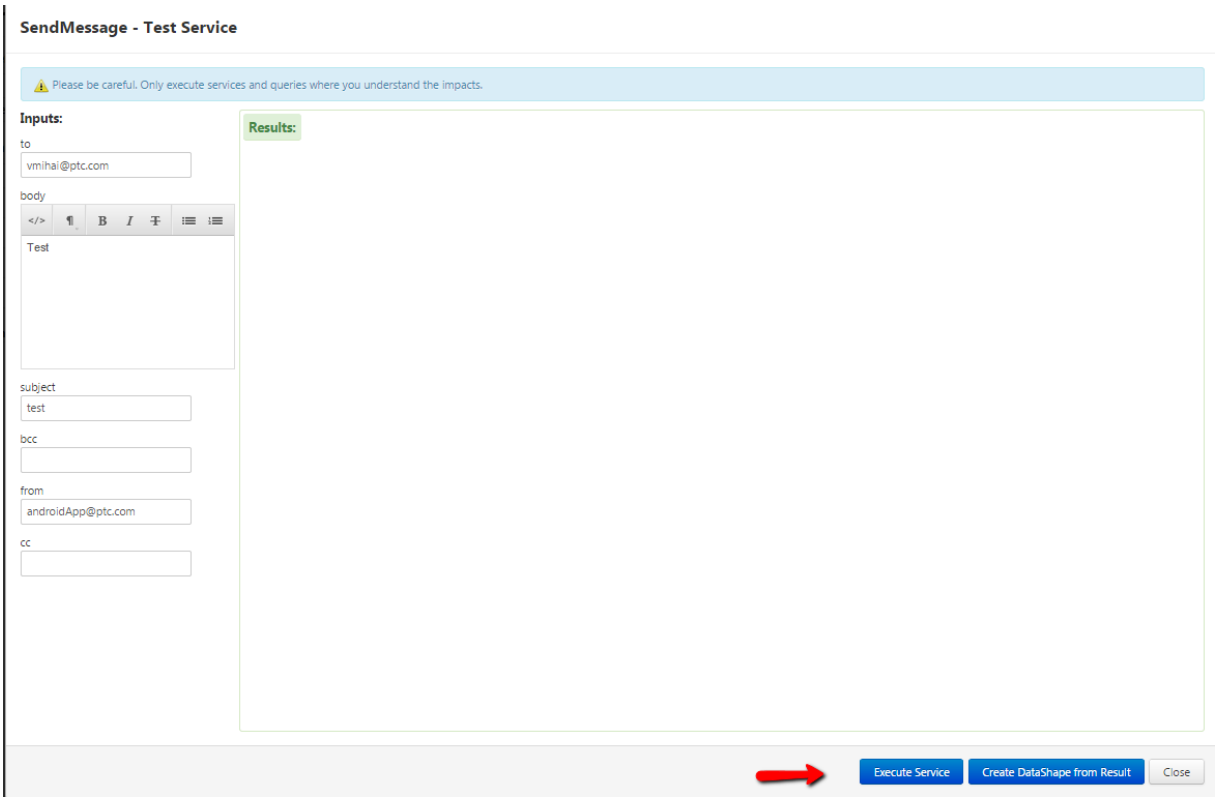
- Save your Thing and go to the Configuration Section.



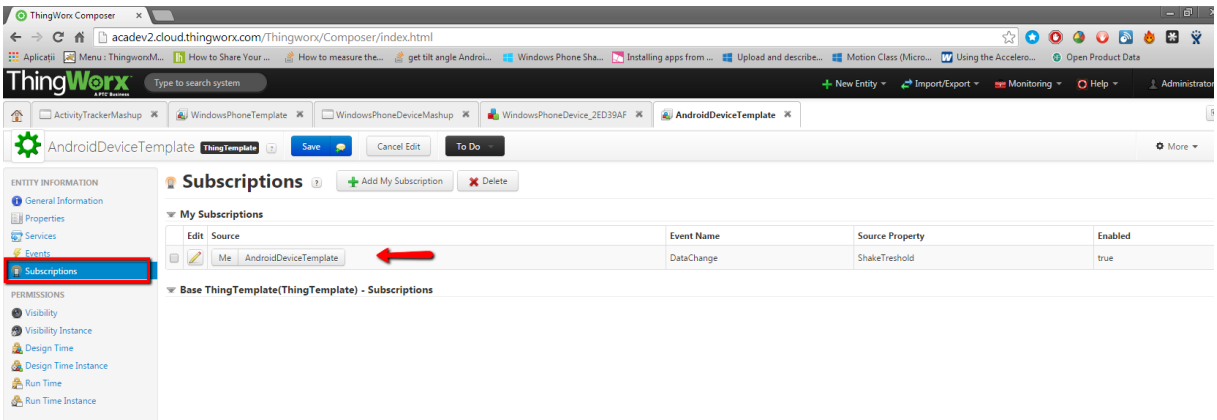
- Here type localhost as SMTP Server, 25 as SMTP Server Port, set the POP3 Server to an empty value and the POP3 Server Port to 110. Check Use TLS and set an empty value for the Mail Account User.

- Save the Thing again. You can also test it by going to Services and testing the SendMessage Service.

- You can put any address you would like in the “from” field. It doesn’t have to be an existing email address. To send the email after having completed the required field, press the Execute Service Button. If the execution was successful you will have an image like below. If you get any error check your network configuration to establish the right configuration for your MailServer.



- Now go back to the AndroidDeviceTemplate Subscriptions Section and edit the Subscription.

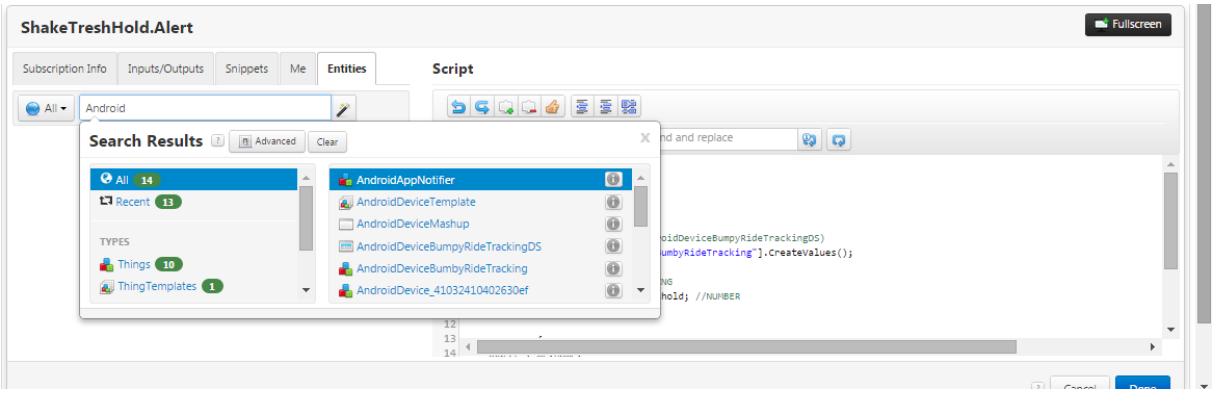


- The code in this subscription adds an Entry in a Stream each time the ShakeThreshold property value changes. Place your cursor at the top of the code and enter the following “if” block :

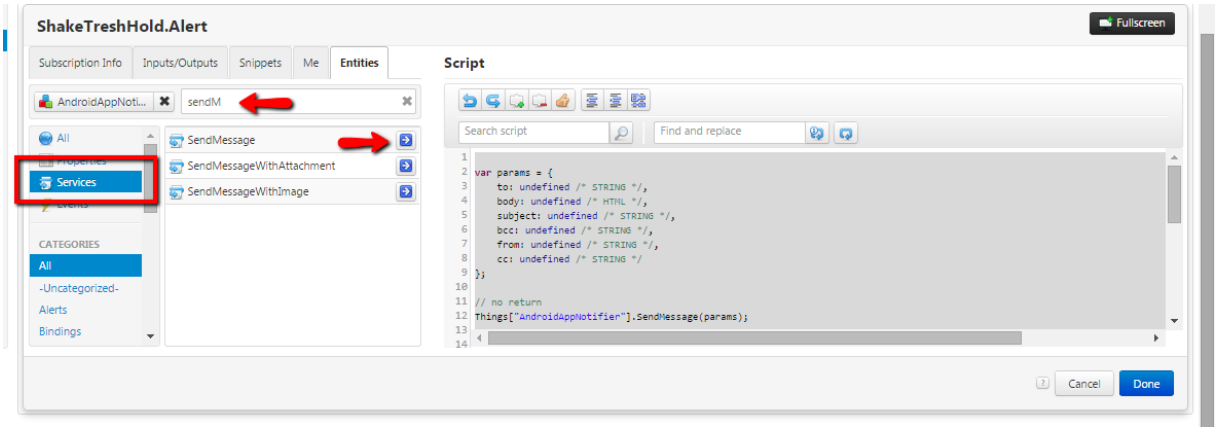
```
if(me.ShakeTreshold==0.7){

}
```

You can change the 0.7 either with 0.5, 0.9 or 1 to indicate the intensity with which your phone was shaken for which you would like to receive an email.

Next place your cursor in the body of the “if” blocks and go to the Entities tab. Here Look for your AndroidAppNotifier Thing.



Select it and go to its Services section. Here look for the SendMessage Service and click it to insert it at your cursor position in the Script Area.

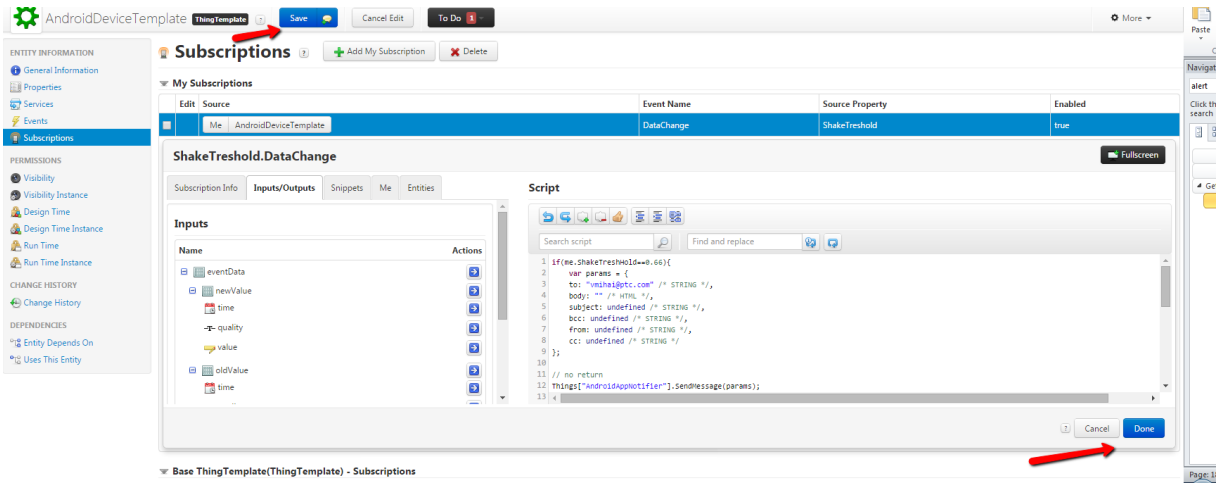


Fill in the “to”, “body”, “subject” and “from” fields with some values.

- This is the added Javascript code at the end of this exercise.

```
if(me.ShakeTreshold==0.7){
        var params = {
        to: "vmihai@ptc.com" /* STRING */,
        body: "The phone was shaken with an intensity of 0.7" /* HTML */,
        subject: “The phone was shaken” /* STRING */,
        bcc: undefined /* STRING */,
        from: “android_app_notifications@ptc.com” /* STRING */,
        cc: undefined /* STRING */
};

// no return
Things["AndroidAppNotifier"].SendMessage(params);

}
```

- Click Done and Save the AndroidDeviceTemplate.



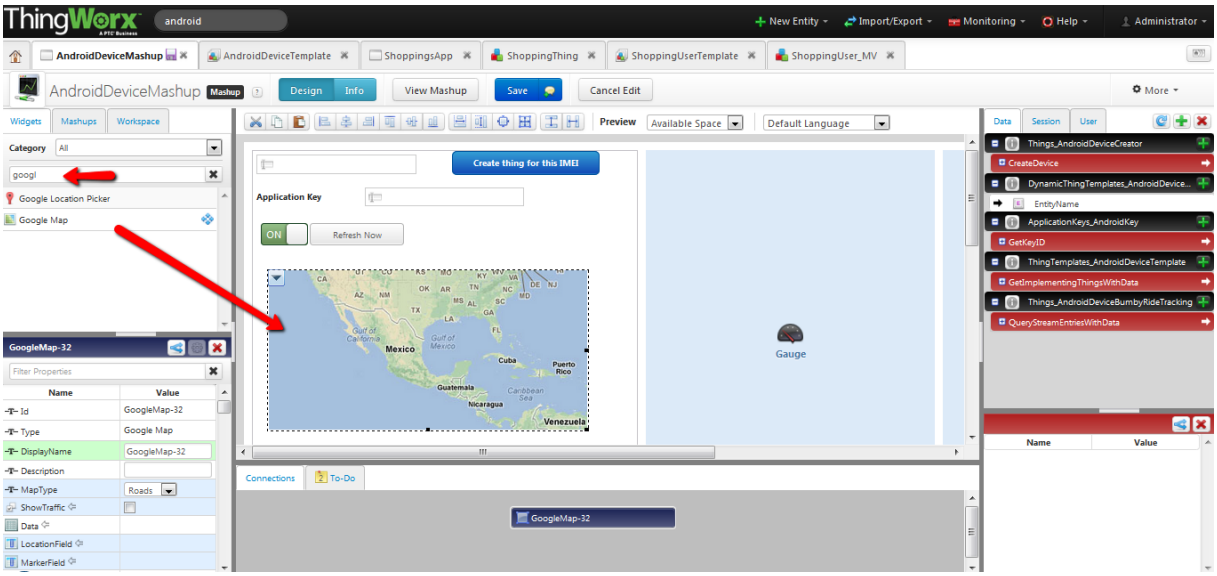
You can now go back to using your App. Now each time your phone is shaken with the intensity you choose in the "if" block you will be sent an email.

STEP 4 - Continued

Adding Google Maps to Your Mash Up:

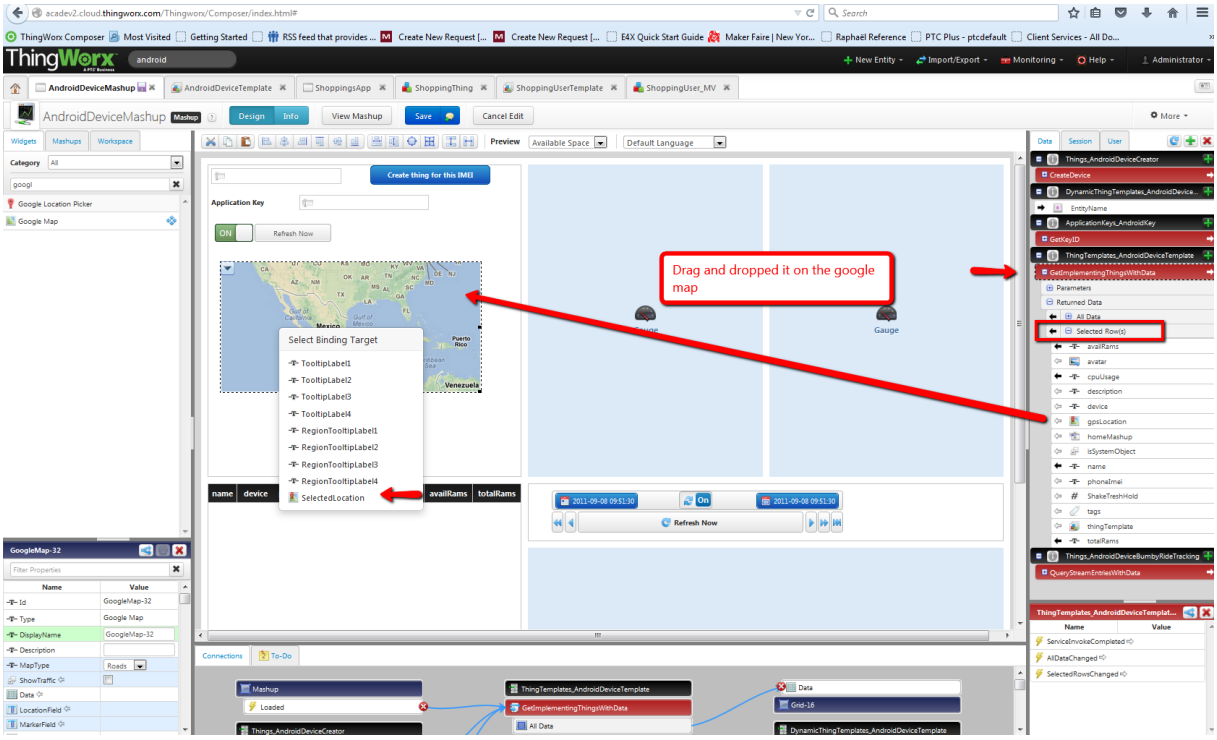
Exercise 3

To add a Google map to your Mashup, open the AndroidDeviceMashup and look for a Google Map [4]in the Widgets area. Drag and drop it below the Application Key field and Auto Refresh Widget.

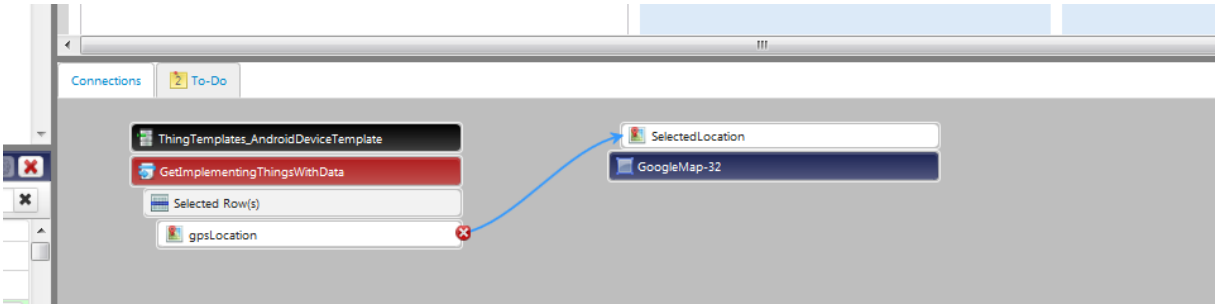


[4] If you cannot find a Google Map widget in your Widgets area ask someone with administrator rights to import it as an extension. You can download the extension package from here. You may need to ask for permission to download them by filling in this form. To import it, go to Import/Export on your ThingWorx Composer, the Extensions Section and choose Import.

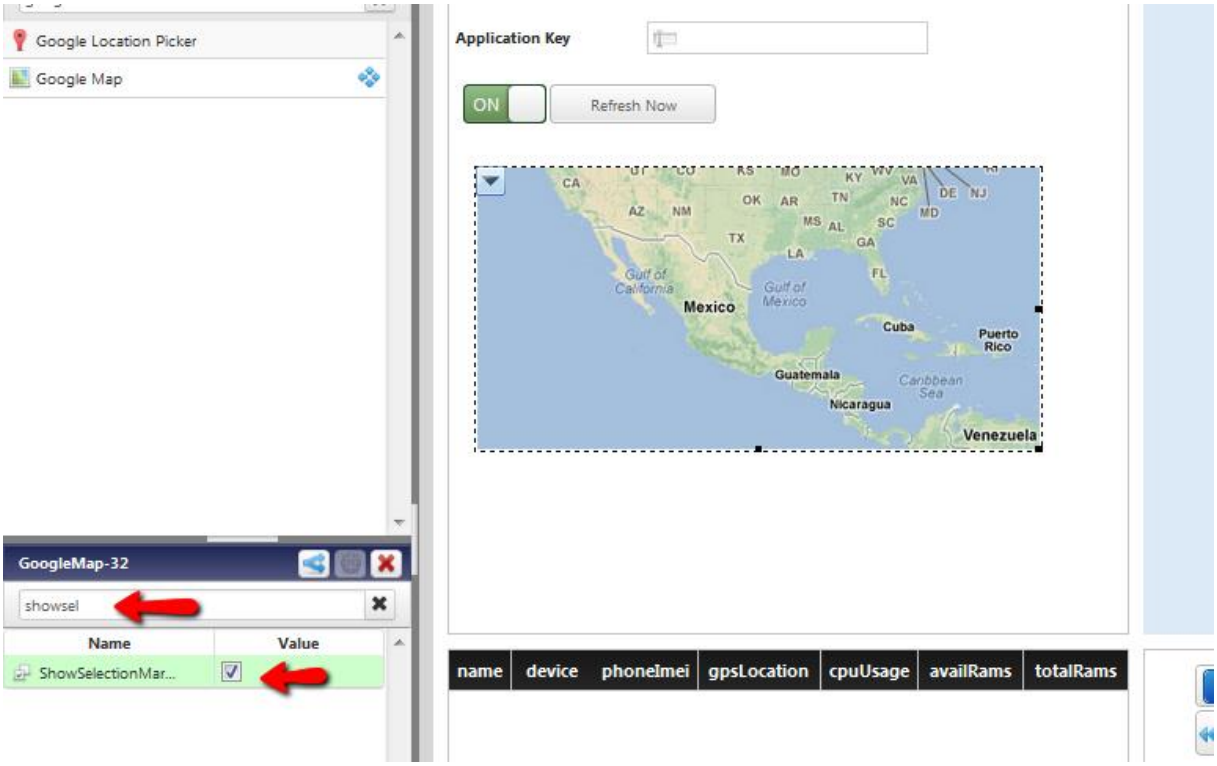
To tell the widget which location to display you need to set a value for its SelectedLocation Property. This is called a binding and it's done by dragging and dropping the value you want to set on the widget. To set the SelectedLocation Property of the GoogleMap widget to your phone's current location value, go to the service GetImplementingThingWithData of your AndroidDeviceTemplate Thing on the right side of the Mashup builder as shown below. Go to Selected Rows Section and drag-and-drop the gpsLocation property on the google map widget. Choose SelectedLocation as property for this binding.



If the binding was successful, the following connections will appear in the Connections area of the Mashup Builder, when you click on the google map.

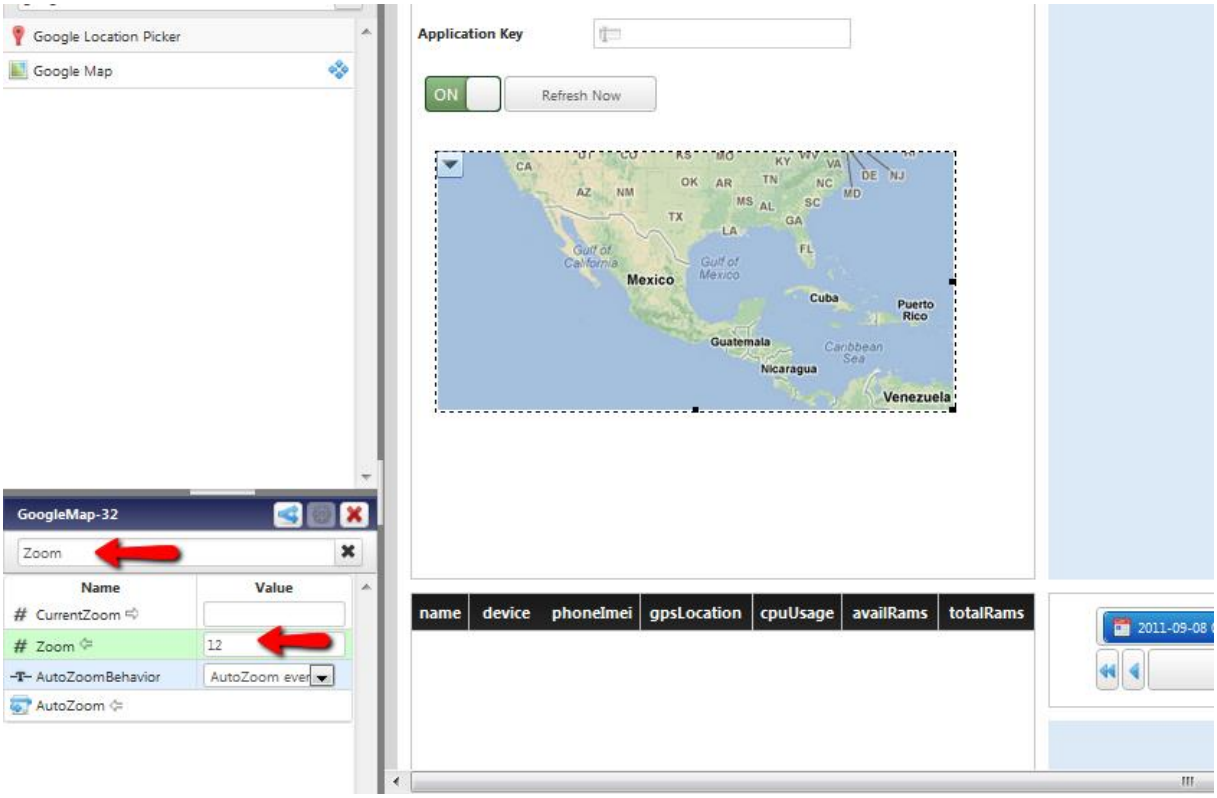


When clicking on the Google Map you can also see its properties on the bottom left side of the Mashup Builder. Type ShowSelectionMarker Property in the Properties filter and check it.

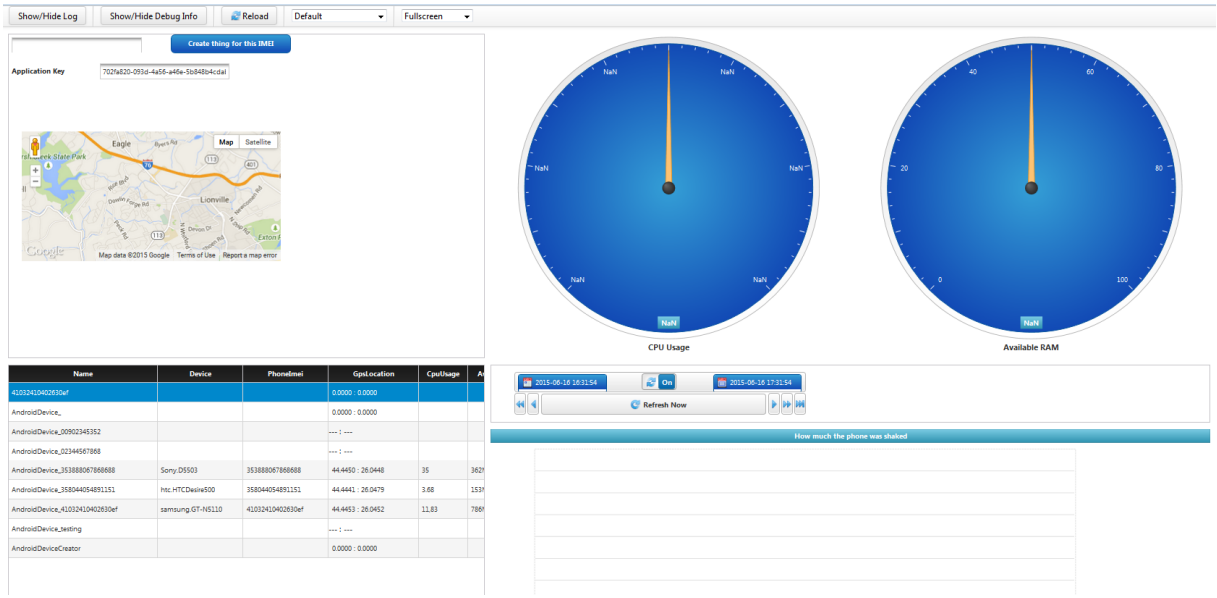


This will show a marker on the map to indicate the SelectedLocation property value.

You may also want to increase the value of the zoom on the map. You can do this by looking for the Zoom Property and setting it to 12 for example. Press enter after typing in the value for this numeric field.



Next, save the Mashup and view it.



When you select your device in the list, the map is updated to reflect the location of your phone. This is the last value for your location that your phone sent to ThingWorx.