THE CUSTOMER SUCCESS PLATFORM
SALES SERVICE MARKETING COMMUNITY ANALYTICS APPS

Build SalesForce IoT applications with ARM mbed™

Build a Salesforce IoT application with ARM mbed in less than an hour!

Doug Anson
IoT Architect – ARM
doug.anson@arm.com

Sam Grove
Senior Applications Engineer – ARM
sam.grove@arm.com

Brian Daniels
Applications Engineer – ARM
Brian.daniels@arm.com

dreamforce

Objective

ARM mbed

* Before we begin: COUNT OFF!! *And… remember your number…*
  * http://mbed.com/df2015 - go here to get this presentation and other details!

* Demonstrate delivery of devices, on constrained networks, into Salesforce.com via ARM mbed Device Server (mDS)

* Introduction to the MBED online developer environment

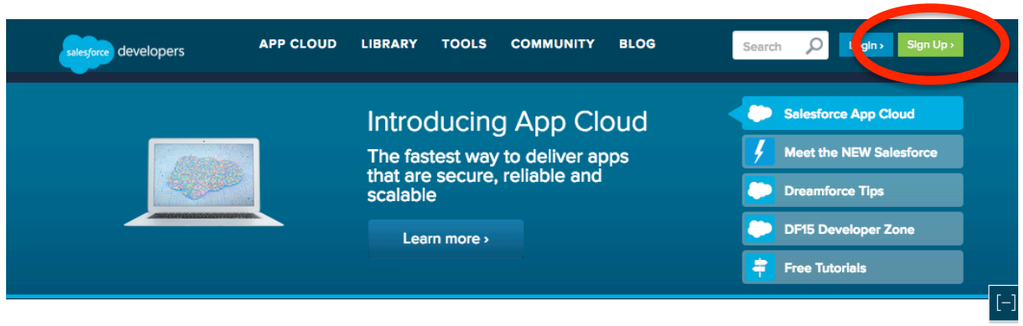
* Setup of mbed device, device connectivity, and mDS Bridge in Salesforce

* See your heart rate and location telemetry in Salesforce!

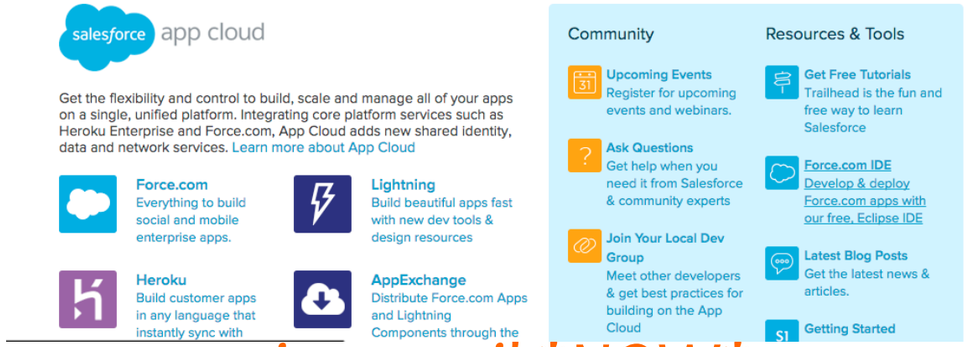
salesforce

Creating a Salesforce Developer Account

* Start Here
  * https://developer.salesforce.com



* Click on “Sign Up”



* Fill in the Form…
  * *IMPORTANT: Use an email address that you can receive email *NOW**
  * Salesforce will send you a confirmation email that must be acknowledged

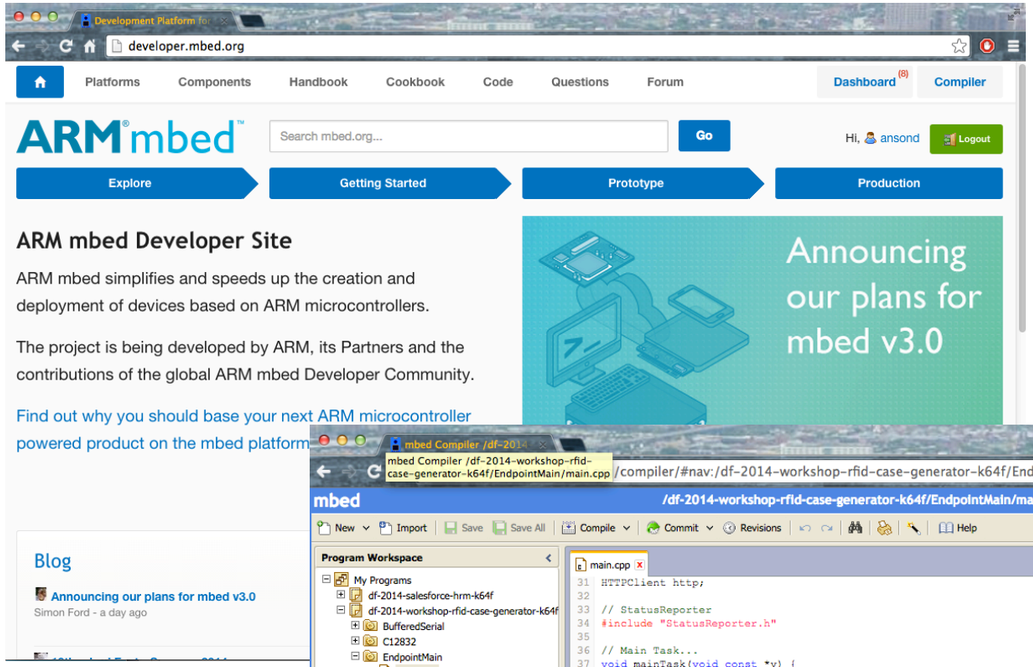
* Once acknowledged, log in here: https://login.salesforce.com
  * Click through, then Click on “Setup” in the upper right-hand side of the page

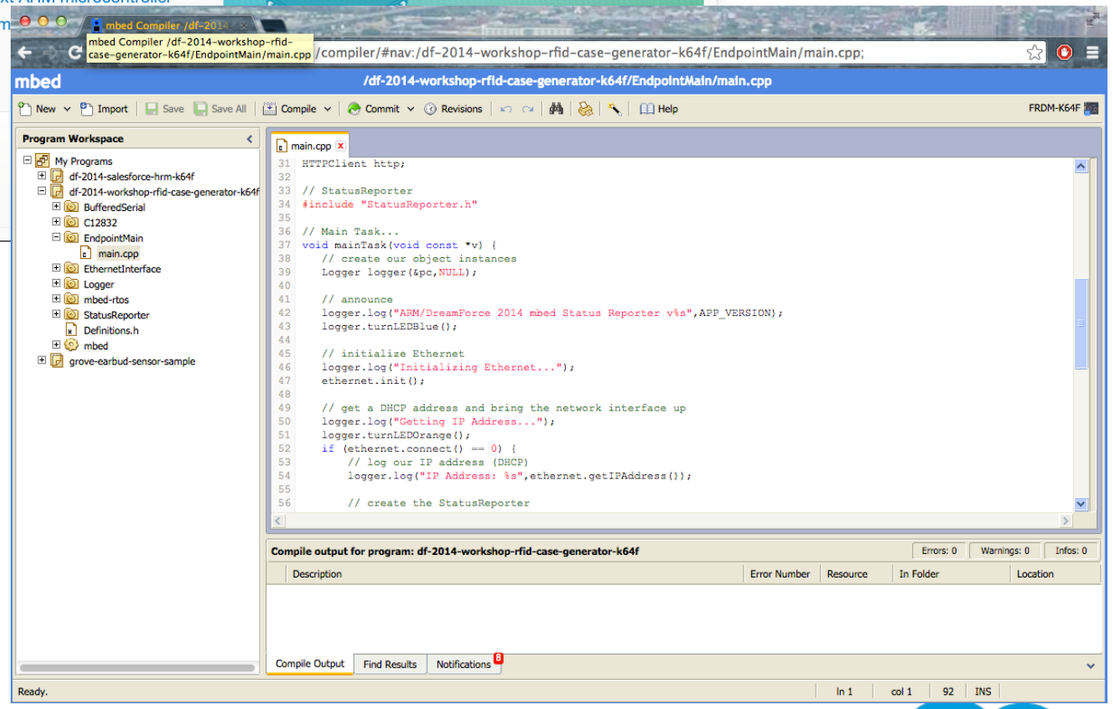
mbed Online Developer Environment

ARM mbed™

* Cloud-based
* Supports multiple compile targets (platforms)
* Full source code control
* Online forum, wiki
* **Go here now and setup your free Account for the exercise:**

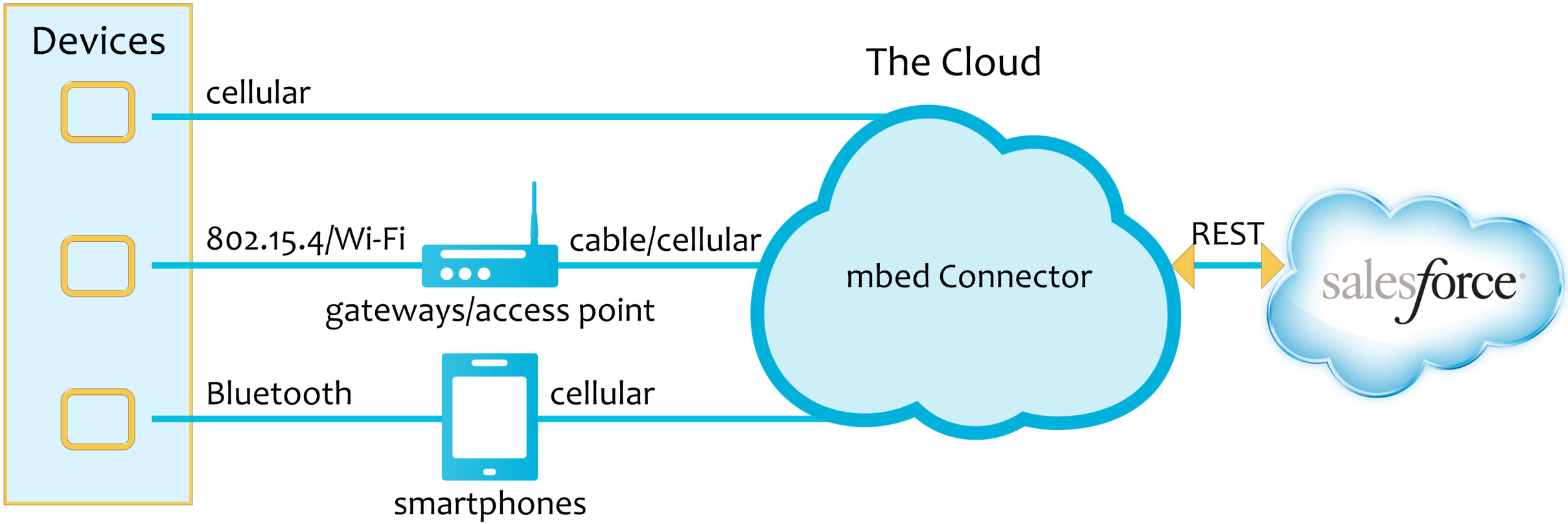
  http://developer.mbed.org





salesforce

mbed Connector: Simplify IoT Device Connection/Access

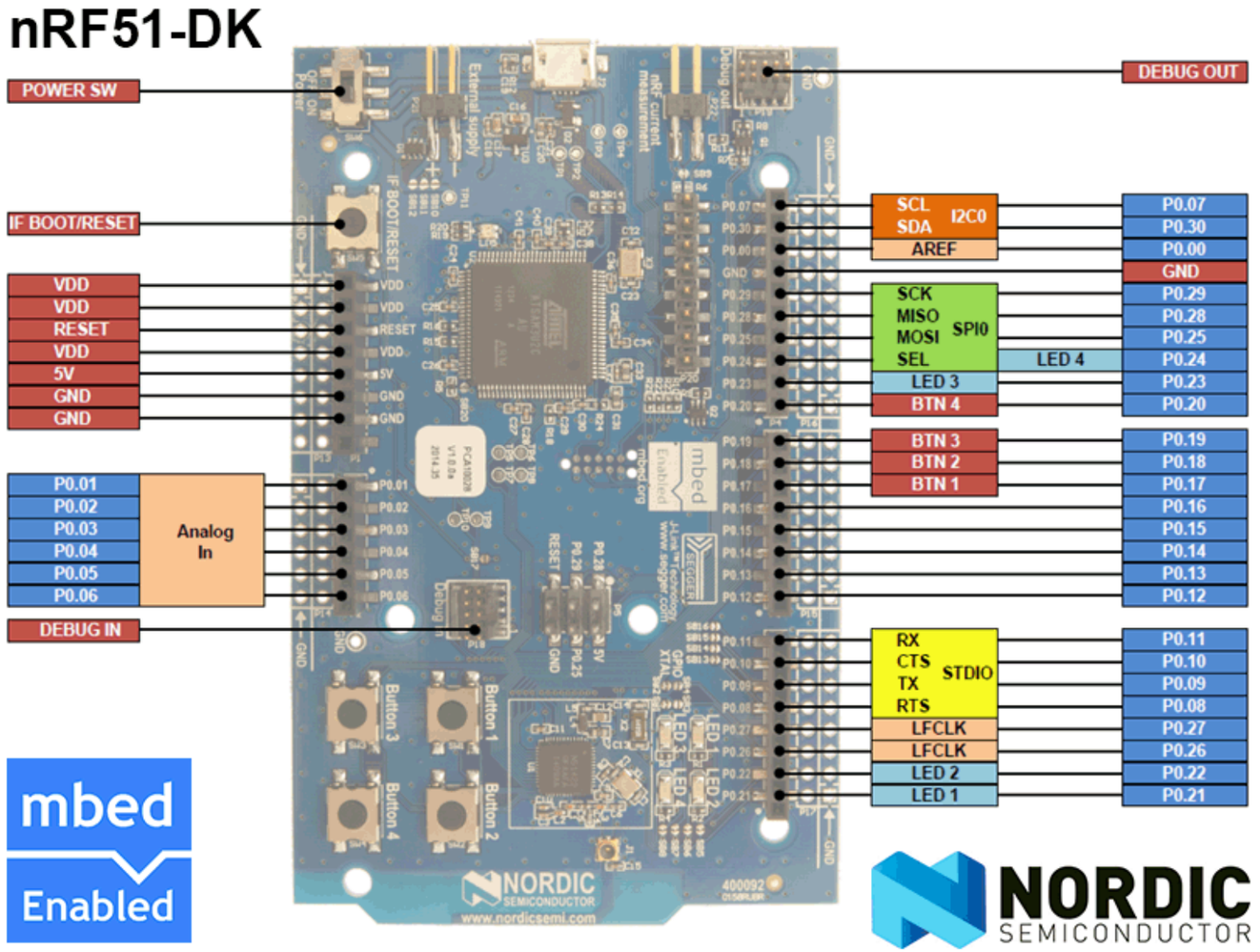


* “Impedance Matcher” between sleepy devices on constrained networks and IPv4/IPv6
* Addressable through REST call interface
* Secure

BLE mbed Device – What is it?

ARM mbed

* Ultra low power Cortex-M0
* 32KB RAM, 128KB Flash
* V4.1 Compliant BLE
* Arduino R3 Compatible Header
* Enumerates as USB flash drive
* Coin cell holder for low power operation
* J-Link and CMSIS-DAP debug Interfaces
* Pins for power consumption measurements



mbed Device Shield – What is it?

ARM mbed

* Grove Base Shield v2 from Seeed Studio
  * No soldering required
  * Simple plug-n-play sensors – supported by mbed!

* Grove Earbud heart rate sensor
  * Simple easy to use heart rate sensor attaches to earlobe

salesforce

Oh… One more thing…

* ARM would like to give each workshop attendee your mbed hardware!

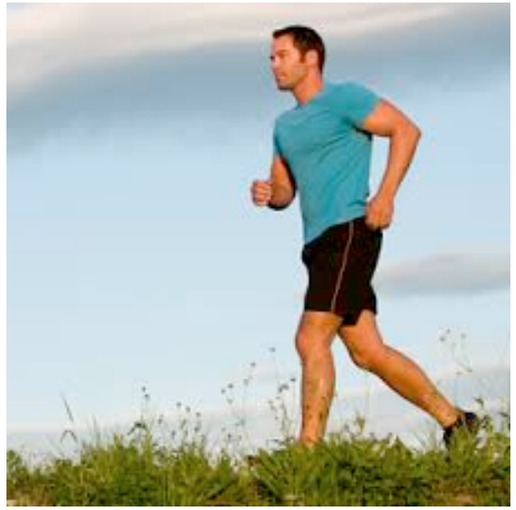
Just:

* Choose between the BLE or K64F mbed board options…
* Complete the workshop…
* The mbed board, Grove Shield, and Heart rate sensor are yours to keep!

Enjoy and Have Fun!

ARM® mbed™

Workshop Scenario: BLE Wearable Heartrate Sensor



* Create and connect a BLE-based heart rate sensor to your Salesforce.com account through ARM's mbed Device Server (mDS).

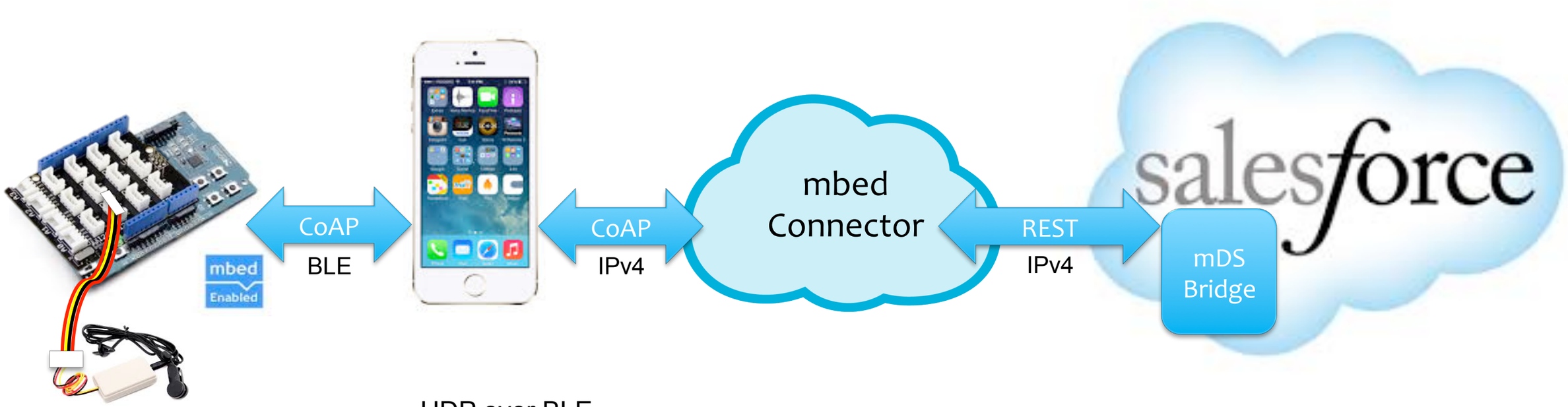
* Install, into your Salesforce.com developer account, an early access version of the ARM mDS Bridge Salesforce application. This application will connect your Salesforce account with a cloud-hosted instance of ARM's mbed Device Server

* Stream, via your Salesforce.com mDS Bridge application, heart rate, location, altitude, and speed onto a Google ™ map page. This page is included in the mDS Bridge application and makes use of Salesforce's Streaming API service.

IMPORTANT: Due to the amount of wireless activity in Moscone and the density of Cellular/WiFi Usage, you have two options

- **Give the BLE exercise a try**
- **Go with a wired ethernet exercise (K64F)**

Workshop Scenario: What we will build (BLE)…



Demonstrate delivery of devices, on constrained networks, into Salesforce.com

ARM® mbed™

LETS GET STARTED!
(10 steps…)

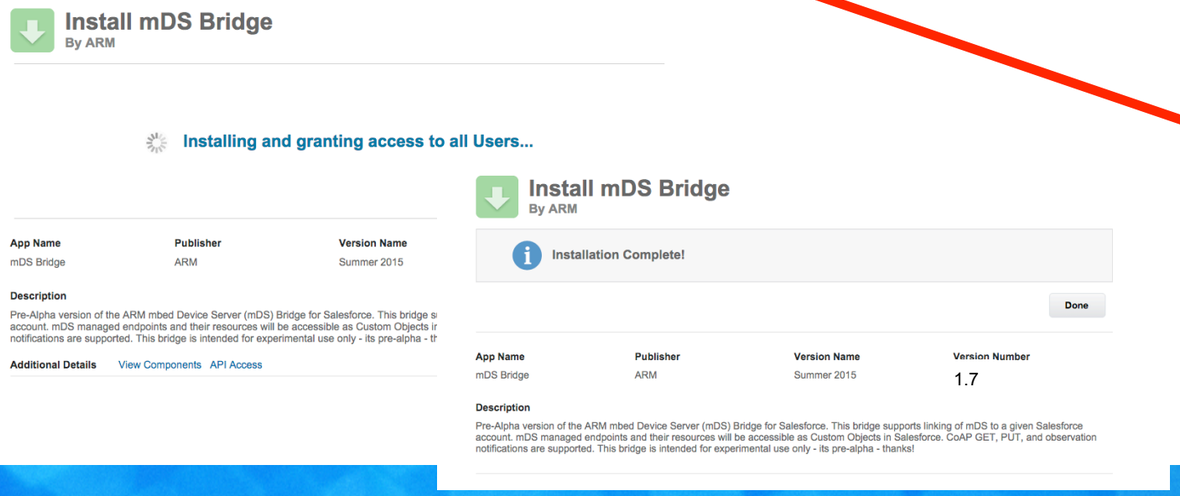
1). Import the mDS Bridge Package into Salesforce

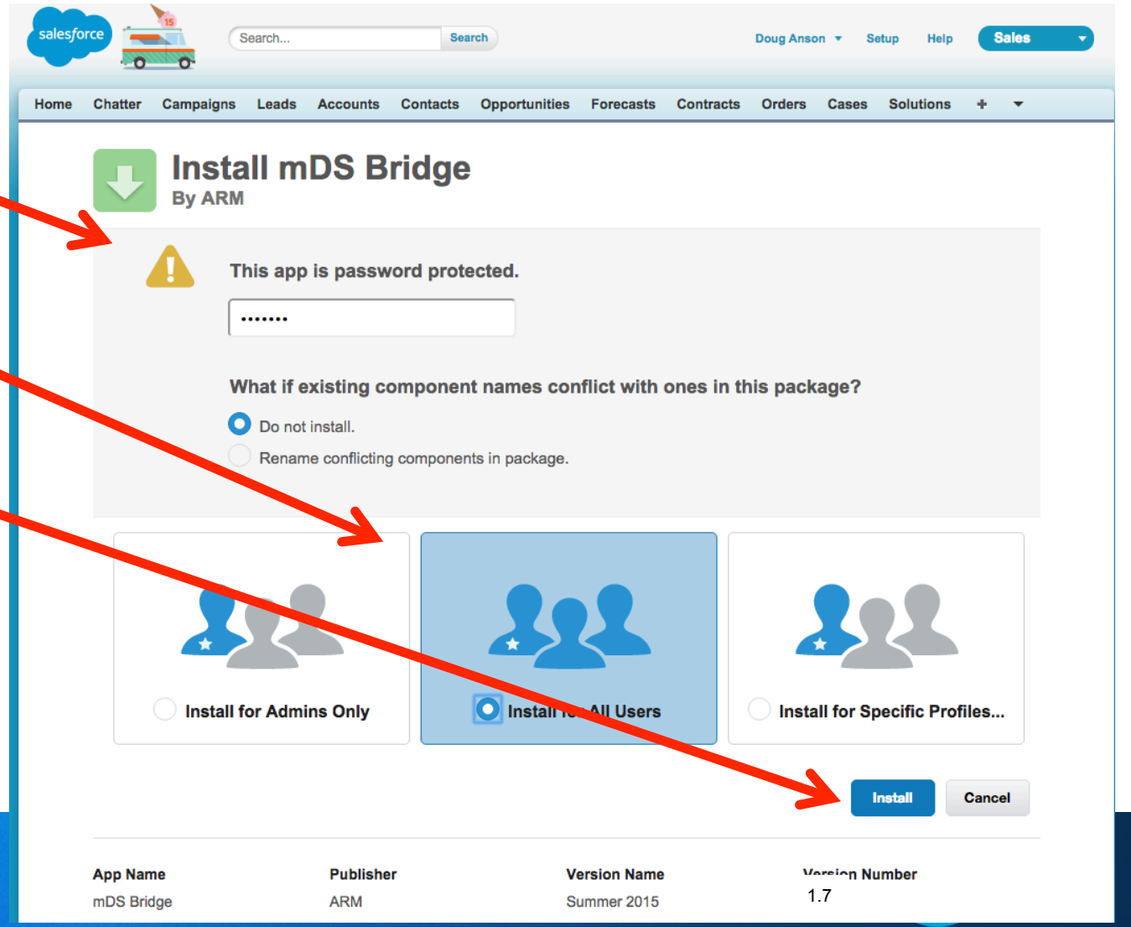
* Log into your Salesforce.com Developer Account

* Navigate to install version 1.7 of the mDS Bridge package :

  https://login.salesforce.com/packaging/installPackage.apexp?p0=04t610000001YSe

  * Password is “arm1234” (no quotes)
  * Select “Install for All Users”
  * Press “Install”



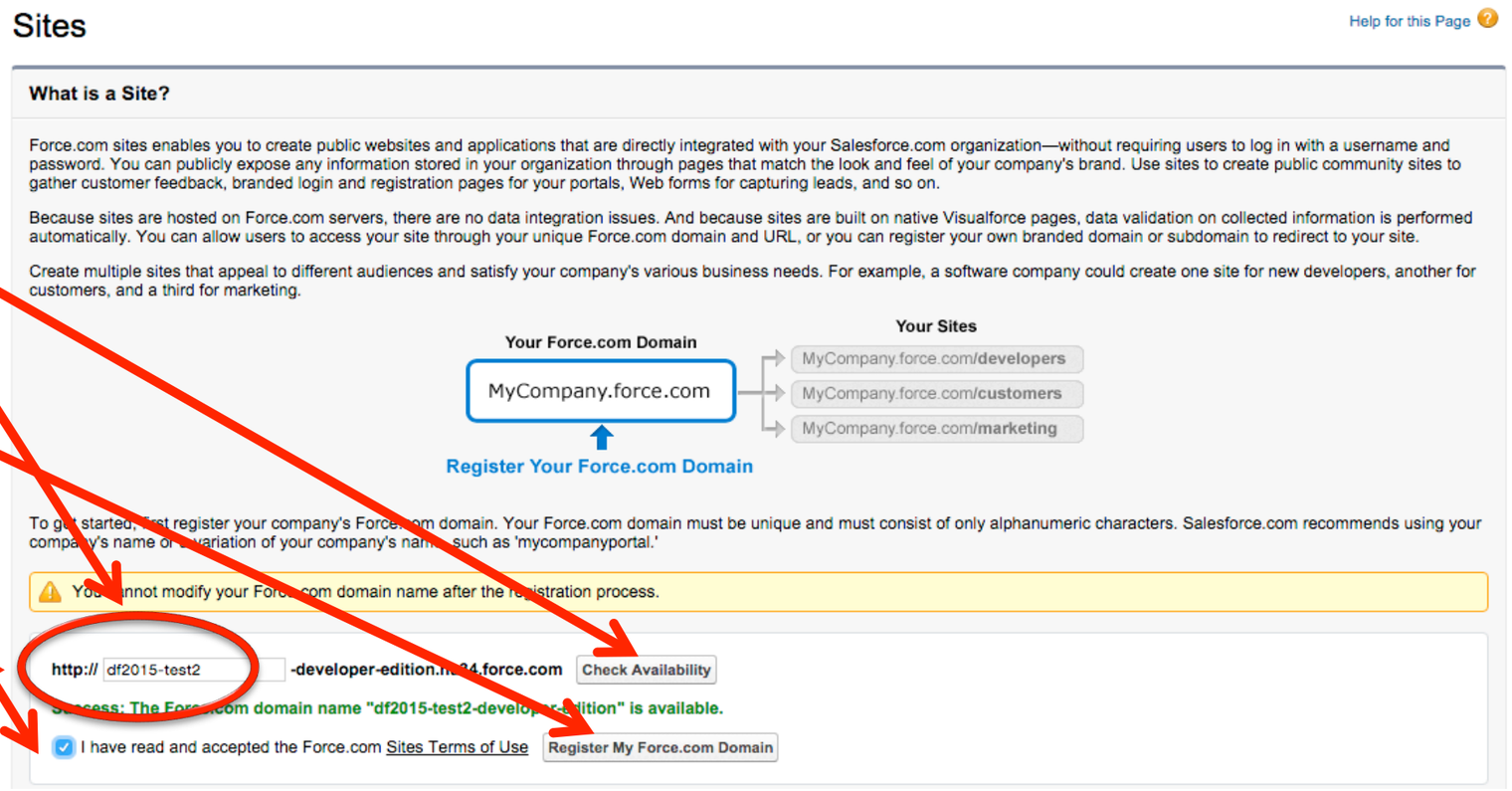


2). Create a “Site” in your Salesforce Developer Account

Starting at the “Setup” page, Look at the left-hand Navigation Pane

- Navigate to: Develop → Sites (Left-hand side… in the “Build” section)
- Select a domain name
- Press “Check Availability”
- Check “Agree to …”
- Press “Register My…”

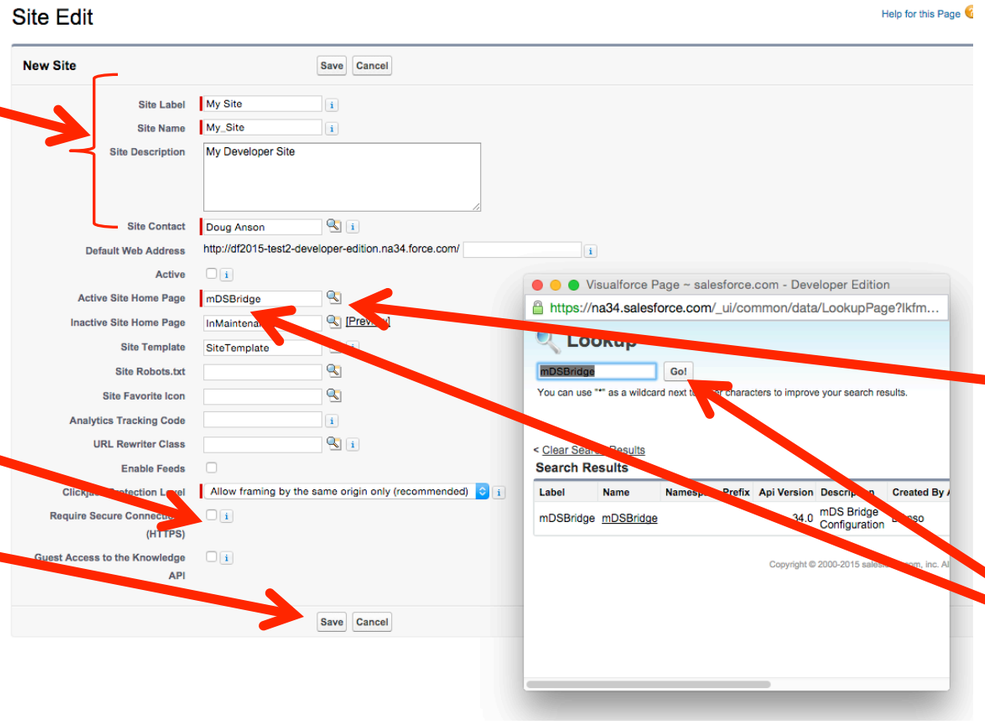
NOTE: Developer accounts are permitted *one* site domain ONLY… and it cannot be changed once set… *so choose this name wisely…*

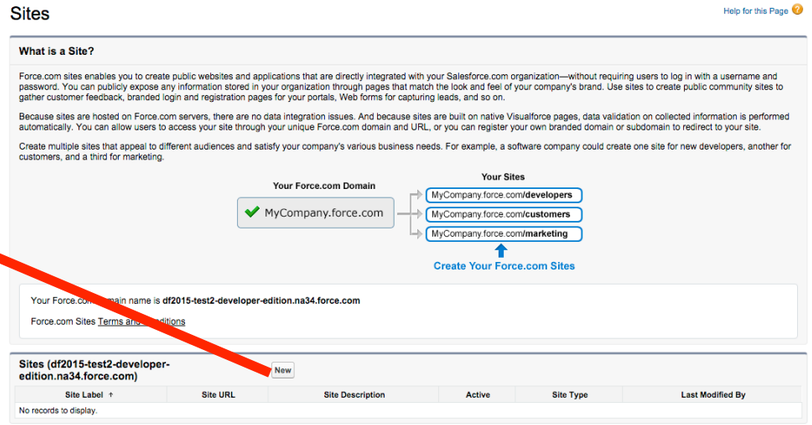


ARM mbed™

2a). Create a “Site” in your Salesforce Developer Account…

* Navigate to: Develop → Sites → Create Site… Click “New”
* Fill In Details
* Require HTTPS
* Press “Save”





- Find “mDSBridge”
- Select it

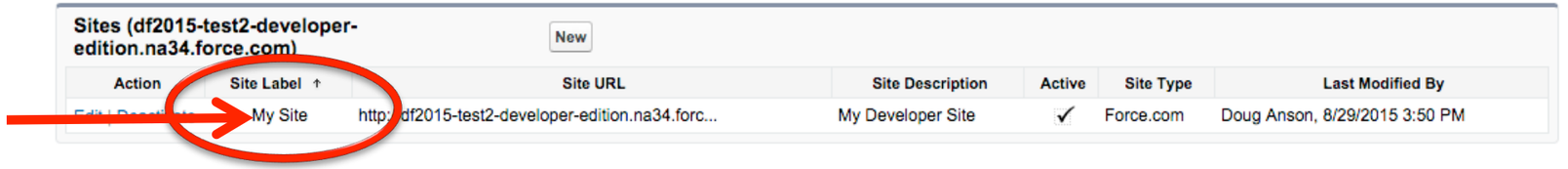
ARM mbed™

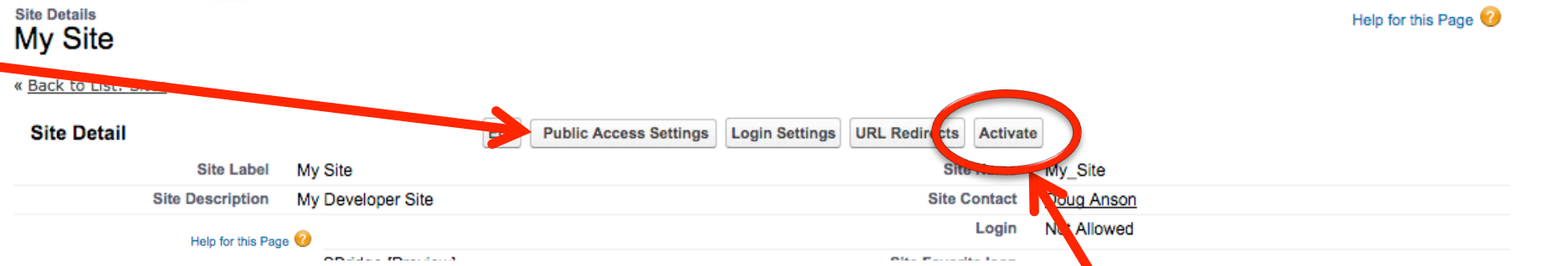
salesforce

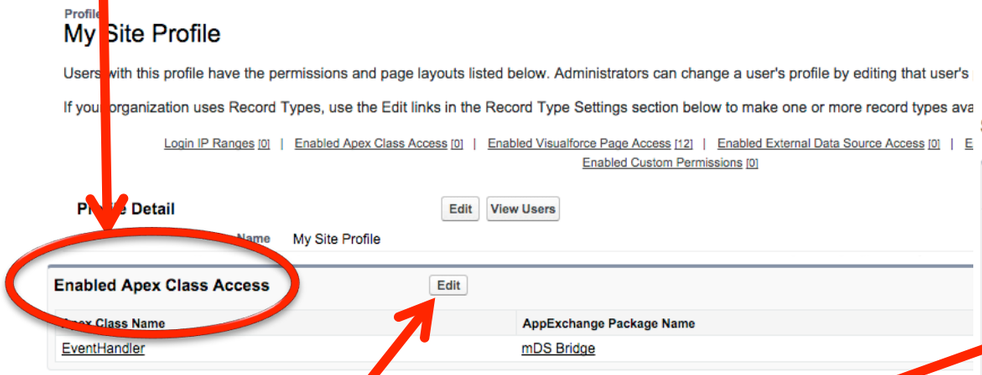
2b). Create a “Site” in your Salesforce Developer Account…

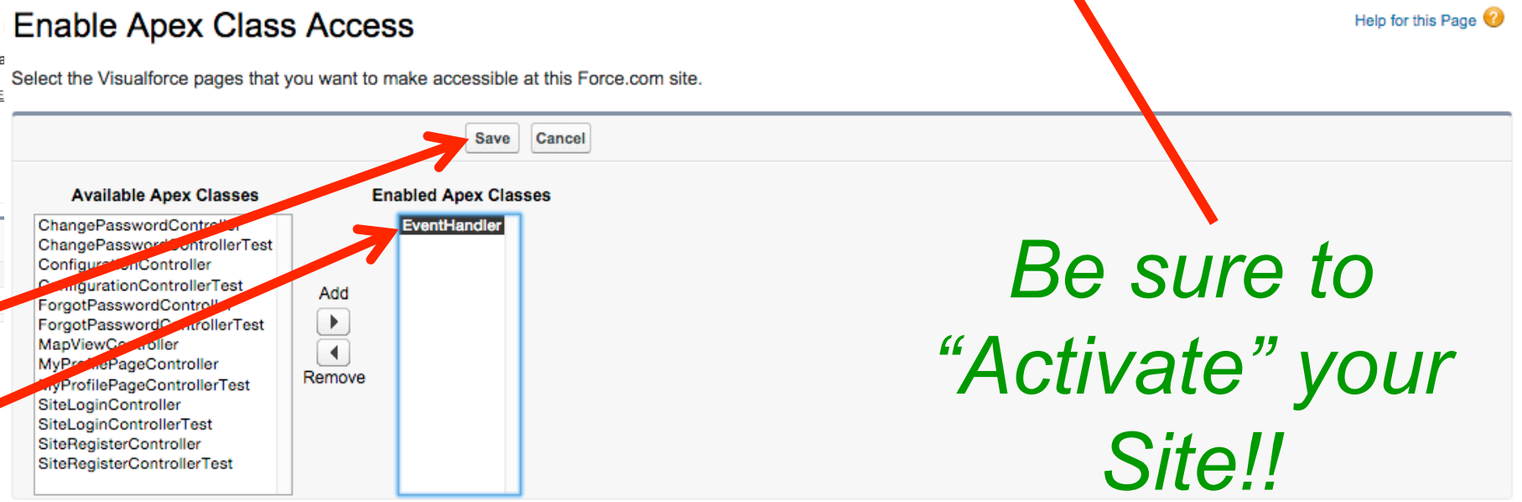
Starting at the “Setup” page, Look at the left-hand Navigation Pane (“Build” section…)

- Navigate to: Develop → Sites
- Click on your sites’ “Site Label”
- Press “Public Access Settings”
- Scroll down to “Enabled…”









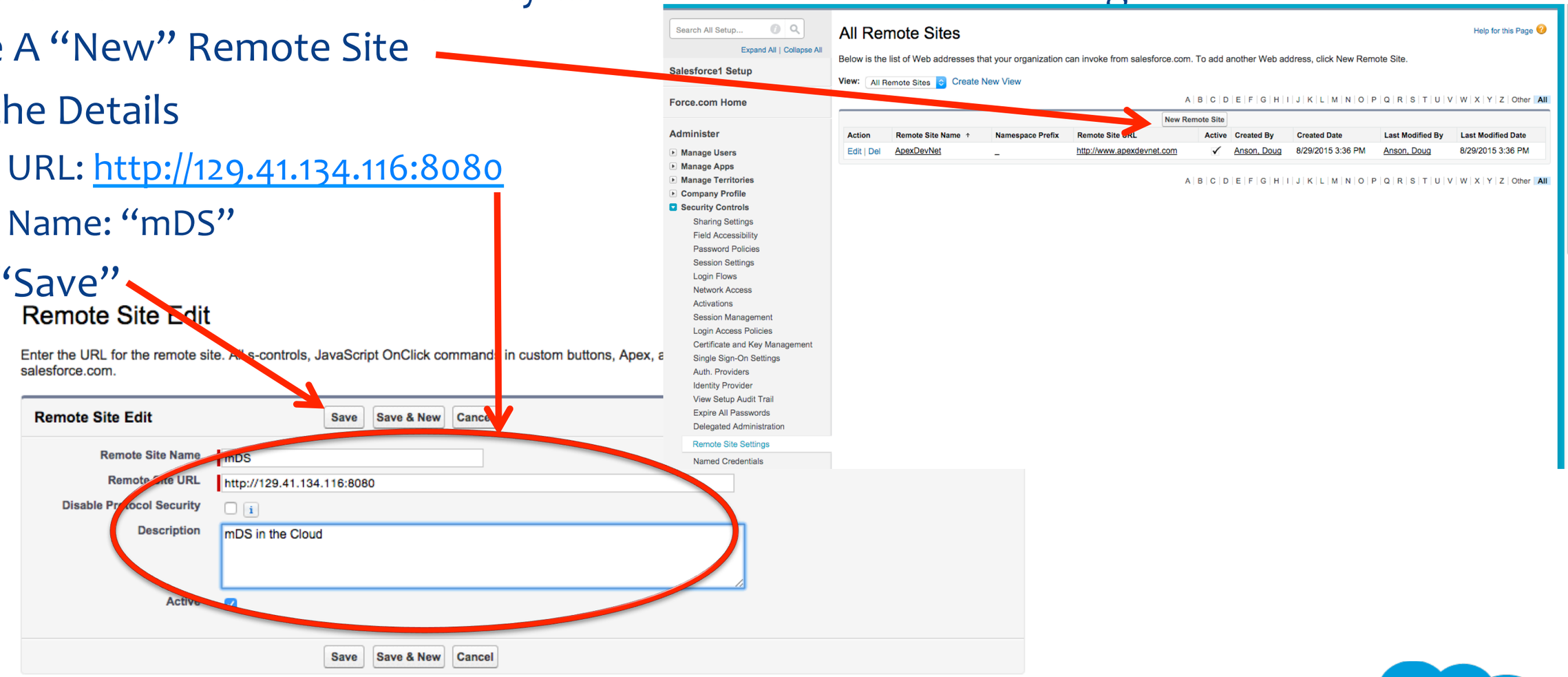
- Press “Edit”
- Find “EventHandler”, press “Add”
- Press “Save”

Be sure to “Activate” your Site!!

3). Add a “Remote Site” (i.e. mDS)

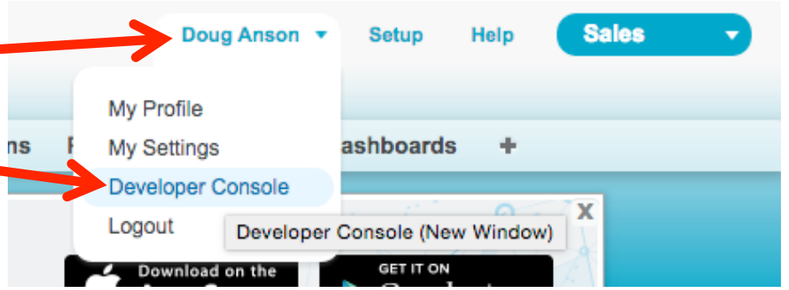
Starting at the “Setup” page, Look at the left-hand Navigation Pane

* Navigate to: Administer → Security Controls → Remote Site Settings
* Create A “New” Remote Site
* Fill in the Details
  * Site URL: http://129.41.134.116:8080
  * Site Name: “mDS”
* Press “Save”

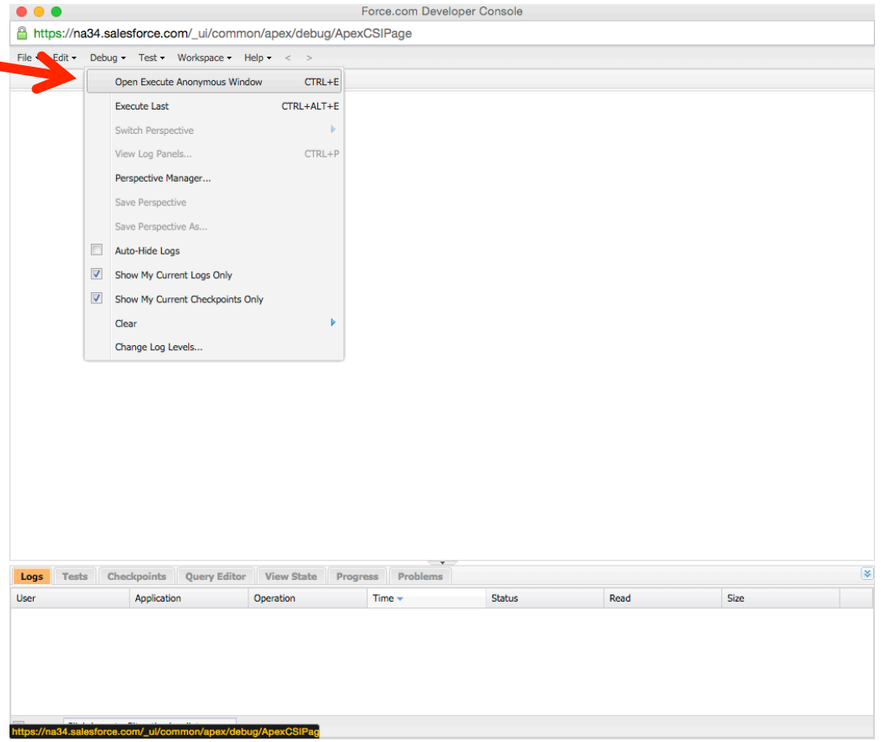


4). Install a PushTopic

* Starting at the “Setup” page, bring up the “Developer Console”

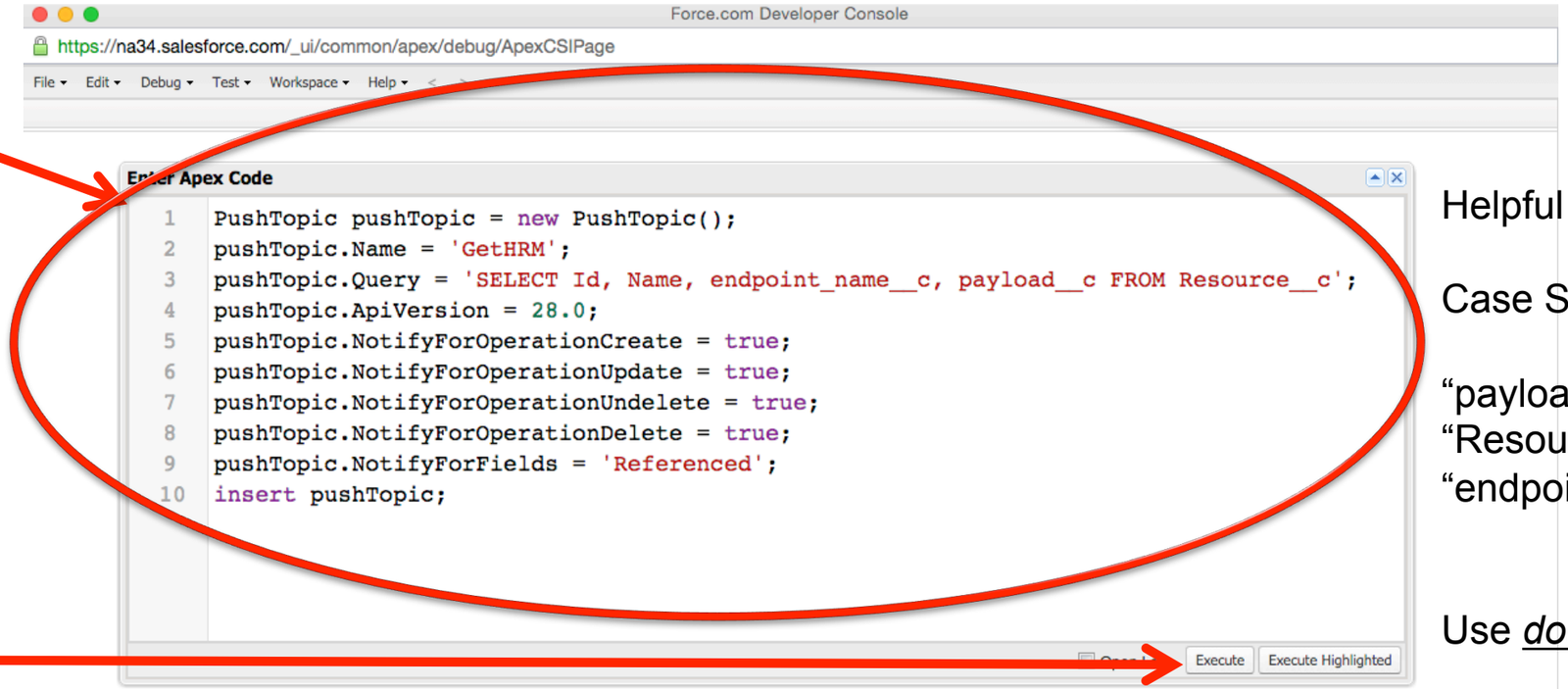


* Debug->Open Execute Anonymous Window



4a). Insert and Install this PushTopic…

* Enter the following code Snippet into the Execute Anonymous Window (ARMPushTopic.txt) :



Helpful Notes:

Case Sensitive!

“payload__c”
“Resource__c”
“endpoint_name__c”

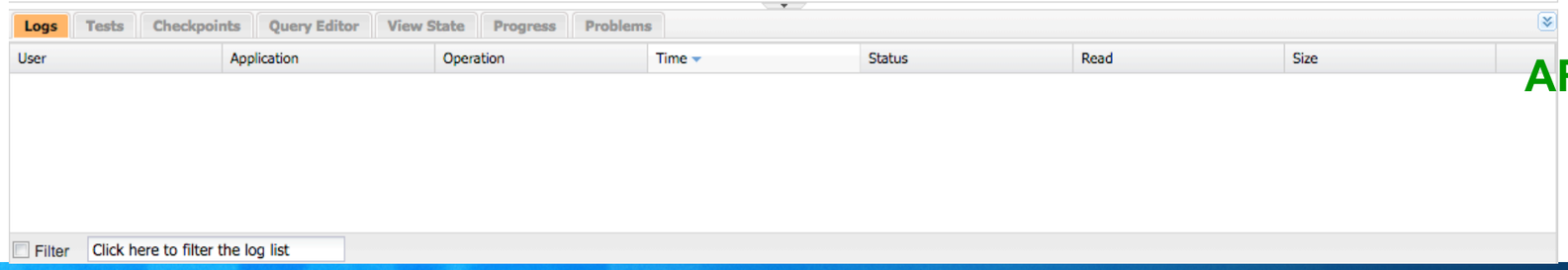
Use *double* underscore

* Press “Execute”

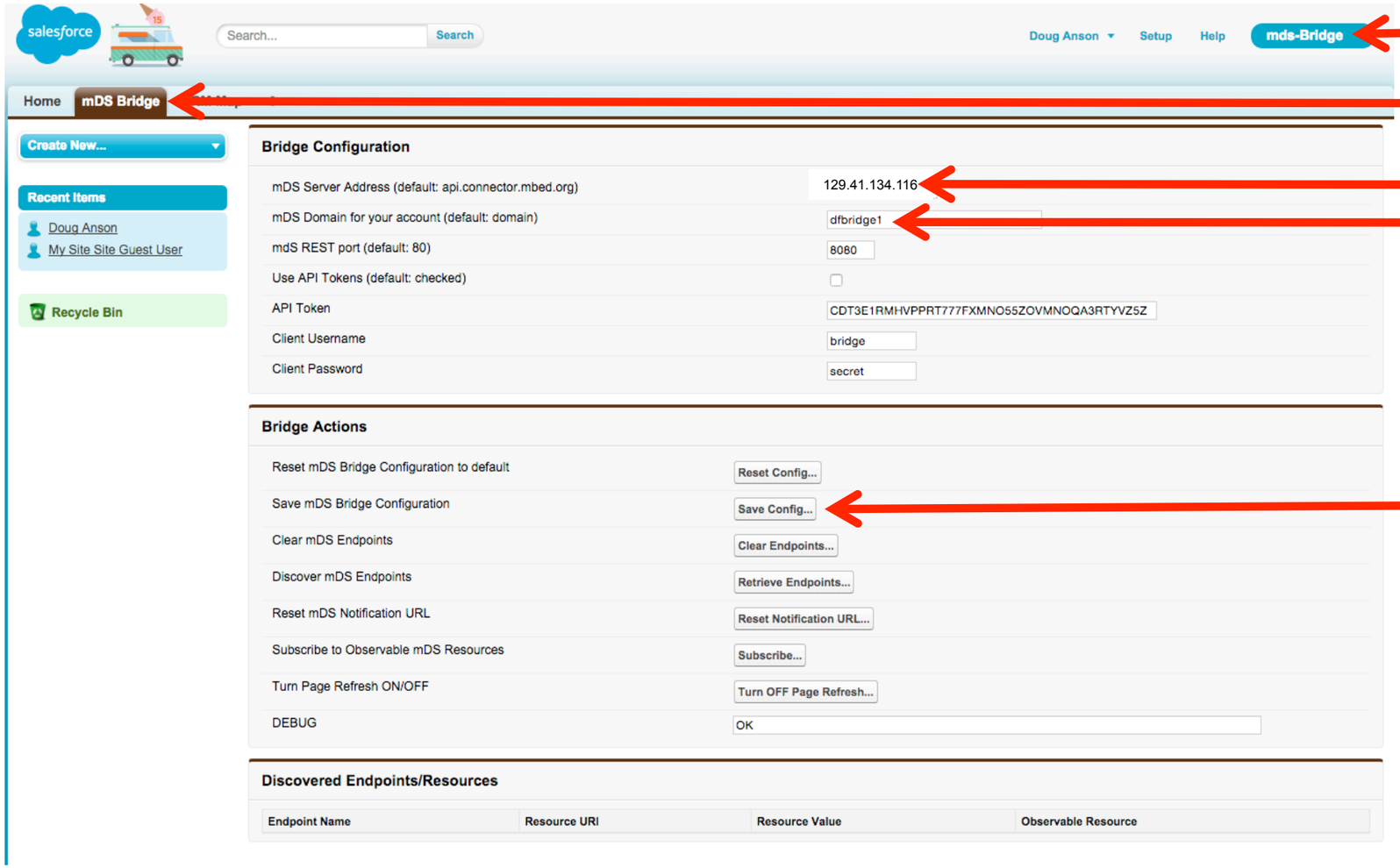
Code should be in a file on your desktop:

ARMPushTopic.txt

* Keep the console open… you can see debugging output from here…



5). mDS Bridge is Installed! Configuring it up….



Select “mds-Bridge” in this drop down

Select the “mDS Bridge” TAB…

Make sure its set to 129.41.134.116

Set to “dfbridgeXX” where ***XX is your number from the count-off at the beginning of the lab..***

Save your configuration!

Lastly, refresh your page and ensure that all of your settings are properly set !

6). BLE: Install the BLE-UDP Proxy on your Smartphone

iPhone Users

* Go to the App Store
* Search for “mbed BLE”
* Install the application
* Allow the application access to your location when asked

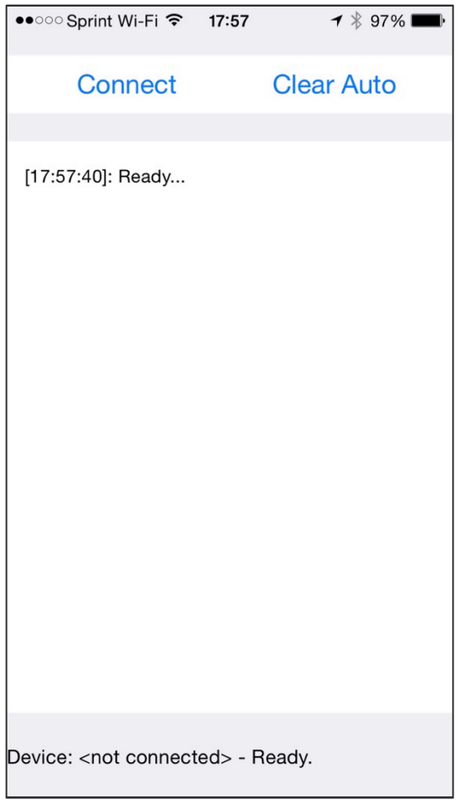
Android Users

* Go to Google Play
* Search for “mbed BLE”
* Install the application
* Allow the application access to your location when asked

Note: Android has had issues with the density of BLE in the room… not to worry… you can retry outside the forum

6a). BLE: Install the mbed BLE UDP Proxy on your Smartphone…

iPhone Users (author: Doug Anson – me!)

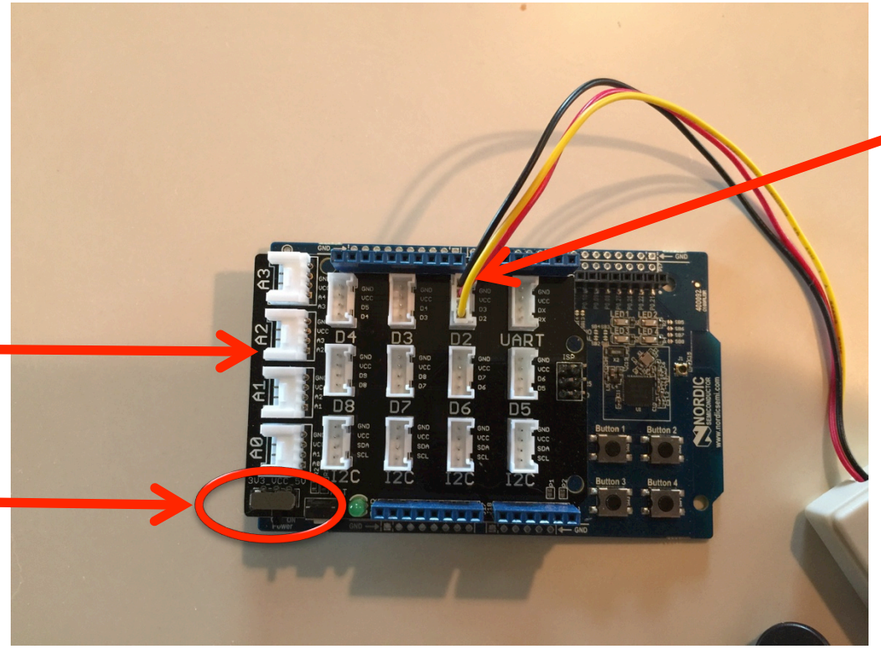


Android Users (author: flightbriefer.net – still me!)

Search for “mbed BLE UDP Proxy” in either the Apple App or Google Play store…

- Install
- Allow access to Location – this will be necessary for the map view

6b) Construct your mbed Device

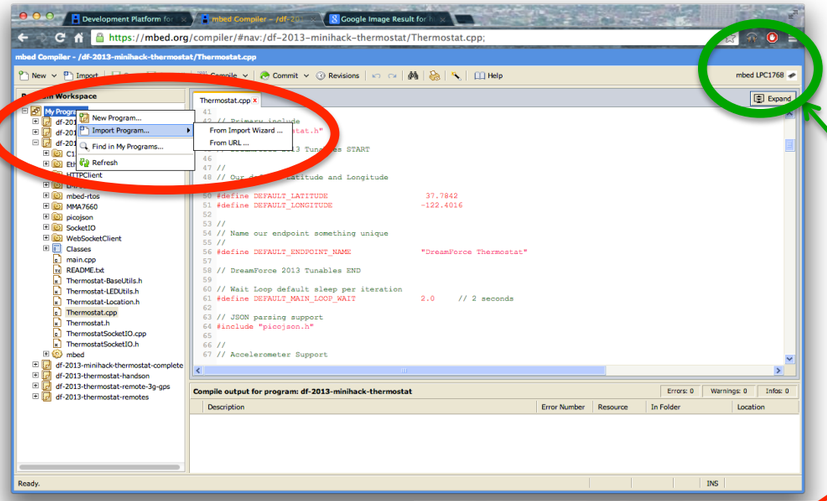


* Carefully remove the pin guards from your Nordic board
* Carefully insert your Grove Shield onto your Nordic board (K64F)
* Insert the heart rate sensor body cable into “D2” on the Grove Shield
* Insert the earbud phono plug into the heart rate sensor body

7). mbed Development: Create your dev account…

* Creating your (free) mbed developer account: http://developer.mbed.org , sign up
* In a new Chrome browser tab, go to http://developer.mbed.org . Log in
* In the upper left-hand part of the page, click “Platforms”
* Search for “Nordic nRF51-DK” – click on the picture
  * K64F Users: search for “FRDM-K64F” instead…
* Note the details about the board (cool… eh?). Click on “Add to Compiler”
* Now, in the upper-right hand part of the page click “Compiler”

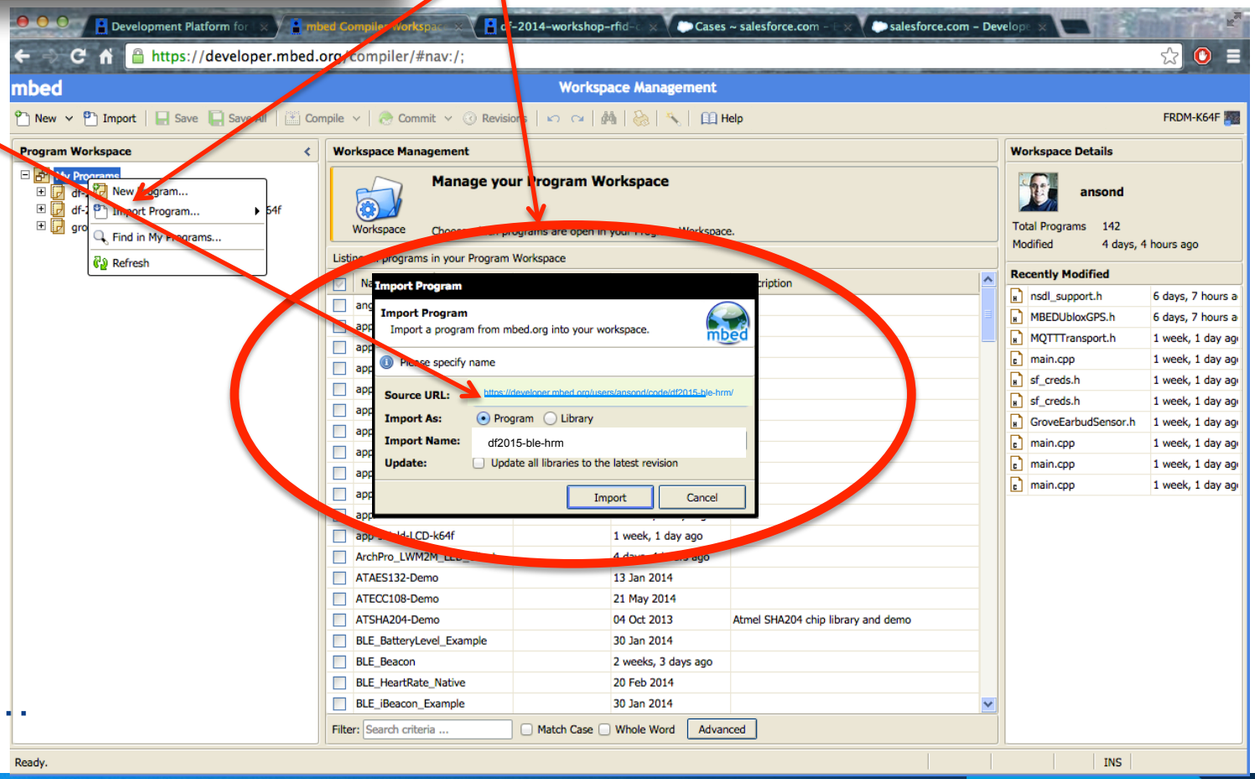
7a). … Import Code



Right-Click “My Programs”
Select: “Import from URL”

Make sure that “Nordic nRF51-DK” target selected… (or FRDM-K64F)

Select first row Press “Import from URL”



* Import using this URL:

BLE: https://developer.mbed.org/users/ansond/code/df2015-ble-hrm/
K64F: https://developer.mbed.org/users/ansond/code/df2015-eth-hrm-k64f/

* Press “Import”. Then, select Compile->Compile All
  * A “hex” (K64F: “bin”) file should be created and downloaded
* Insert the usb cable to the Nordic/K64F board, then the PC.
  * Let Windows find the device…
* You should see a flash drive called “MBED”
* Copy the downloaded “hex” (“bin”) file to the MBED flash drive…

7b). mbed Development: Customize and Re-Compile

* Expand the Imported Project – look for “main.cpp”

* Click on “main.cpp”… go to lines 47 and 50… Change as follows:

```
// My Endpoint Name
#define MY_ENDPOINT_NAME          ”dougs-hrm-XX”

// My NSP Domain
#define MY_NSP_DOMAIN             "dfbridgeXX"
```

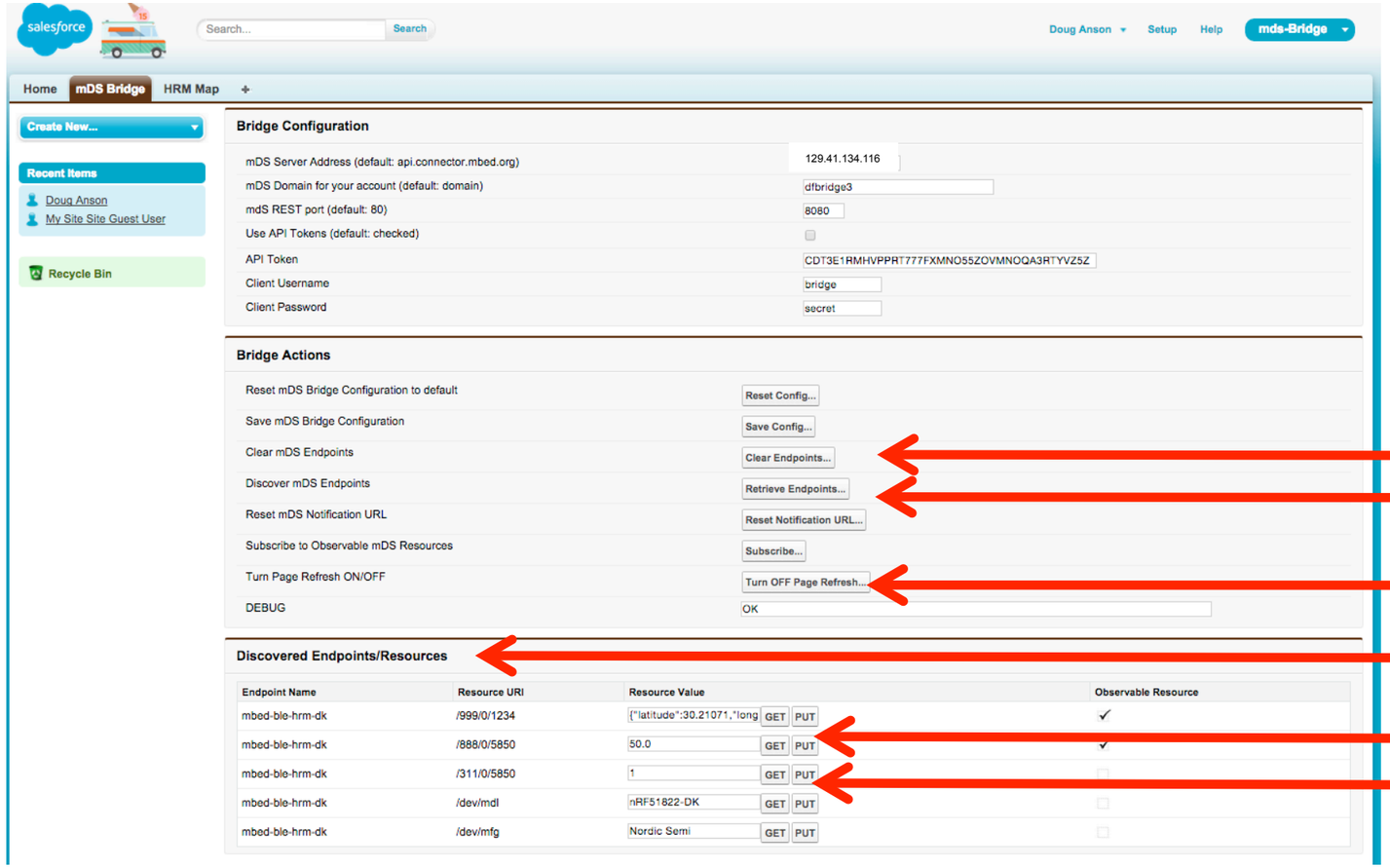
Set to something unique! You can use your number as part of the name

Set this to the same value that you set in the mDS Bridge Tab…

* Have a look at the other parts of the code… its very simple!
  * mbedEndpointResources directory has the actual CoAP implementation for the HRM and location resources
* Save the file, select “Compile->Compile All”.
  * Copy the downloaded “hex” (“bin”) file to the “MBED” flash drive
  * In your CoolTerm and/or Putty terminal, disconnect and reconnect… Send the “Break” command
    * Windows users may have COM issues with sending “Breaks”… so just disconnect and reconnect the cable

8). See your Endpoint in mDS Bridge on Salesforce.com

* Back on your Salesforce Developer Page:



Clear your endpoint out of Salesforce

Retrieve Your Endpoints into Salesforce

Click to turn “OFF” the Page Refresh

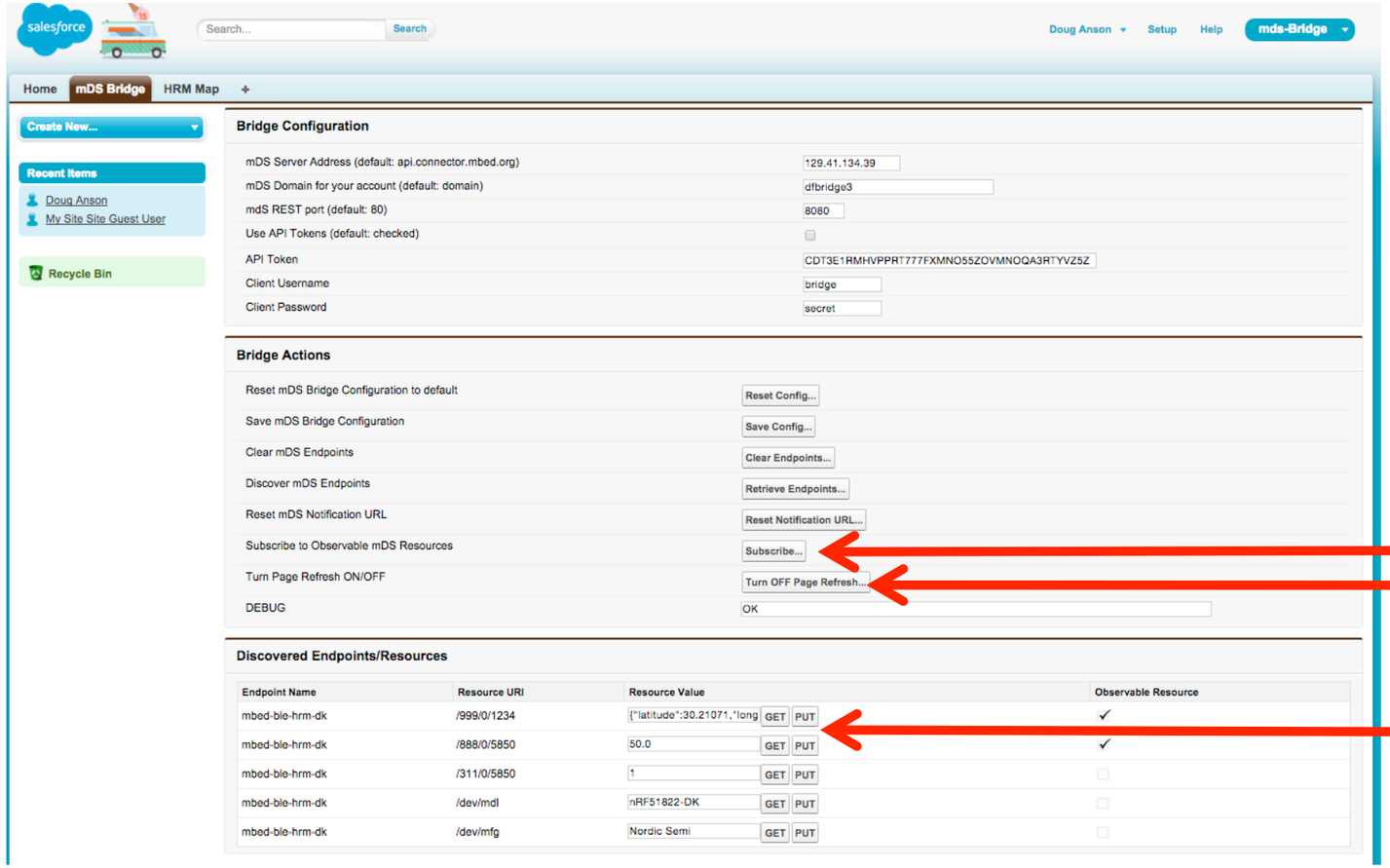
Your Endpoints Resources are listed!

Location and Heartrate Resource values

“PUT” a “0” or “1” to turn on/off LED light

9). Streaming Heartrate/Location….

* Back on your Salesforce Developer Page:



K64F Users – some additional notes:

- ***/dev/mdl and /dev/mfg will be different!***
- ***Endpoint name will also be different***

Press “Subscribe”

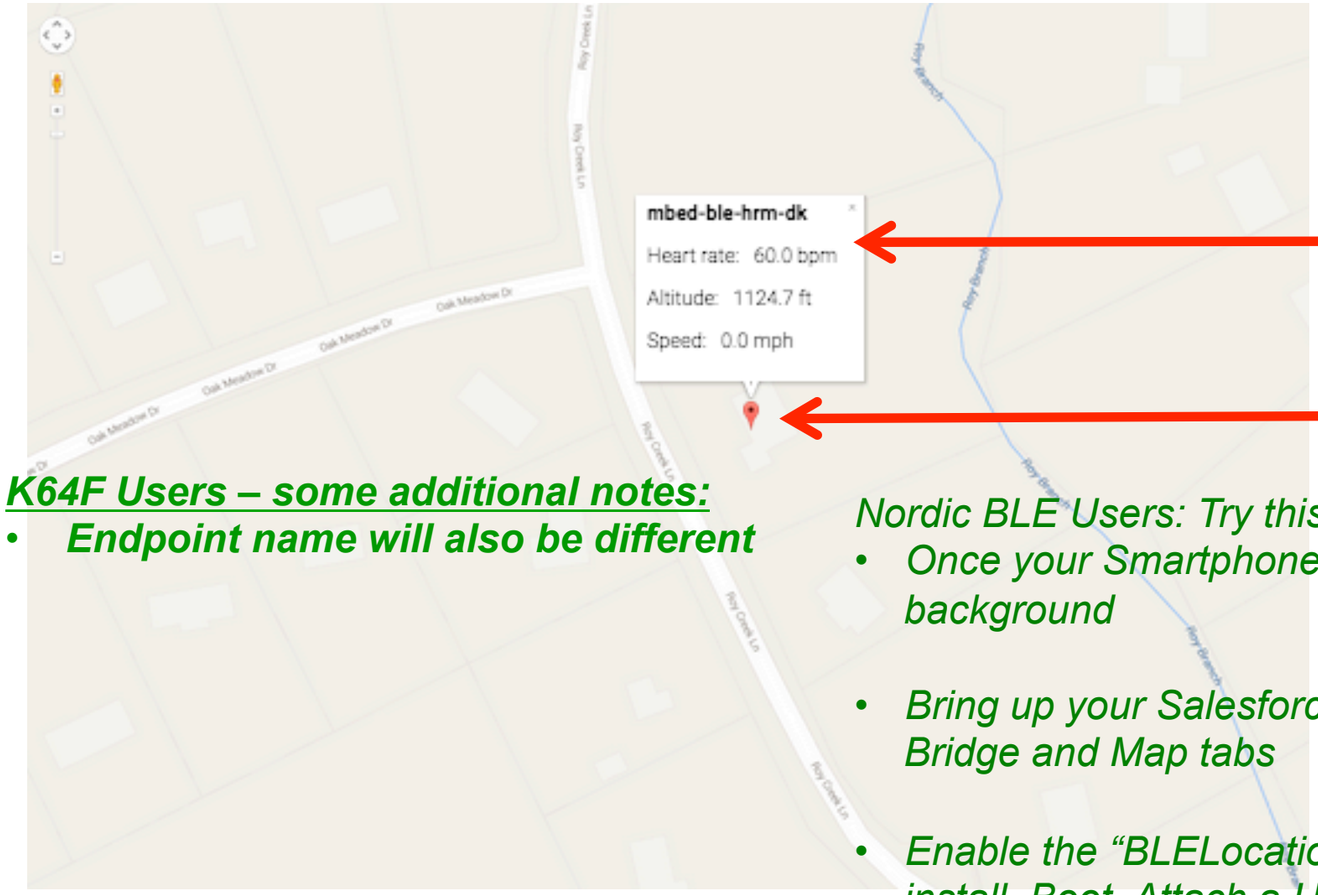
Click to turn “ON” the Page Refresh

Location and Heartrate Resource values should now update automatically…

This is called CoAP “Notifications”. Both Heartrate and Location resources are “Observable” (main.cpp)

10). Streaming Heartrate/Location….

* Now… check the “HRM Map” Tab/Page in the upper left-hand corner



Heartrate and Location are “Streamed” within Salesforce from the mDS Bridge

As your smartphone and heart rate sensor move around, this page will update with your latest heart rate, altitude, speed, and position on the map.

K64F Users – some additional notes:

- ***Endpoint name will also be different***

Nordic BLE Users: Try this:

- *Once your Smartphone’s Proxy app is connected your device, you can send it to the background*
- *Bring up your Salesforce page in your smartphone browser… log in… go to your mDS Bridge and Map tabs*
- *Enable the “BLELocation” experimental option in your mbed main.cpp. Recompile, install. Boot. Attach a USB-Battery. Re-subscribe in SF. Now, this page will display on your phone and will update is location, over 3G/LTE as you walk around..*

Congrats! Mission Accomplished!

* You have successfully connected your heart rate sensor, through mbed Device Server via CoAP, (over BLE) and IPv4/3G into Salesforce.com as a custom Object… so, you can now stream observable CoAP resources into Salesforce.com

* mbed Device Server and Salesforce.com together form a powerful platform for bringing devices, connected via various constrained networks, into Salesforce.com and its ecosystem.

* The mbed online development environment is a powerful and flexible ecosystem to build IoT applications on low power mbed devices

Thank You

ARM mbed™

Doug Anson – doug.anson@arm.com
Sam Grove – sam.grove@arm.com
Brian Daniels – brian.daniels@arm.com