



Massachusetts Institute of Technology
Fall Semester, 2011

MIT 6.S083: Building Mobile Applications

Assignment 6: *The preliminary part of this assignment is due before coming to lab on November 8. The rest of the assignment is to do in lab.*

App Inventor Bridge to Java

As you work on your projects this semester, you'll find things you want to do that are beyond App Inventor's capabilities (at least the current version of App Inventor). There are several approaches to extending App Inventor's functionality, and you can work with your coach to create appropriate extensions for your project if necessary.

This assignment treats one such approach, where you don't use the blocks language. Instead, you use Java and Eclipse and the Android SDK to create the same kinds of programs as you do with App Inventor. Only this time, since you are writing in Java, you can add new Java code to access additional features of the Android phone or perform other Java computations.

Don't worry on this assignment if you don't know Java. The actual Java programming required here is simple, and we can walk you through it in lab.

Before you can do the lab, however, you'll need to make sure that your computer is set up with Eclipse and the Android SDK. So do this setup work before coming to lab. The setup includes installing a lot of stuff. And you'll also need to go through a tutorial to create a "Hello World" Android project. All in all, the setup can take an hour or more, so give yourself adequate time.

Part 0: Setup

You must do this setup part of the assignment before coming to lab on November 8.

0.1: Install Eclipse

If your computer does not already have Eclipse installed, install "The Eclipse IDE for Java Developers", which you can get from <http://www.eclipse.org/downloads/>. Many MIT students have used Eclipse, and you should be able to get help if you run into trouble.

0.2: Install the Android SDK

Get and install the Google Android SDK (Software Developer Kit) from

<http://developer.android.com/sdk/installing.html>

Follow the instructions there, which include installing the ADT (Android Developer Tool)

package for Eclipse.

0.3 Do the Android Hello World Tutorial

When you get to “Next Steps” in the Android SDK installation process, you’ll be directed to do the Android Hello World Tutorial at

<http://developer.android.com/resources/tutorials/hello-world.html>

Go through the part entitled “Run the Application”.

Don’t go on to the section “Upgrade the UI to an XML Layout”. Instead, plug in your phone and try Run > Run again. This time you’ll get the option to select the phone versus the emulator. Select the phone. You should see your (trivial) application running on the phone.

Finally, look at the Hello World project structure at the left. You’ll see an entire directory tree, with parts including src, assets, res, and others. These were created by the Eclipse ADT plugin. Every Eclipse Android project will have a tree like this.

Part 1: Working with the Java Bridge

Download the file [BridgeToJava.zip](#) and unzip and save the result as a folder on your computer. Now start Eclipse and create an Android project as you did in the Hello World tutorial by going under the File menu in Eclipse and selecting New > Project and then choose Android Project. Only this time, when you get to the *New Android Project* window, select the radio button for *Create Project from Existing Source*, and browse to the BridgeToJava folder you just saved. Eclipse should create a project called BridgeSample, which you should see listed at the left of the screen.

Highlight this new project and run it by choosing Run > run, as you did with the HelloWorld tutorial. Try running it both with the emulator and the phone and play with it a bit to see what it does.

Go to the BridgeSample project tree at the left of the screen and find the file src > com.appinventor.sample.bridge > BridgeSample.java. Read through the Java code there to see how the application is implemented. The comments show the main conventions for creating simple apps:

- every component has a *container* and a component is created by passing its container as an argument. The top-level container for the app is (*this*).
- You make a component handle an event by *registering (this)* to response to that event, and then creating an *event dispatcher* (here the procedure dispatchEvent) that checks the component and the event name and performs the required operation.

Make a change to the app. It can be anything you like. Just verify that you can modify the code and rerun it.

Part 2: More examples

Browse to the directory <http://code.google.com/p/appinventor-java-translation/downloads/list>. Here you'll see Java translations of some of the App Inventor tutorial apps. You might have built one or two of them at the beginning of the semester,

Download one of these apps, study the code, and modify it.

Turning in your work

There's nothing to turn in for this assignment. Keep working with your team. But you might be the Java Bridge technique useful as you work on your project for the rest of the semester..



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/) .