

Test Driven and Behaviour Driven Development For Mobile Applications

Chris Shotton

BT Glasgow GDC - Mobile Applications Development

This document will cover the set-up and reconfiguration of mobile applications projects for iOS and Android for use with the new CI testing server. It will also cover the use of the TDD (Test Driven Development) and BDD (Behaviour Driven Development) methodologies and technologies required to use it effectively.

***Index Terms*—TDD, BDD, Android, iOS, Mobile Applications Development, Jenkins, Continuous Integration Testing**

I. INTRODUCTION

THE new Jenkins CI server should be up and running now in the lab ready for use with TDD and BDD technologies to allow us to improve our CIT (Continuous Integration Testing) practises and reduce the load when performing regression testing. Initially it has been configured to handle Android and iOS projects allowing them to be checked out of our shared SVN repository, be built locally on the server, unit tested and behaviour tested automatically. There are a number of interlinked technologies that need to be used to get the most out of this system so I'll be doing my best in this document and the associated example project files to outline what you need to do to use the system effectively for new and existing projects.

Last Updated:
16th August 2013

II. JENKINS CI SERVER

Pre-requisites:

- Java 1.6
- Android SDK
- X-Code
- Red side connection x

Currently the build and CI server we are using should be a Jenkins CI¹ server deployed on a Mac Mini in the lab. The Mac must be used to allow Android and iOS builds on the same machine so this guide will specifically cover the Jenkins and related packages set-up on OS X however all development related sections are platform independent. The role of the CI server is to pull project code directly from the careapps SVN, build it, execute our test packages on it and provide a central resource for the results for each build. This section will cover the set-up and configuration used for this server should anyone need to set it up again in the event of a natural disaster, Godzilla, terrorist attack or coffee related incident.

The distribution used for the active server is Jenkins 1.527 for OS X which is included in this package and can also be obtained from the Jenkins web page.

Install the package. The package is currently broken and doesn't install correctly on OS X. To resolve this run the shell script in this package located at `'/Scripts/configure_jenkins.sh'` which should open a file in X-Code. In this file change the entry for 'Username' to 'jenkins', save and close.

To start Jenkins use the `'start_jenkins.sh'` script in the included Scripts folder. Open up Jenkins in the browser using address `http://localhost:8080`². Navigate to the 'Manage Jenkins' screen and select 'Configure Global Security'. On this page select 'Jenkins's own user database', check 'Allow users to sign up'. Select 'Matrix-based security' for the authentication method and enter the name of the admin account you want to use. Check ALL the tick boxes for this user. For Anonymous check only 'Overall:Read'. Make sure you've added the admin account before you save the page or BIG TROUBLE as you'll be locked out and as I found out many times undoing this is a total pain. Save and it should take you to the login page. Click 'Create an account' and make an account with the same name you used for the admin account, set a password and log in. New users will only be allowed read access but will be able to check the results of any active projects.

Next there are a few plug-ins we need to support the testing packages used. Return to the Manage Jenkins page and select 'Manage Plugins'. On the 'Available' tab select and install the following plug-ins:

- Android Emulator Plugin
- ANT Plugin
- cucumber-reports
- Environment Injector Plugin

Once these are installed return to the 'Manage Jenkins' page and select 'Configure System'. For now all you need to configure is the Android section. Enter the home directory for your installed Android SDK into the Android SDK section and check 'Automatically install Android components when required'.

That's it. Jenkins should now be basically installed and configured. Your server can be started and stopped using the start and stop scripts in the included Scripts folder.

¹<http://jenkins-ci.org/>

²If this doesn't work see: <http://shashikantjagtap.net/adventures-with-jenkins-macosx-linux/> for further details

III. TESTING FOR ANDROID

For Android we will be using two packages for our testing; JUnit Android Instrumentation for TDD and Calabash-Android for BDD. For this your project needs to be configured correctly and have both related testing set-ups included for the testing to function. First we'll cover the basic set-up of an Android project using BTWiFi as an example of an existing project modified to work with the testing system. This will be followed by a brief overview of how JUnit and Calabash can be used to create tests for your Android project. This document will cover some simple examples but will not go into great detail of how these testing systems are used as it will vary greatly project to project and is beyond the scope of this project.

A. Preparing an Android Project For Testing

First of all we need to adjust your Android project such that it can be built and processed by the testing server. To do this we need 3 things:

- Android Project
- Android Testing Project
- ANT Build Script

To do this check out your or create your project in Eclipse. To create an associated Android testing project select 'File->New->Other' and from the options select 'Android->Android Test Project'. Use the project name of the project you are creating the test package for and append '-Test' to it just for ease e.g. BTWifi-Rebrand-Test'. Hit next and select the project you're creating the test project for from the next menu and hit next. Select the same SDK as your project uses and finish. This will create your testing project and associate it with your core project. For now we're not going to create any tests here but see the JUnit Instrumentation section below for more detail on what we'll use this test project for.

- Green side connection

Next we need to create a Jenkins job for your project. This will contain all the information required to locate your project repo, build the project and execute the required packages on it to produce your test results. To begin go to the Jenkins home screen and select 'New Job' from the side menu. For our jobs we want to use 'Build a free-style software project'. Name your project what you like for this example we'll use 'BTWifiTestingExample'.

To hook it up to the careapps SVN select 'Subversion from the 'Source Code Management' section. Enter the

B. Configuring JUnit

APPENDIX A

PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

APPENDIX B

Appendix two text goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.



Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.