Step 1 Baseline 1

Step 1.1 – pull latest code and instructions 1

Step 1.2 – run project 1

## Step 1 Baseline

In this lab, we are going to be new developers on the san-francisco-tourism project. The project currently uses JUnit 4. The goal of step 1 is to confirm we have a known good state to start with.

### Step 1.1 – pull latest code and instructions

Clearly, the first thing we have to do is pull the latest code for the project! Your teammate (me) helped by pulling an older version of the project to test connectivity and make it as easy as possible.

TODO – create screenshots for eclipse and intellij once set up VM

There were likely changes to this document after it was created so please copy the latest from https://github.com/boyarsky/JavaOne2017-HOL-JUnit5

### Step 1.2 – run project

This project currently uses JUnit 4. We want to make sure that we can run builds now before changing anything. After all, we want a known good state before writing code.

TODO – create screenshots for eclipse and intellij (maven and built in UI) once set up VM

## Step 2 – Convert to JUnit 5

JUnit 5 is available and we want to use it. The goal of step 2 is run all the JUnit 4 style tests using a JUnit 5 runner. This is the first step in migration on a real project. Once all the plumbing is ready, it will be time to actually using JUnit 5 syntax.

Note: If you aren’t sure what the updated pom.xml should look like based on this description, feel free to look at the solution.

1. Add property to pom.xml so can run unit tests

<surefire.version>2.19.1</surefire.version>

1. Add properties to pom.xml so can

<junit.jupiter.version>5.0.0-RC3</junit.jupiter.version>

<junit.platform.version>1.0.0-RC3</junit.platform.version>

1. d