Faculty of Science

Course: CSCI 2020U: Software Systems Development and Integration
Term: Winter 2022
Lab: #3
Topic: Build tool 2

Overview

In this lab, you'll build the same project directory as with the previous lab, and improve your Gradle build. The build will now generate a JAR file, include a run from JAR task, and try out a plug-in (minify).

Part 1: Run Project

Boot into your Linux partition (or open Linux in a VirtualBox VM). You will create a new directory in your home directory called `csci2020u/lab03`. We'll add this folder to the Git repositories. Use the following steps to complete this lab:

1. Change into the csci2020u directory
2. Make a new directory, called lab03
3. Either copy the contents of lab02 to lab03 (recommended), or else re-create them
4. Add the capability to run the program
5. Copy the test file, data.csv, from the course website, and save it to the lab03 directory
6. Run the program

Part 2: Deploy Project to JAR File

7. Modify the build file to be able to create a custom JAR file (version: 1.0), using the `CSVDemo` class (created in lab 02)
8. Observe the JAR file in `lab03/build/libs` with the `ls` command

Part 3: Execute Plug-In

9. Download ProGuard from `http://sourceforge.net/projects/proguard/files/` (download the .tar.gz file for the latest version)
10. Copy/move the .tar.gz file for ProGuard from `/home/csci2020/Downloads` to `/home/csci2020`
11. Extract the .tar.gz file using the tar command (e.g. `tar zxvf proguard-7.1.0-beta5.tar.gz`)

12. Note the folder created, as a result of the above command
13. Add ProGuard to the build file (remember the directory created, above, and the JAR filename created in **step 2**)
14. Create a `proguard.cfg` file, using the one from the lectures as a template
15. Run the task to minify the code in the JAR file
16. Note the new .jar file in the `build/libs` directory

***Note**: This minify task is quite useful in large projects, but isn't very useful for our single-method class. It was merely to provide a demonstration of how typical plug-ins can be used within Gradle.*

Extra Challenge (Optional)

Try to implement the Lint task, developed by Netflix, into your build. You will find instructions on how to use it at `https://github.com/nebula-plugins/gradle-lint-plugin`.

How to Submit

In session

(Preferably)

- Show your local and remote repositories to the TA to prove that you have finished this lab.
    - This can happen by your sharing your screen to the TA or direct messaging them with screenshots.

After lab hours

(1 week to submit - before your next lab session)

- In one PDF documents attach the following:
- Your full name, SID, and lab session (time slot)
- Screenshot your command line terminal with the steps
- Screenshot your local repository where relevant
- Screenshot your remote repository showing creation, commit, branching, and merging history
- Link of your GitHub repository (if it is a public repository)

The TA can provide oral feedback if you do not receive full marks for any lab assignment, but it is most appropriate to ask the TA for this feedback in a timely fashion (i.e. ask now, not at the end of the term).