

COM S 514

Engage Application Deployment Documentation

Team 2 - Dipanjan Karmakar, Lei Liu, Prateek Gupta and
Nikita Tiwari

Deployment Document

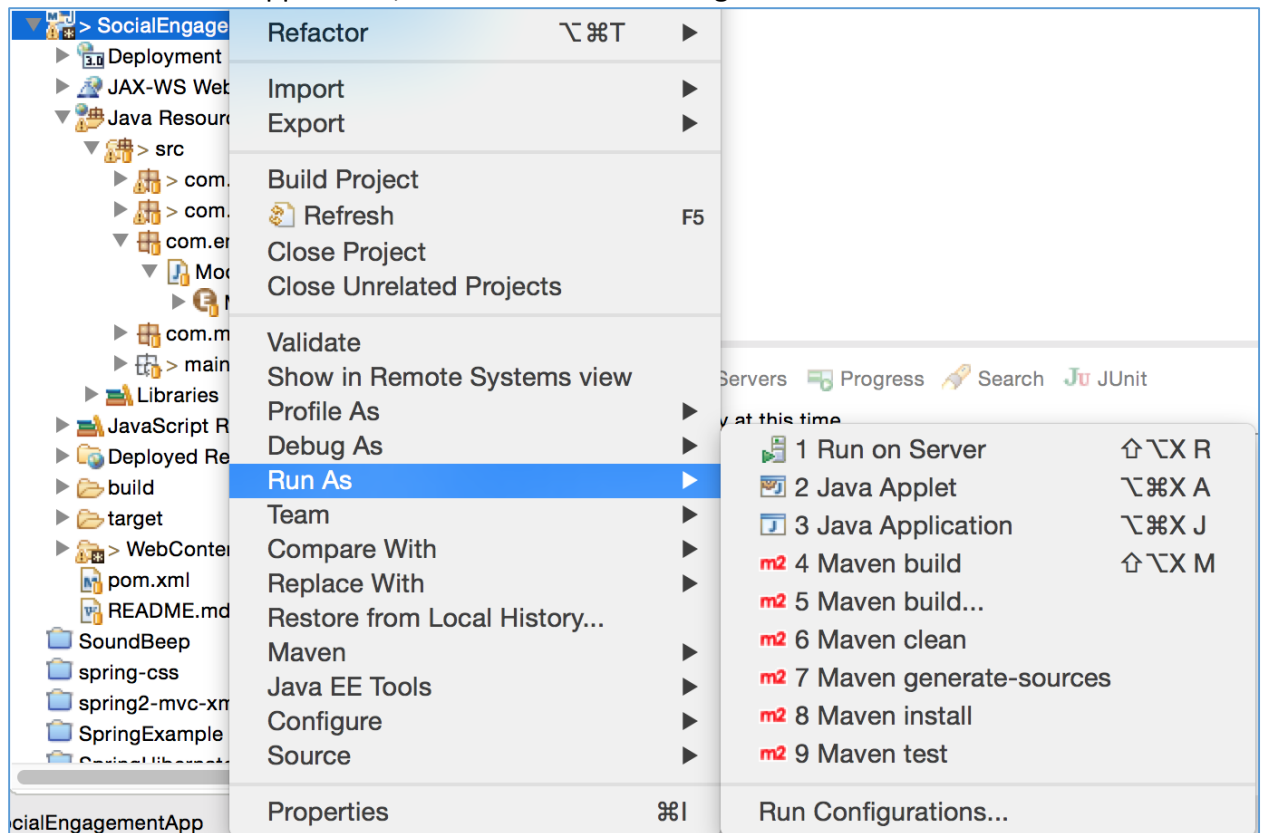
Engage is a Java based Spring-3 Web application.

We need to deploy it in an application server [below example shows Tomcat server].

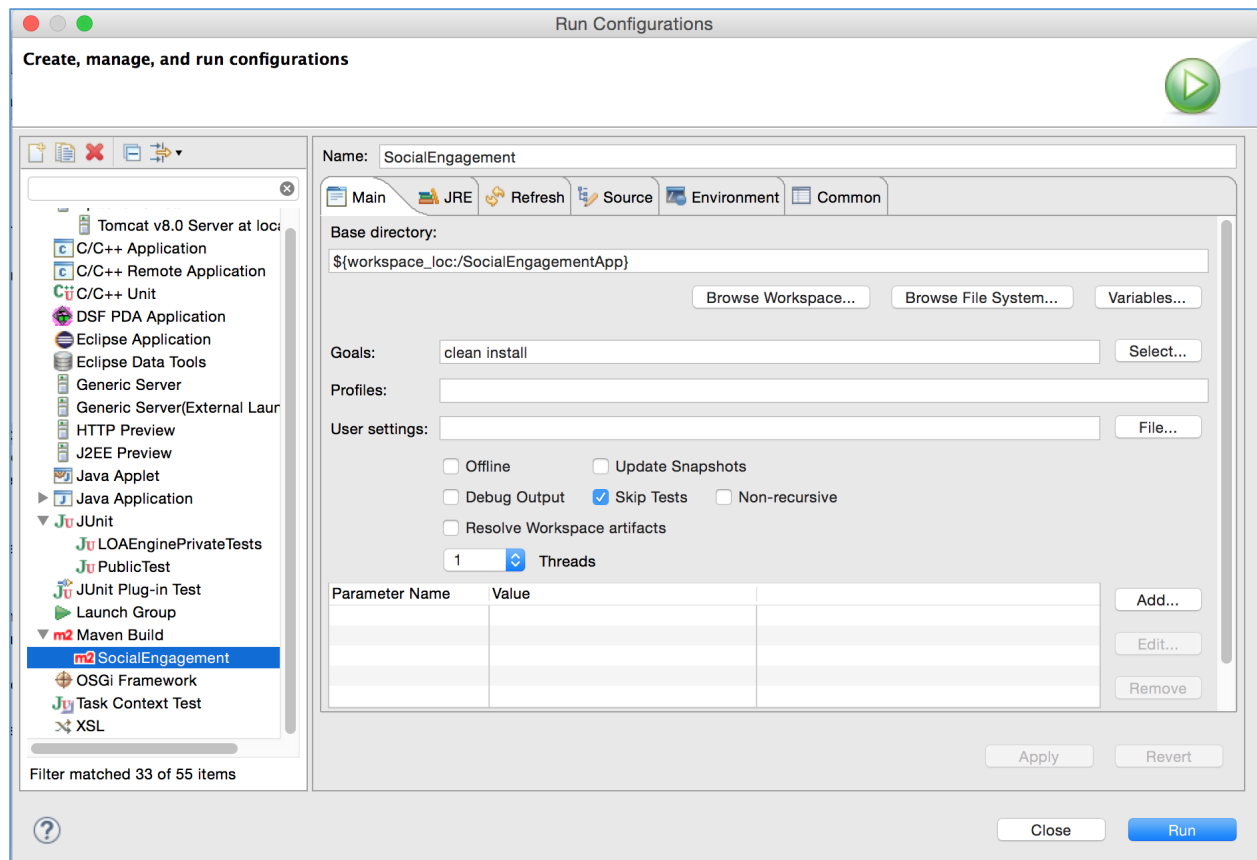
We are using Maven for dependency management so that we won't need to explicitly need to add jar files to CLASSPATH.

We need to deploy the web application as a **.war** file. Steps to deploy are as follows:

1. As this is a maven application, we need to build it using maven.

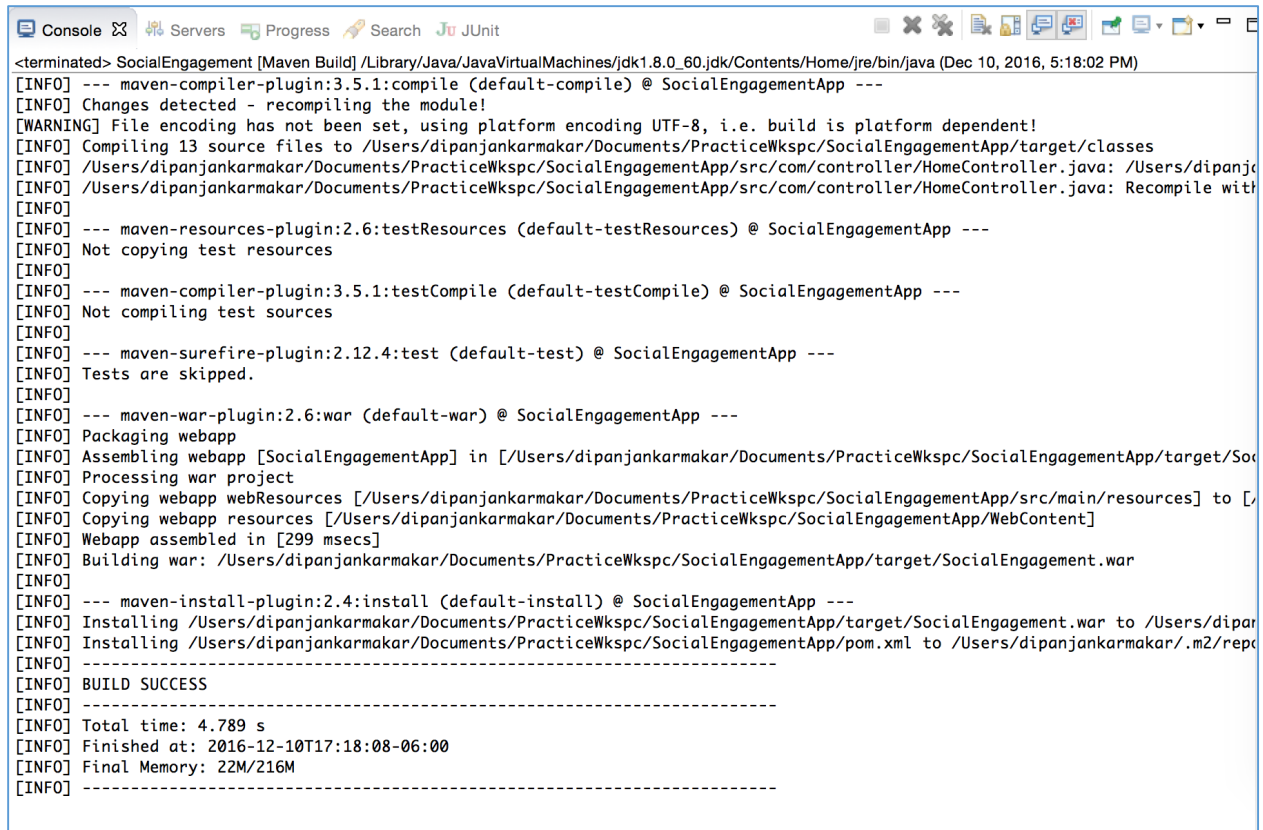


Right click the application in Eclipse and click on **“Run As”**. Then click on **“Run Configuration”** submenu.



Create a new Maven Build configuration and select the project in Base Directory. Enter “clean install” in Goals textbox. Then press the Run button.

This will **build** the project and create a .war file that can be deployed in the Tomcat server.

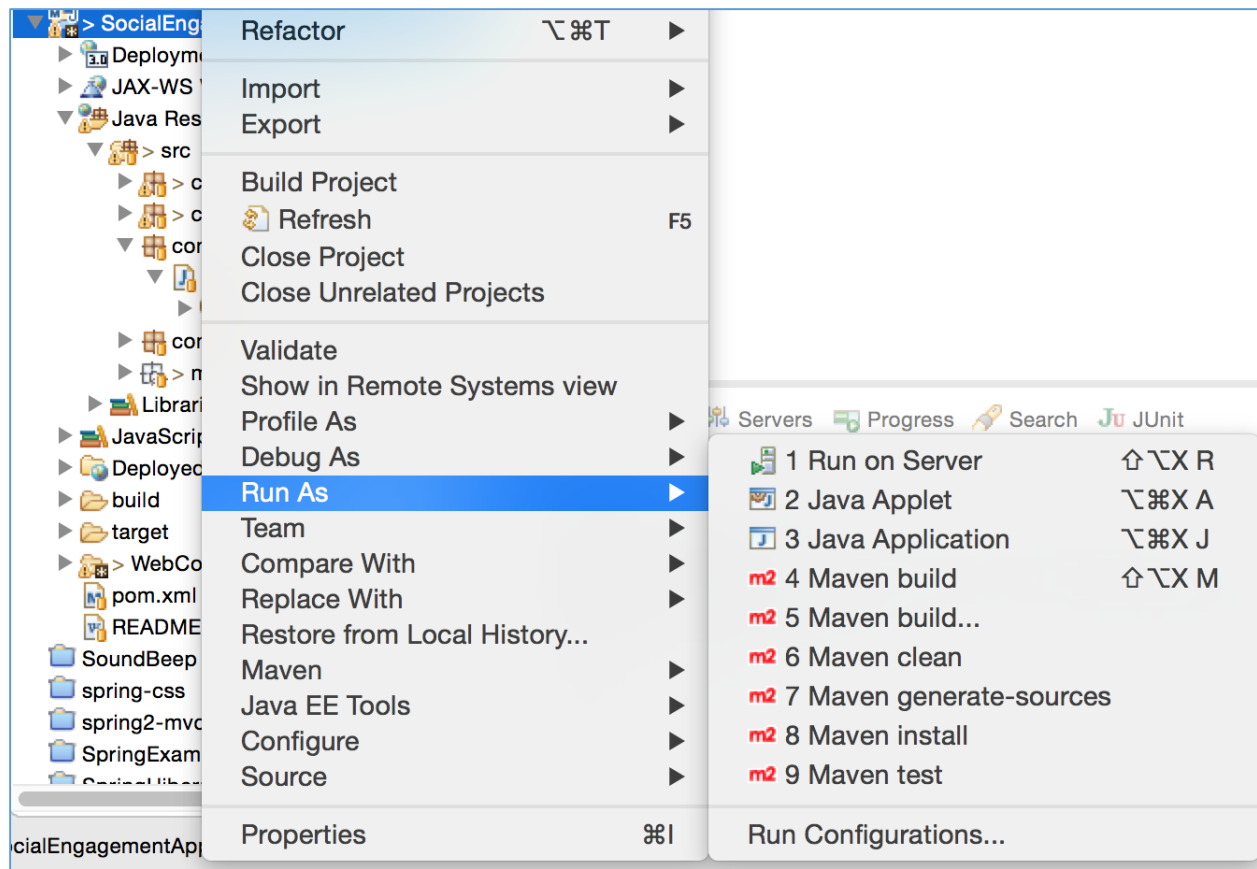


```
<terminated> SocialEngagement [Maven Build] /Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home/jre/bin/java (Dec 10, 2016, 5:18:02 PM)
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ SocialEngagementApp ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 13 source files to /Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/target/classes
[INFO] /Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/src/com/controller/HomeController.java: /Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/src/com/controller/HomeController.java: Recompile with -Xlint:unchecked for details.
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ SocialEngagementApp ---
[INFO] Not copying test resources
[INFO] --- maven-compiler-plugin:3.5.1:testCompile (default-testCompile) @ SocialEngagementApp ---
[INFO] Not compiling test sources
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ SocialEngagementApp ---
[INFO] Tests are skipped.
[INFO] --- maven-war-plugin:2.6:war (default-war) @ SocialEngagementApp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [SocialEngagementApp] in [/Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/target/SocialEngagementApp]
[INFO] Processing war project
[INFO] Copying webapp webResources [/Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/src/main/resources] to [/Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/target/SocialEngagementApp/WEB-INF/classes]
[INFO] Copying webapp resources [/Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/WebContent] to [/Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/target/SocialEngagementApp/WEB-INF/classes]
[INFO] Webapp assembled in [299 msecs]
[INFO] Building war: /Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/target/SocialEngagementApp.war
[INFO] --- maven-install-plugin:2.4:install (default-install) @ SocialEngagementApp ---
[INFO] Installing /Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/target/SocialEngagementApp.war to /Users/dipanjankarmakar/.m2/repository/com/example/SocialEngagementApp/1.0.0/SocialEngagementApp-1.0.0.war
[INFO] Installing /Users/dipanjankarmakar/Documents/PracticeWkspc/SocialEngagementApp/pom.xml to /Users/dipanjankarmakar/.m2/repository/com/example/SocialEngagementApp/1.0.0/SocialEngagementApp-1.0.0.pom
[INFO] BUILD SUCCESS
[INFO] Total time: 4.789 s
[INFO] Finished at: 2016-12-10T17:18:08-06:00
[INFO] Final Memory: 22M/216M
```

If your project is successfully build, you will get an output like this.

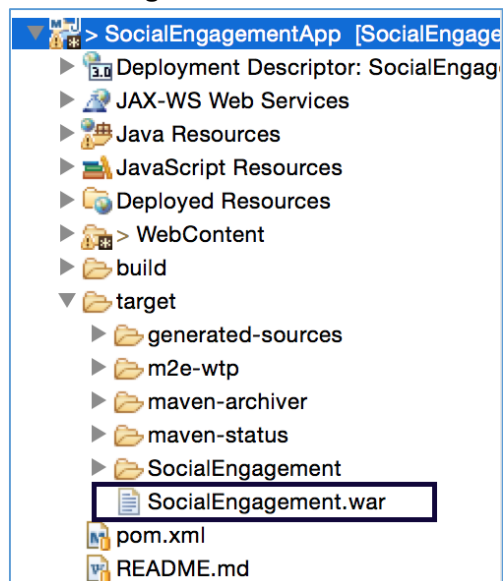
2. Now we shall see how to run this application.

We have assumed that you already had configured Tomcat in your Eclipse environment, if not then please click on this link <http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.stardust.docs.wst%2Fhtml%2Fwst-integration%2Fconfiguration.html>. Once Tomcat is configured then just Right Click the project and click on Run as server. You can select the Tomcat server of your Eclipse workspace.



This will run the application within the Eclipse.

Let us now see how to deploy the application within the ISU server. When the application is build, a war file is created inside the target folder.



Copy this file to the folder **/var/lib/tomcat** in the server. Restart the tomcat server using command **"sudo service tomcat restart"**.

This war file will be explored and a folder with the same name will be created there.

Note: We need to change the permission of this folder so that it can access the database properly. So, run the command **"chmod -R 777 <folder name>"**. Now restart the Tomcat server once more. You can access the project at:

< server – url >: 8080: < folder – name >
-- proj-514-02.cs.iastate.edu:8080/SocialEngagement

please note that 8080 is the default port of tomcat.

In case this port is not opened you need to open this port using firewall-cmd. The application should be running.

Database credentials

We tried to take full advantage of the Spring framework and saved the database credentials in the *dbDetails.properties* file.

Benefit: In case we need to change this credentials, we just modify this file and restart the server directly. We do not need to build the project and deploy.

In case the application is already running, go the folder:

/webapps/<folder-name>/WEB-INF/classes

Example:

```
[dipanjan@proj-514-02 classes]$ ls -lrt
total 12
-rw-rw-r-- 1 tomcat tomcat 1283 Dec  3 15:35 log4j.properties
-rw-rw-r-- 1 tomcat tomcat  637 Dec  3 15:35 dbDetails.properties
drwxrwsr-x 6 tomcat tomcat  58 Dec  3 15:37 com
[dipanjan@proj-514-02 classes]$ pwd
/var/lib/tomcat/webapps/SocialEngagement/WEB-INF/classes
```

In this folder you can see this file. Just change it and restart the server. No deployment needed.

Note: When running this application from outside the server, like in Eclipse, the database string can be:

jdbc:mysql://proj-514-02.cs.iastate.edu:3306/socialDb?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC

However, when you need to deploy this application to the server, this should be changed to:

jdbc:mysql://localhost:3306/socialDb?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC

The reason for this is that the database username that we are using *'coms514user'* is not allowed to access the Sql database using the fully qualified url. It needs to be changed to localhost.

Resolution can be to take help from SSG team and try providing required permission to this username.

Log files

Log files are generated in the folder `/var/log/tomcat`. In this folder you will be able to see all the log files generated.

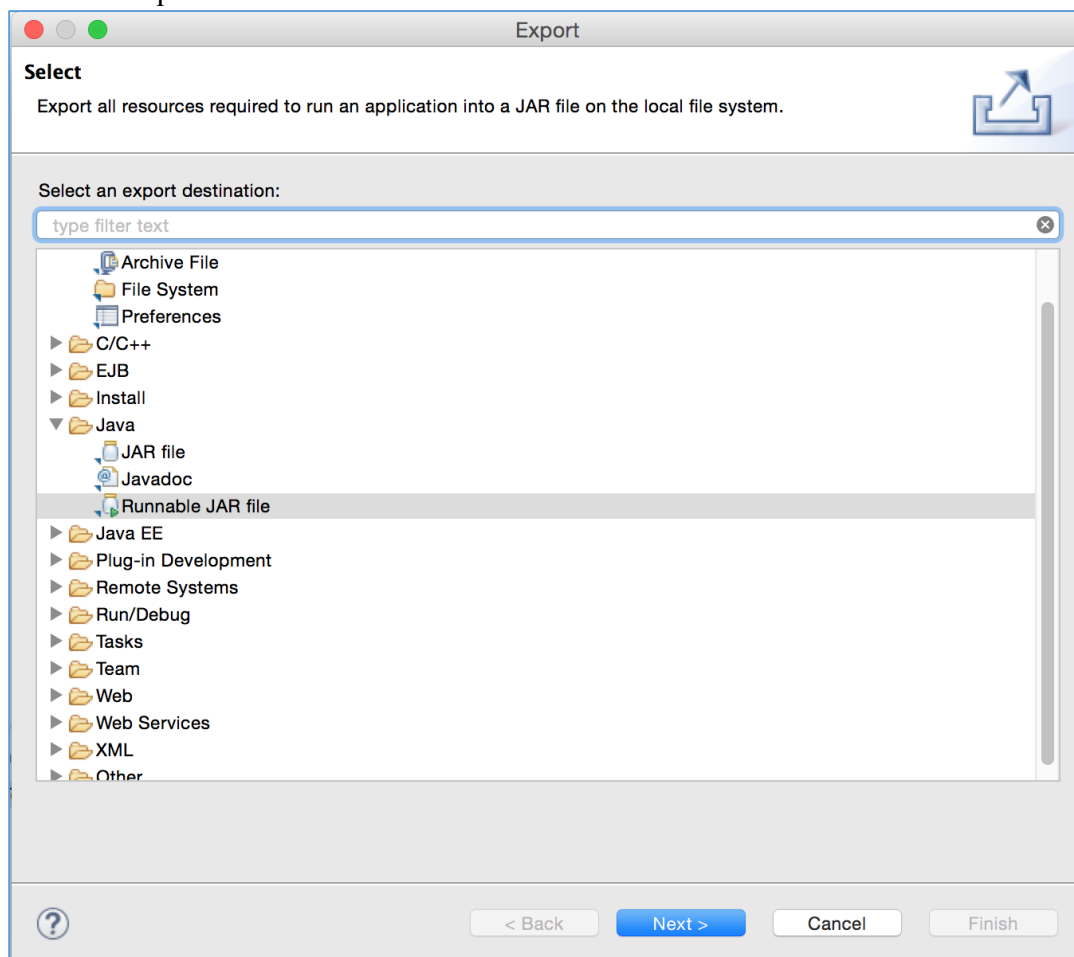
Moreover, our project using *log4j* and we have generated our own logs in the '*debuglog.out*' file in this folder. The configuration of *log4j* can be found in the file *log4j.properties* in the 'main.resources' package in the application. We can change it if necessary.

Sms Notification module

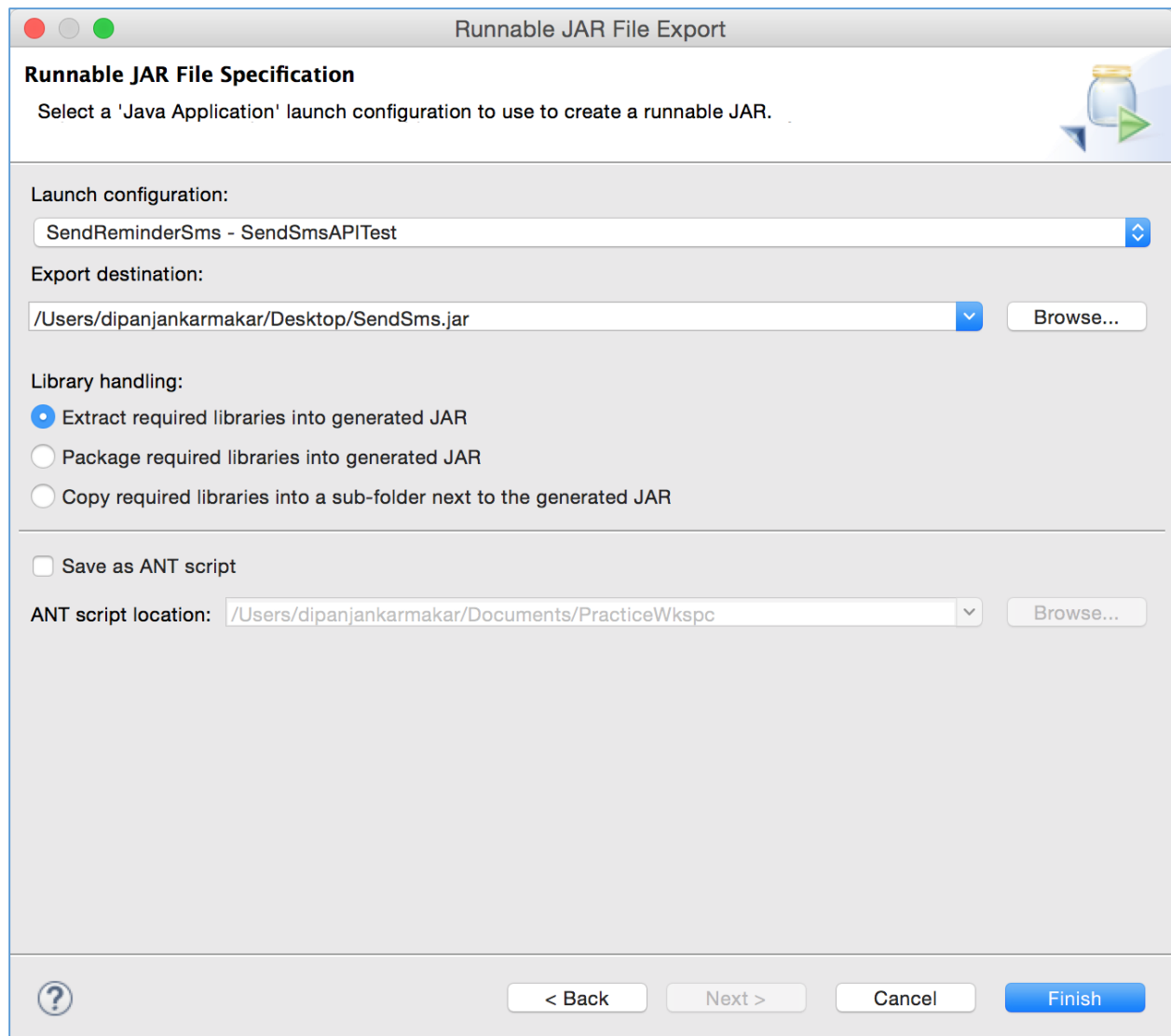
The Sms notification module is just a Java project. We can right click the project and zip it as an executable jar file.

Steps:

Right Click → Export → Runnable JAR file.



Select the launch configuration as the `SendSmsAPITest` and select an Export destination. Do select the option "Export required libraries into generated JAR"



After the JAR is exported, save it to some location in the Server. We have saved it in: /home/dipanjan/

```
[root@proj-514-02 ~]# cd /home/dipanjan/
[root@proj-514-02 dipanjan]# ls
cronOutput.txt  file.txt  SendSms.jar  WriteFile.jar
[root@proj-514-02 dipanjan]#
```

After that we have to trigger this jar using cron. We update the crontab using the “crontab -e” command. We add the line to it:

```
*/* * * * * java -jar /home/dipanjan/SendSms.jar /home/dipanjan/file.txt >> /home/dipanjan/cronOutput.txt 2>&1
```

The cron is set to run every 2 minutes. However, it can easily be updated to run as necessary.


```
# created by dipanjan karmakar for scheduling automated notifications
# */10 * * * * java -cp /home/dipanjan/WriteFile.jar test.write.WriteToFile /home/dipanjan/file.txt >> /home/dipanjan/cronOutput.txt
2>&1
* /2 * * * * java -jar /home/dipanjan/SendSms.jar /home/dipanjan/file.txt >> /home/dipanjan/cronOutput.txt 2>&1
```

The code is checked in the “smsnotification” branch in git.

Project has been deployed to:

<http://proj-514-02.cs.iastate.edu:8080/SocialEngagement/hello>