

## HW3 – Supplementary Guide

### How to import the jar files into your Android Studio project:

First thing you need to do is copy them into your project's library folder located here:

`<project_root>/app/libs`

Now, in Android Studio, open up the *build.gradle* file located in your *app* folder (NOT the *build.gradle* file in your project root). You should see something like this toward the bottom of the file:

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'com.android.support:appcompat-v7:23.0.0'  
}
```

Add the following lines within the *dependencies {}* block so that it looks like this:

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'com.android.support:appcompat-v7:23.0.0'  
    compile files('libs/mail.jar')  
    compile files('libs/activation.jar')  
    compile files('libs/additional.jar')  
}
```

### Using the javamail library and helper class:

The javamail library is a javax library used for sending/receiving email. This was ported to Android, and you will use it for task 3. To make things a bit simpler for you, your super awesome TA wrote a helper class which implements Runnable, meaning that you can use the helper class when constructing a new Thread (you need to run this class in its own thread, since it calls methods which perform network I/O).

Assuming the three jar files for javamail-android are added as libraries to your project, do the following:

- Copy EmailSender.java and SendEmailTask.java to your

`<project_root>/app/src/main/java/<package_name>`

- You will get errors about the package path not being set in both classes. To fix this, either copy/paste the package path (into both of these classes) from another java file in your project, or hover the mouse over the class name, click on it, and click on the red light bulb which appears. Then click 'Set package name to...'

The helper class is SendEmailTask. This handles the setup of an EmailSender object, which will actually send the email.

### Example usage of the helper class:

I wrote the helper class to be super easy :-). Here is an example showing how easy it is to use:

```
SendEmailTask emailTask = new SendEmailTask(  
    "tubularmcaawesome@gmail.com",  
    "<gmail_password>",  
    "tubularmcaawesome@gmail.com");
```

```
emailTask.setEmailRecipient("tague@cmu.edu");
emailTask.setEmailSubject("Awesome HW3 Stuff!!");
emailTask.setEmailBody("This email was sent from an HTC One, without user interaction.\nSent to
    your address to be awesome...");
```

```
Thread emailThread = new Thread(emailTask);
emailThread.start();
```

The constructor for SendEmailTask has three parameters:

- *smtp\_username*: Your username for the SMTP server which will handle your mail. By default EmailSender is configured for Google's gmail SMTP server with auth. Thus, unless you change this, you will need to use a google account (you can use a gmail account, or for CMU SV students, your west.cmu.edu also works). If you have credentials for another SMTP server that you'd like to use, you can modify EmailSender.java accordingly. You can even disable auth if you want to use an SMTP server which allows for this (not recommended).
- *smtp\_password*: This will be your google account password if sticking with the default SMTP server
- *sender\_email\_address*: This is the sender's email address. While you can make this up, note that what you put here won't necessarily end up being the sender. For example, google's SMTP server uses your login credentials to determine the sender, and will treat the specified sender email address as the original sender. However when viewing in gmail, the google-determined sender will show up, not the original sender (it will be listed when you view the email details however). Probably best just to keep this the same as your username.

Here are the setters in SendEmailTask you need to set:

- *setEmailRecipient(String)* - This sets the recipient of your emails. You can use the same email account that you are sending from, or can pick another address. Just make sure you have access to the address which you put :-)
- *setEmailSubject(String)* - This sets the email's subject.
- *setEmailBody(String)* - This sets the body of the email. Put your data here (APs et al).

### Checking email send state:

How will you know if an email sent or not? Afterall, the app is stealth, and the helper class doesn't send back any status to it's calling thread. Never fear, for your TA has your back! The helper class will log, to logcat, the email send results. It will notify in the log when an email is attempting to be sent, if it sent, and if an exception occurred (along with backtrace).

Bombarded by logcat messages? Never fear, for you TA has your back (again). In the logcat window, next to 'log level' you will see a search box. Type the filter term '[PT]' into that box. Now only stuff related to SendEmailTask will show up. Note, this filter term comes from the LOG\_TAG string in SendEmailTask. It may be a good idea to either modify this tag value or use it (the '[PT]' part) in other classes so you can get more info dumped to your logcat window using a common filter term. For those curious what PT stands for, its an abbreviation for the name of the TA's proof-of-concept app for HW3 (nope, it doesn't stand for Patrick Tague).

### A Note on Google accounts

Google now requires by default all apps which connect to gmail using SMTP to authenticate using oauth2. Well we don't wanna do that. To get around this issue, do the following in your google account settings (Login to your google account, click the 'Google Apps' button and select 'My Account'):

- click *Signing in to Google*
- scroll down the settings until you see *Allow less secure apps*.
- click the slider to turn it on

If you don't do this, you'll get a really nasty message from Google when you attempt to authenticate using your tracker app, and you are going to feel really bad about yourself. This way you won't receive such a message, and your 1337 h4x0r status will be preserved.

If you plan to use a temporary email address for this assignment (a throwaway email), it may be a better idea, rather than signing up for a gmail account, to use an email account that doesn't require phone verification to sign up. Yandex allows for such accounts (simply click '*I don't have a phone number*' when signing up). If you use Yandex, make the following code change in *EmailSender.java*, line 48:

change "*smtp.gmail.com*" to "*smtp.yandex.com*"