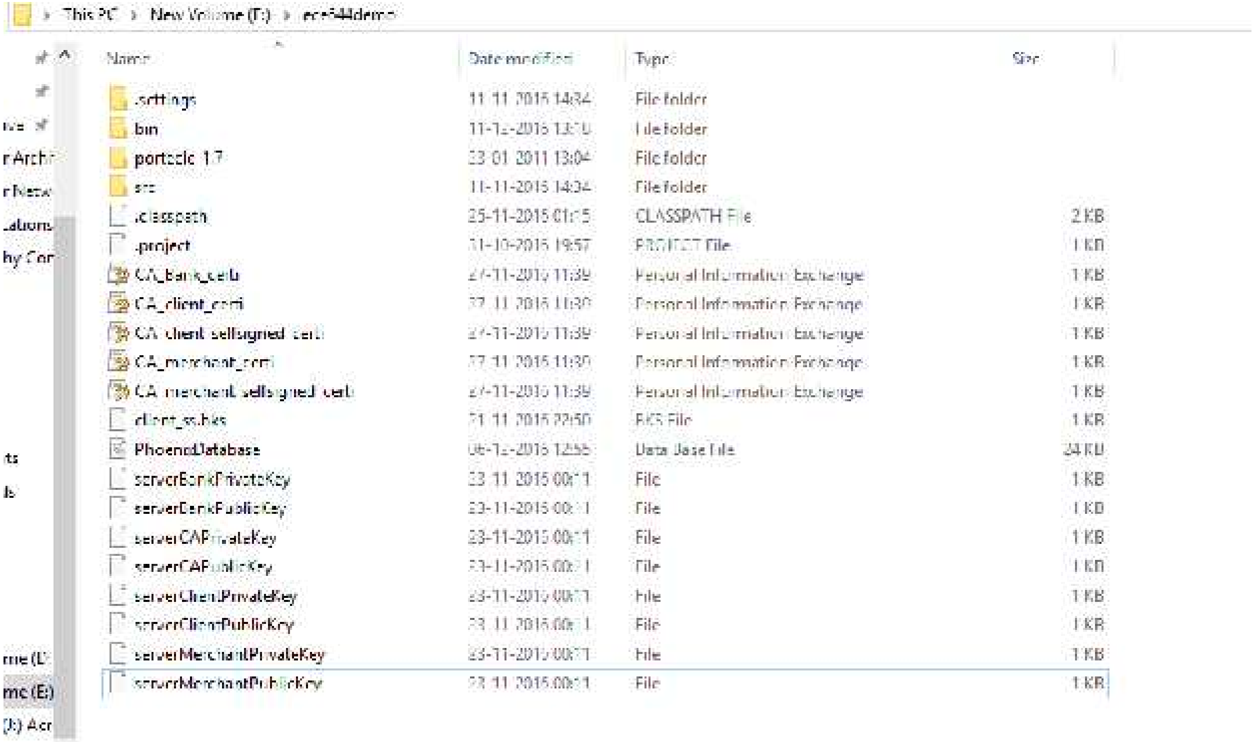
1. Import Phoenix_java in Eclipse.
2. You might get errors due to jars not present.
3. All the jars are present in Phoenix_jars folder. Import them in the project.
4. Open the SendEmail.java file and put your gmail username and password. (Highlighted by comments)
5. Now, we are giving all the key, certificate and database files. This is because we have put the same certificate and the key generated here in the android application.
6. If you want you can create your own key, certificate and database files by running KeyGenerator.java, CertificateAuthorityServer. Java and Database.java respectively in the same order. Then you will have to copy the files in the android application. This is a lot of work and you are bound to face a lot of errors.
7. So, it’s better to use the keys and certificates already generated. But, the code we used to generate them is present in the java files.
8. Next, paste the key, certificate and database files from the folder Phoenix_extra in the directory of your eclipse project. (Your folder should look like this after pasting)

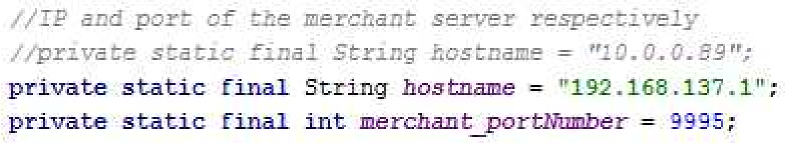


This PC › New Volume (F:) › ece544demo

Name	Date modified	Type	Size
.settings	11-11-2015 14:34	File folder	
bin	11-12-2015 13:10	File folder	
portecle 1.7	23-01-2011 13:04	File folder	
src	11-11-2015 14:34	File folder	
.classpath	25-11-2015 01:15	CLASSPATH File	2 KB
.project	31-10-2015 19:57	PROJECT File	1 KB
CA_Bank_certi	27-11-2015 11:39	Personal Information Exchange	1 KB
CA_client_certi	27-11-2015 11:39	Personal Information Exchange	1 KB
CA_client_selfsigned_certi	27-11-2015 11:39	Personal Information Exchange	1 KB
CA_merchant_certi	27-11-2015 11:39	Personal Information Exchange	1 KB
CA_merchant_selfsigned_certi	27-11-2015 11:39	Personal Information Exchange	1 KB
client_ss.bks	21-11-2015 22:50	BKS File	1 KB
PhoenixDatabase	06-12-2015 12:55	Data Base File	24 KB
serverBankPrivateKey	23-11-2015 00:11	File	1 KB
serverBankPublicKey	23-11-2015 00:11	File	1 KB
serverCAPrivateKey	23-11-2015 00:11	File	1 KB
serverCAPublicKey	23-11-2015 00:11	File	1 KB
serverClientPrivateKey	23-11-2015 00:11	File	1 KB
serverClientPublicKey	23-11-2015 00:11	File	1 KB
serverMerchantPrivateKey	23-11-2015 00:11	File	1 KB
serverMerchantPublicKey	23-11-2015 00:11	File	1 KB

9. The SendSMS. Java file is responsible to send SMS to the phone. Since we are not a using premium account of Twilio, we can send SMS to only one number. That is why the numbers are hard-coded in the file. So, the OTP will be sent only to this number.
10. You need to create an ad-hoc network for connecting android and Java. Follow the instruction given at - http://www.redmondpie.com/create-ad-hoc-wifi-hotspot-in-windows-10-heres-how/
11. Connect your android phone to this network. Check the IP of your system in this network. It should be 192.168.137.1. If it is the same, then just install the apk-debug.apk from the folder Phoenix_apk in your phone. (You might need to change the setting of installing from unknown

sources). If this is not the IP, then you have to import the Phoenix folder (application) in Android Studio and change the IP in all the files.



```
//IP and port of the merchant server respectively
//private static final String hostname = "10.0.0.89";
private static final String hostname = "192.168.137.1";
private static final int merchant_portNumber = 9995;
```

12. Now, run the files MerchantServer.java and BankServer.java.
13. Open the android application and start with login/registration.
14. You can see the console for outputs.
15. When you reach the OTP stage, enter the OTP from the BankServer.java console. You won't get this on your mobile phone due to the limitation mentioned earlier.
16. After entering the OTP, you should receive an email for a successful/unsuccessful transaction . You may have to change the settings in Gmail to connect to unauthorized apps. (You will get a notification for this in your mail when you reach the email stage. After you give the permissions, try running the application again to verify that you are getting the mail)
17. If you want to check the database file, you can do it using DB Browser for SQLite. (http://sqlitebrowser.org/)