網路程式設計-期末作業

Outline

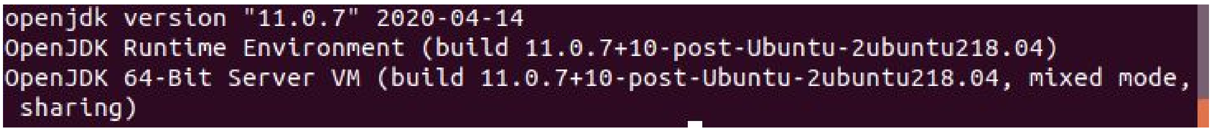
- 環境設定(ubuntu)
- 程式架構
- 使用方式與功能
- 心得

環境設定(ubuntu)

1. 安裝JAVA

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install default-jre
$ sudo apt-get install default-jdk
```

此次安裝版本為OpenJDK 11



```
openjdk version "11.0.7" 2020-04-14
OpenJDK Runtime Environment (build 11.0.7+10-post-Ubuntu-2ubuntu218.04)
OpenJDK 64-Bit Server VM (build 11.0.7+10-post-Ubuntu-2ubuntu218.04, mixed mode,
 sharing)
```

設定環境變數

```
$ vim ~/.bashrc
```

加入

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

2. 安裝apache-tomcat

去官往下載 apache-tomcat， 此次使用版本為 `v9.0.36`。
將下載的tar.gz解壓縮並移動到`/usr/local/`

```
$ tar zxvf apache-tomcat-9.0.14.tar.gz
$ mv apache-tomcat-9.0.14 apache-tomcat
$ mv apache-tomcat /usr/local/
$ sudo chmod -R 777 /usr/local/apache-tomcat
```

設定環境變數

```
$ vim ~/.bashrc
```

加入

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export TOMCAT_HOME=/usr/local/apache-tomcat
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:${TOMCAT_HOME}/lib/servlet-api.jar
export PATH=.:${JAVA_HOME}/bin:$PATH
```

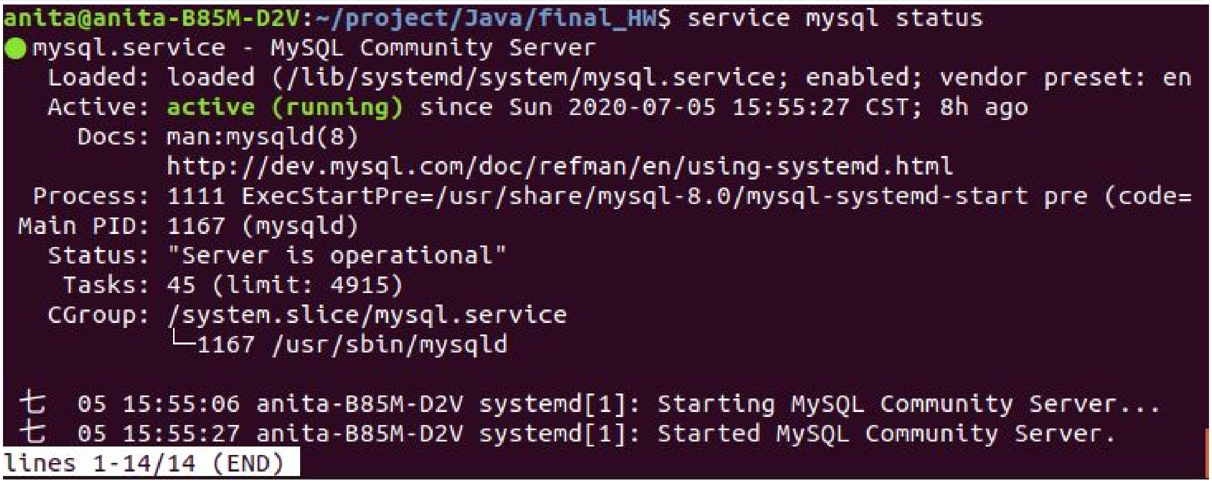
3. 安裝MySQL

下載 MySQL 並安裝

```
$ curl -OL https://dev.mysql.com/get/mysql-apt-config_0.8.12-1_all.deb
$ sudo dpkg -i mysql-apt-config*
$ sudo apt-get update
$ sudo apt-get install mysql-server
```

查看是否 MySQL 已順利運行

```
$ service mysql status
```



```
anita@anita-B85M-D2V:~/project/Java/final_HW$ service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Sun 2020-07-05 15:55:27 CST; 8h ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 1111 ExecStartPre=/usr/share/mysql-8.0/mysql-systemd-start pre (code=
 Main PID: 1167 (mysqld)
   Status: "Server is operational"
    Tasks: 45 (limit: 4915)
   CGroup: /system.slice/mysql.service
           └─1167 /usr/sbin/mysqld

 七  05 15:55:06 anita-B85M-D2V systemd[1]: Starting MySQL Community Server...
 七  05 15:55:27 anita-B85M-D2V systemd[1]: Started MySQL Community Server.
lines 1-14/14 (END)
```

設定root密碼

```
$ mysql -u root
$ mysql> use mysql;
$ mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY '123456';
$ mysql> FLUSH PRIVILEGES;
```

4. 安裝JBDC driver

```
$ sudo apt-get install libmysql-java
```

設定環境變數

```
$ vim ~/.bashrc
```

將CLASSPATH修改成

```
export
CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:${TOMCAT_HOME}/lib/servlet-api.jar:/usr/share/java/mysql.jar
```

將`mysql.jar`複製一份到 `/usr/local/apache-tomcat/lib` 資料夾下，不然伺服器會不認得JBDC driver

5. 設定Unirest-Java

程式會用到Unirest套件，其套件相關檔案在`/jar_files`裡。
將該資料夾指到`CLASSPATH`路徑

```
$ vim ~/.bashrc
```

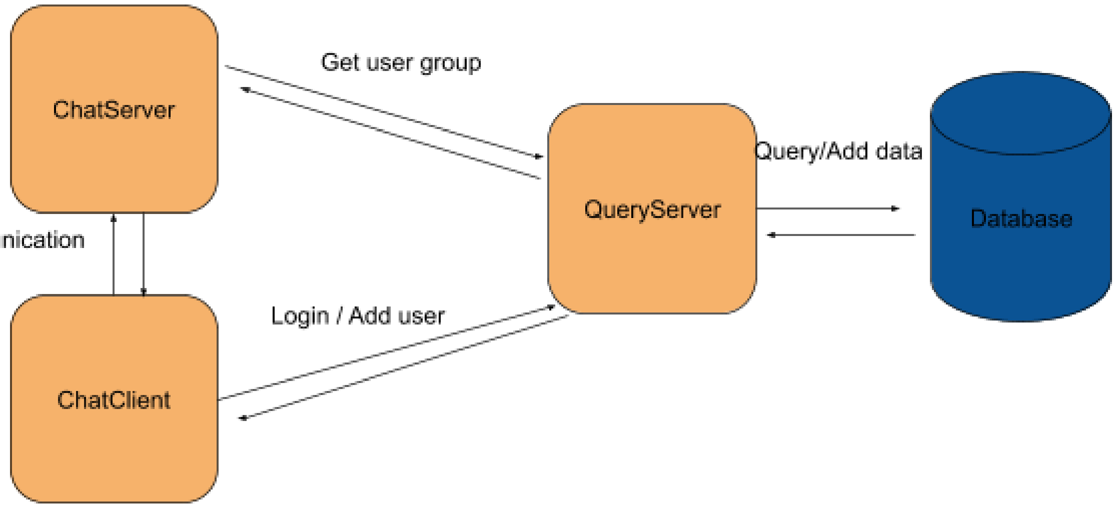
```
export
CLASSPATH=.:${JAVA_HOME}/lib:${TOMCAT_HOME}/lib/servlet-api.jar:/usr/share/java/mysql.jar:/home/anita/project/Java/final_HW/jar_files/*
```

程式架構

架構

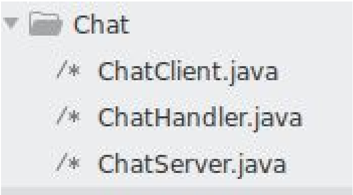
Client 先登入QueryServer成功後再與ChatSevrer連線。

ChatSevrer 在Client 連線後，再去與ChatSevrer取user group, 再將得到的group回傳給 Client 。



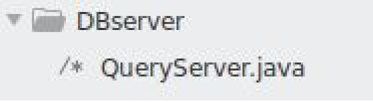
檔案資料夾說明

Chat 資料夾負責處理 ChatClient 與ChatSevrer通訊，與ChatClient , QuerySevrer 的登入/新增使用者部份。



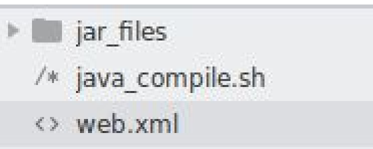
```
▾ Chat
    /* ChatClient.java
    /* ChatHandler.java
    /* ChatServer.java
```

DBserver資料夾負責處理 QueryServer與資料庫的資料新增查詢。



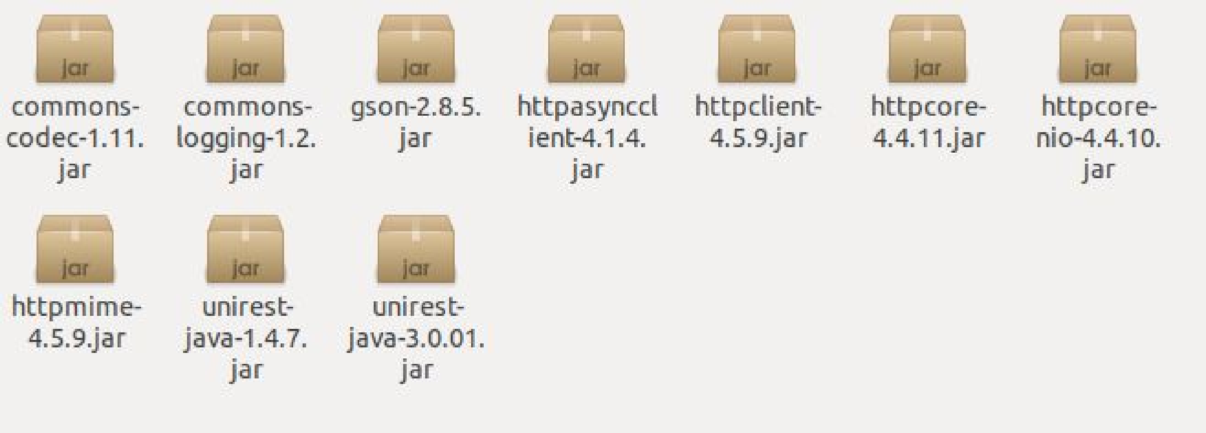
```
▾ DBserver
    /* QueryServer.java
```

剩下檔案



```
▸ jar_files
  /* java_compile.sh
  <> web.xml
```

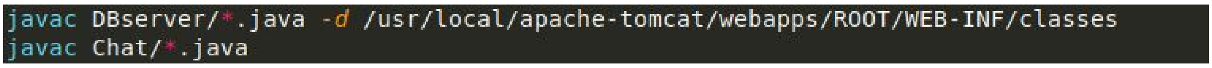
jar_files為使用到的.jar lib檔案



commons-codec-1.11.jar
commons-logging-1.2.jar
gson-2.8.5.jar
httpasyncclient-4.1.4.jar
httpclient-4.5.9.jar
httpcore-4.4.11.jar
httpcore-nio-4.4.10.jar
httpmime-4.5.9.jar
unirest-java-1.4.7.jar
unirest-java-3.0.01.jar

程式說明

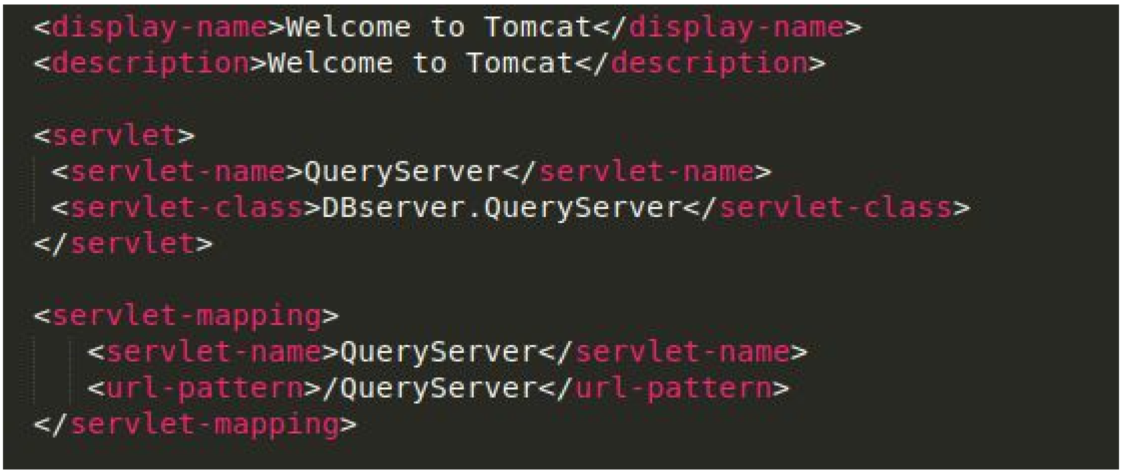
java_compile.sh 用來compile 所有的java檔案



```
javac DBserver/*.java -d /usr/local/apache-tomcat/webapps/ROOT/WEB-INF/classes
javac Chat/*.java
```

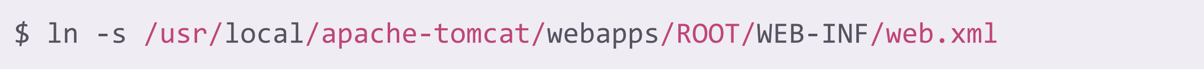
第一行會將compile的.class檔案移到tomcat指定資料夾裡
第二行則是compile Chat/ 資料夾裡的檔案

web.xml 為設定QueryServer的路徑與路口點



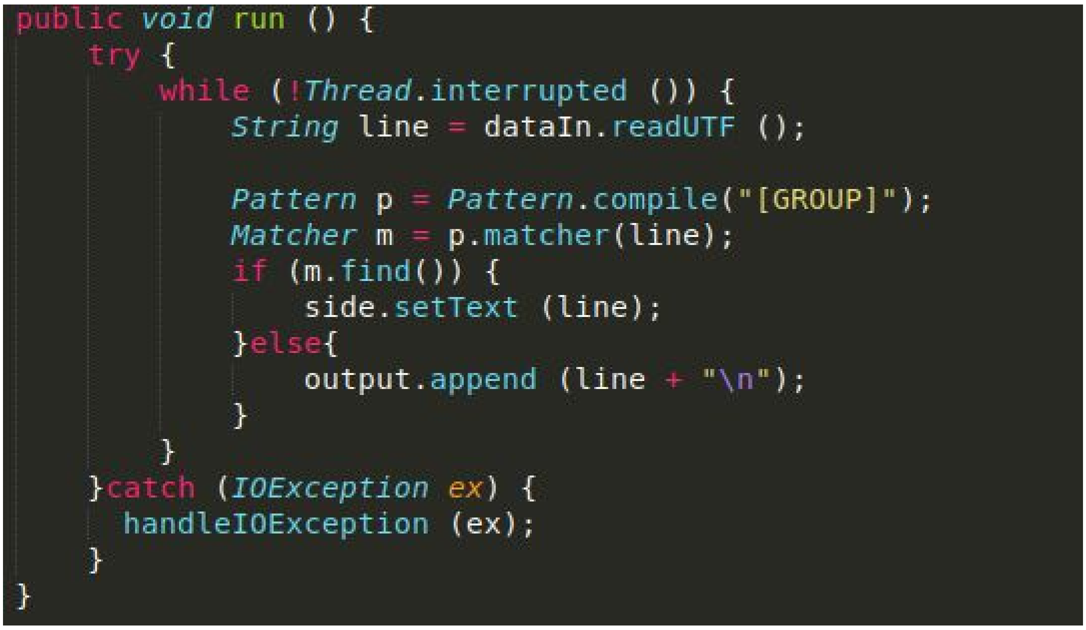
```
<display-name>Welcome to Tomcat</display-name>
<description>Welcome to Tomcat</description>

<servlet>
 <servlet-name>QueryServer</servlet-name>
 <servlet-class>DBserver.QueryServer</servlet-class>
</servlet>

<servlet-mapping>
   <servlet-name>QueryServer</servlet-name>
   <url-pattern>/QueryServer</url-pattern>
</servlet-mapping>
```

將tomcat資料下的web.xml 指到開發資料夾下



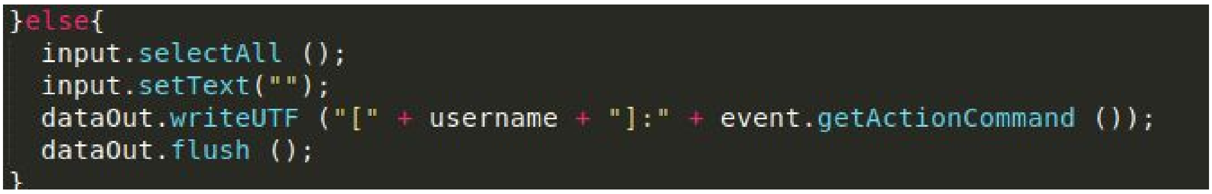
```
$ ln -s /usr/local/apache-tomcat/webapps/ROOT/WEB-INF/web.xml
```

以下節錄ChatClient.java, ChatServer.java 通訊部份
ChatClient.java 接收訊息



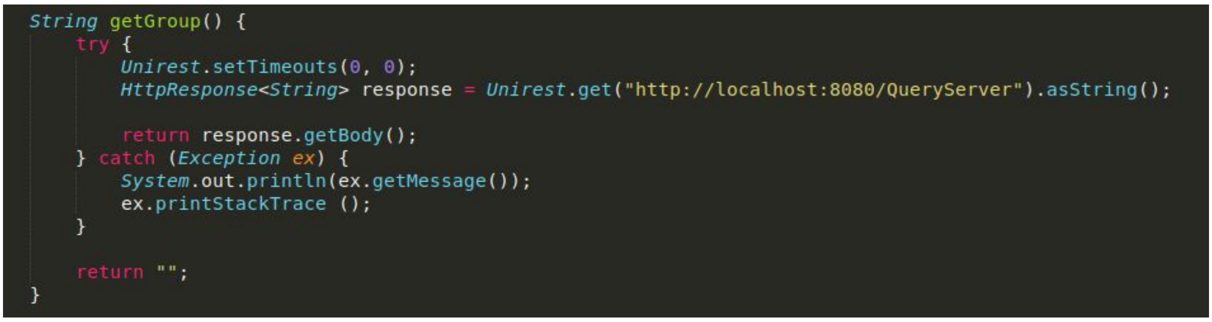
```java
public void run () {
    try {
        while (!Thread.interrupted ()) {
            String line = dataIn.readUTF ();

            Pattern p = Pattern.compile("[GROUP]");
            Matcher m = p.matcher(line);
            if (m.find()) {
                side.setText (line);
            }else{
                output.append (line + "\n");
            }
        }
    }catch (IOException ex) {
      handleIOException (ex);
    }
}
```

ChatClient.java 傳送訊息



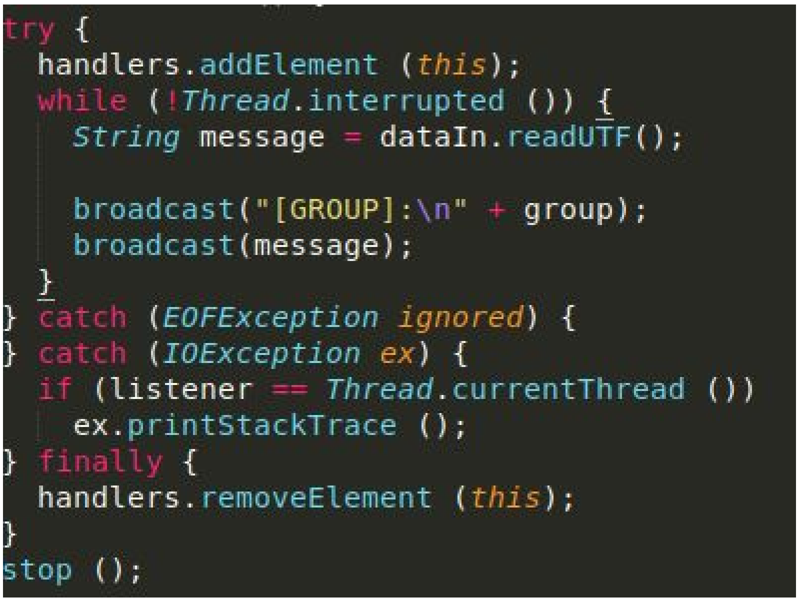
```java
}else{
  input.selectAll ();
  input.setText("");
  dataOut.writeUTF ("[" + username + "]:" + event.getActionCommand ());
  dataOut.flush ();
}
```

ChatServer.java 跟QueryServer取得group list部份



```java
String getGroup() {
    try {
        Unirest.setTimeouts(0, 0);
        HttpResponse<String> response = Unirest.get("http://localhost:8080/QueryServer").asString();

        return response.getBody();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
        ex.printStackTrace ();
    }

    return "";
}
```

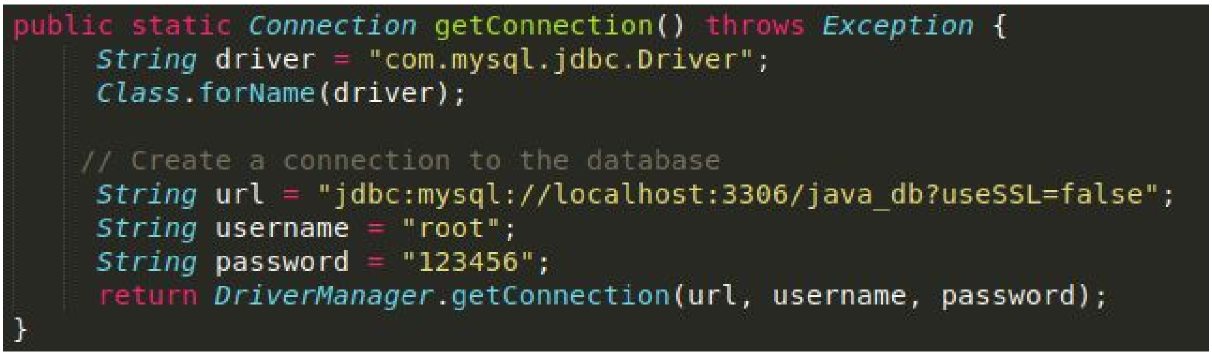
ChatHandler.java 傳送group list與其他Client訊息部份



```
try {
  handlers.addElement (this);
  while (!Thread.interrupted ()) {
    String message = dataIn.readUTF();

    broadcast("[GROUP]:\n" + group);
    broadcast(message);
  }
} catch (EOFException ignored) {
} catch (IOException ex) {
  if (listener == Thread.currentThread ())
    ex.printStackTrace ();
} finally {
  handlers.removeElement (this);
}
stop ();
```

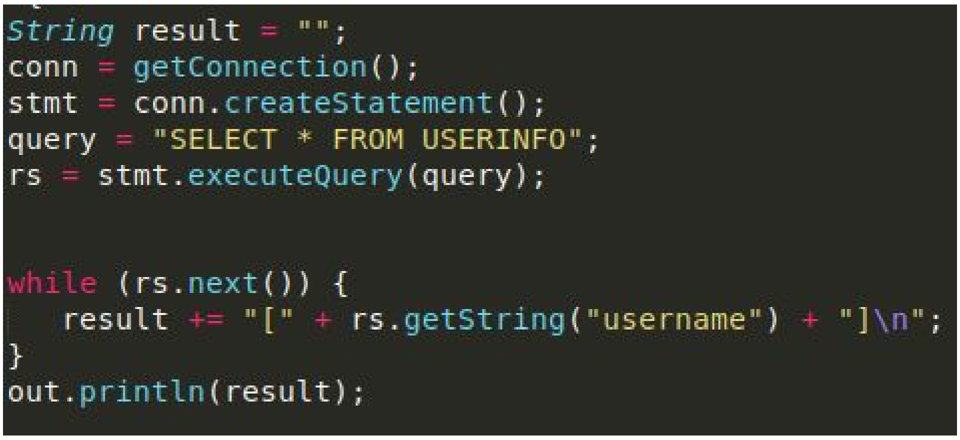
以下節錄QueryServer.java

資料庫連線部份



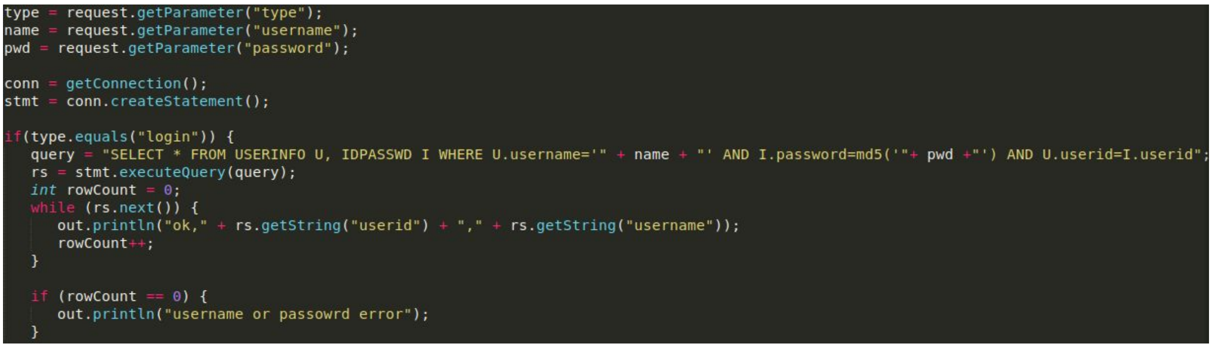
```
public static Connection getConnection() throws Exception {
      String driver = "com.mysql.jdbc.Driver";
      Class.forName(driver);

     // Create a connection to the database
      String url = "jdbc:mysql://localhost:3306/java_db?useSSL=false";
      String username = "root";
      String password = "123456";
      return DriverManager.getConnection(url, username, password);
}
```

Get group list 部份



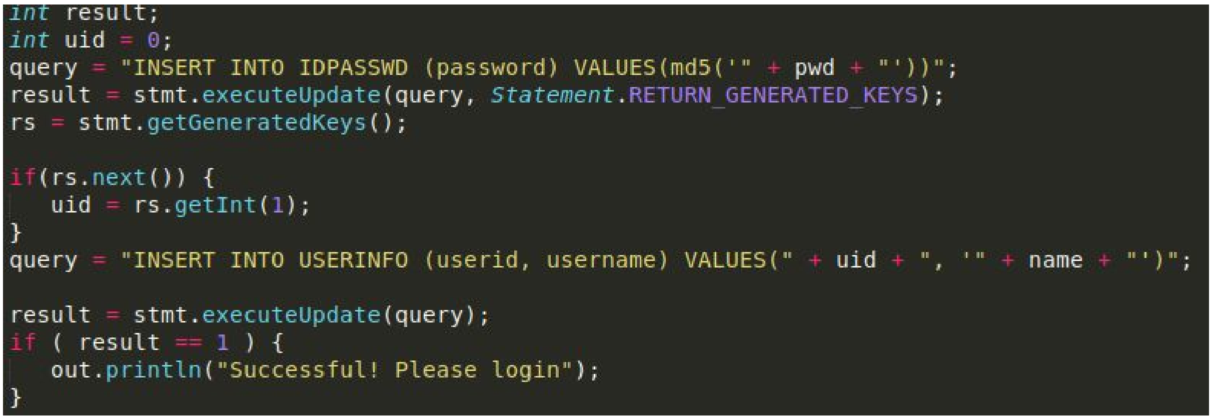
```
String result = "";
conn = getConnection();
stmt = conn.createStatement();
query = "SELECT * FROM USERINFO";
rs = stmt.executeQuery(query);


while (rs.next()) {
   result += "[" + rs.getString("username") + "]\n";
}
out.println(result);
```

Login 部份



```
type = request.getParameter("type");
name = request.getParameter("username");
pwd = request.getParameter("password");

conn = getConnection();
stmt = conn.createStatement();

if(type.equals("login")) {
   query = "SELECT * FROM USERINFO U, IDPASSWD I WHERE U.username='" + name + "' AND I.password=md5('"+ pwd +"') AND U.userid=I.userid";
   rs = stmt.executeQuery(query);
   int rowCount = 0;
   while (rs.next()) {
      out.println("ok," + rs.getString("userid") + "," + rs.getString("username"));
      rowCount++;
   }

   if (rowCount == 0) {
      out.println("username or passowrd error");
   }
```

Add User 部份

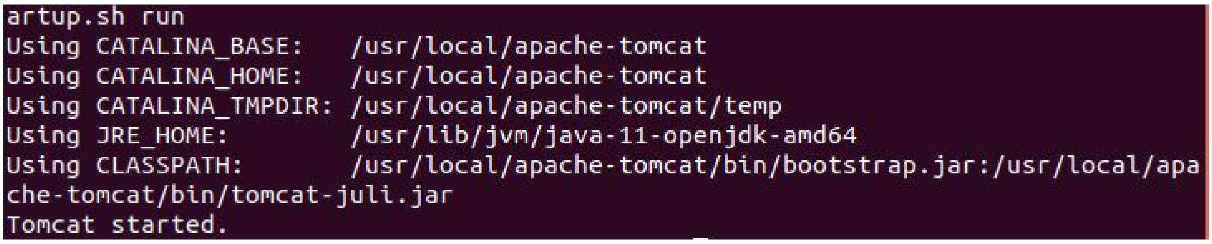


```
int result;
int uid = 0;
query = "INSERT INTO IDPASSWD (password) VALUES(md5('" + pwd + "'))";
result = stmt.executeUpdate(query, Statement.RETURN_GENERATED_KEYS);
rs = stmt.getGeneratedKeys();

if(rs.next()) {
   uid = rs.getInt(1);
}
query = "INSERT INTO USERINFO (userid, username) VALUES(" + uid + ", '" + name + "')";

result = stmt.executeUpdate(query);
if ( result == 1 ) {
   out.println("Successful! Please login");
}
```

使用方式與功能

1. 開啟tomcat

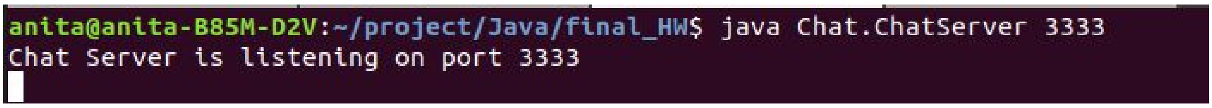
```
$ sh /usr/local/apache-tomcat/bin/startup.sh run
```



```
artup.sh run
Using CATALINA_BASE:   /usr/local/apache-tomcat
Using CATALINA_HOME:   /usr/local/apache-tomcat
Using CATALINA_TMPDIR: /usr/local/apache-tomcat/temp
Using JRE_HOME:        /usr/lib/jvm/java-11-openjdk-amd64
Using CLASSPATH:       /usr/local/apache-tomcat/bin/bootstrap.jar:/usr/local/apa
che-tomcat/bin/tomcat-juli.jar
Tomcat started.
```

2. 開啟ChatServer

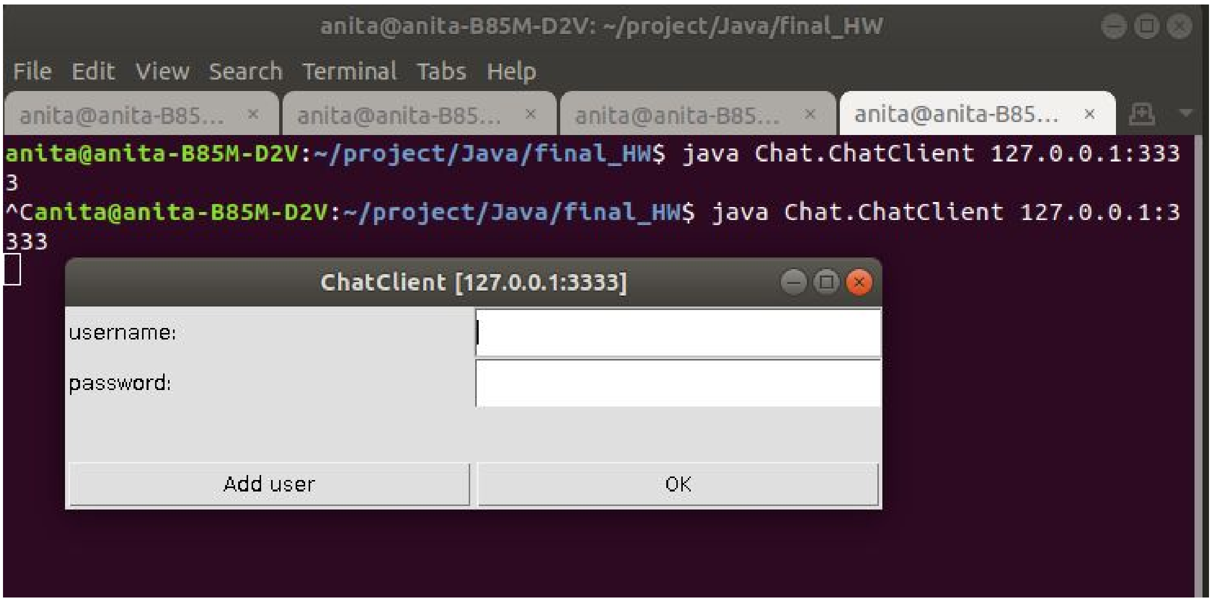
```
$ java Chat.ChatServer 3333
```



```
anita@anita-B85M-D2V:~/project/Java/final_HW$ java Chat.ChatServer 3333
Chat Server is listening on port 3333
```

3. 開啟ChatClient

```
$ java Chat.ChatClient 127.0.0.1:3333
```



anita@anita-B85M-D2V: ~/project/Java/final_HW
File Edit View Search Terminal Tabs Help
anita@anita-B85... ×
anita@anita-B85... ×
anita@anita-B85... ×
anita@anita-B85... ×

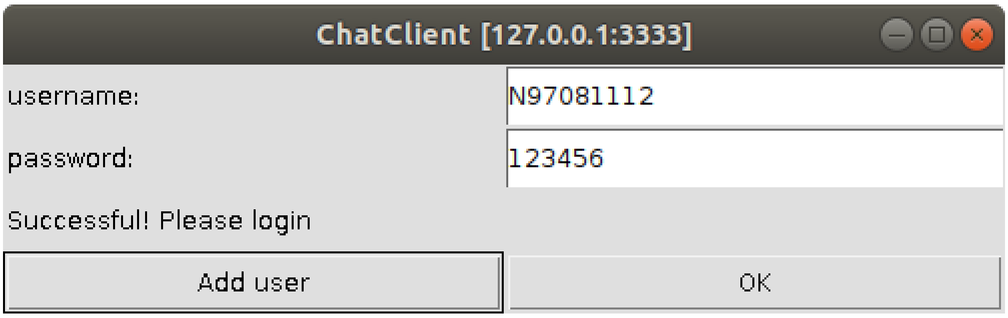
```
anita@anita-B85M-D2V:~/project/Java/final_HW$ java Chat.ChatClient 127.0.0.1:333
3
^Canita@anita-B85M-D2V:~/project/Java/final_HW$ java Chat.ChatClient 127.0.0.1:3
333
```

ChatClient [127.0.0.1:3333]

username:

password:

輸入username, password 點擊”Add user”新增新用戶



ChatClient [127.0.0.1:3333]

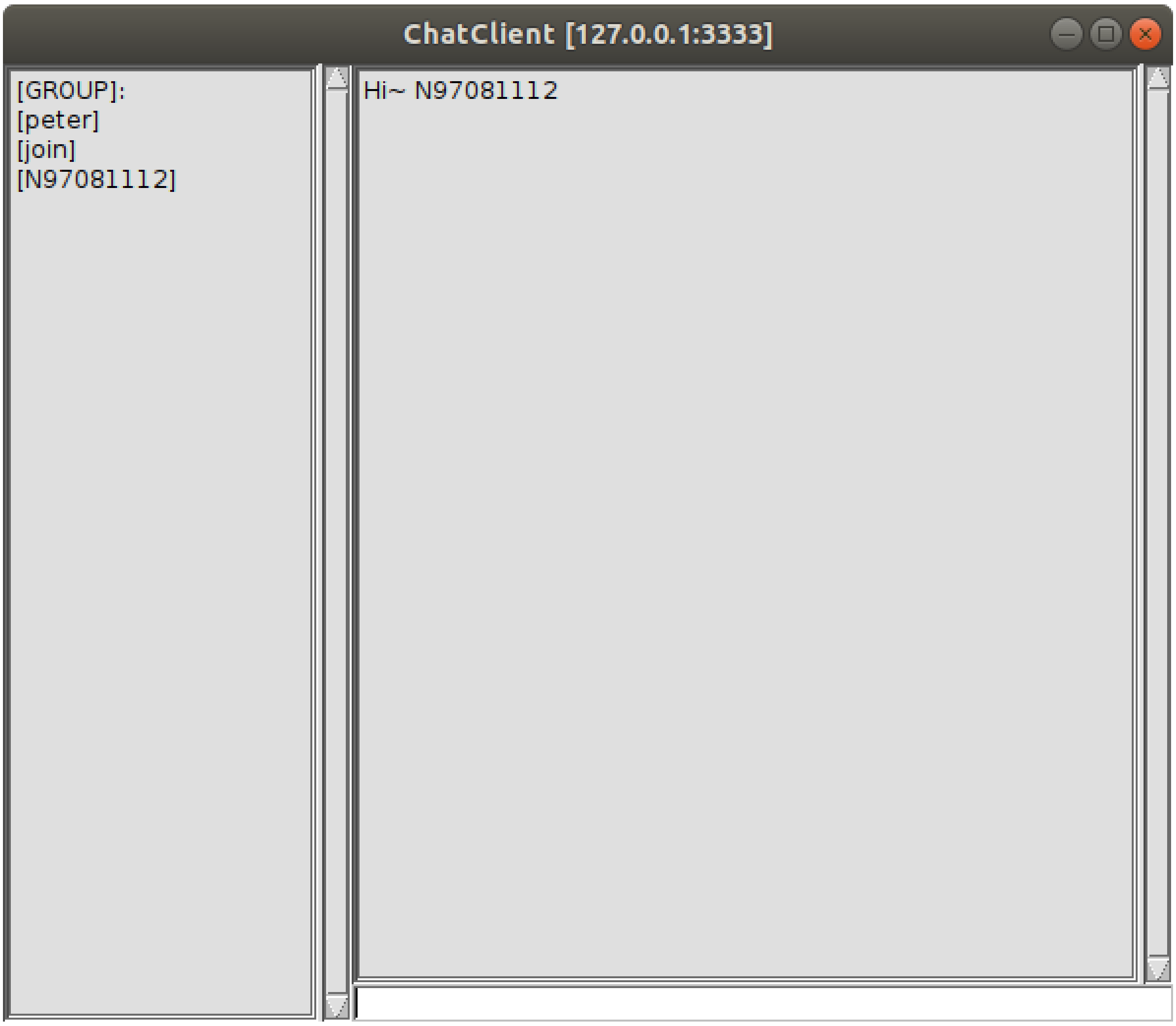
username: N97081112

password: 123456

Successful! Please login

再按一次 ”OK” 登入，成功後會出現對話匡。
左側為目前在登入在資料庫的用戶，右邊為對話訊息欄。
下方還有一輸入對話的欄位。

用2 Client相互輸入訊息



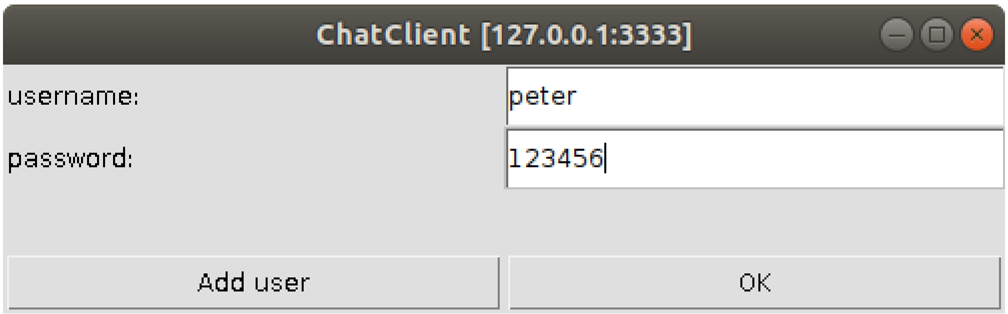
[GROUP]:
[peter]
[join]
[N97081112]

Hi~ N97081112

在開啟另一個client

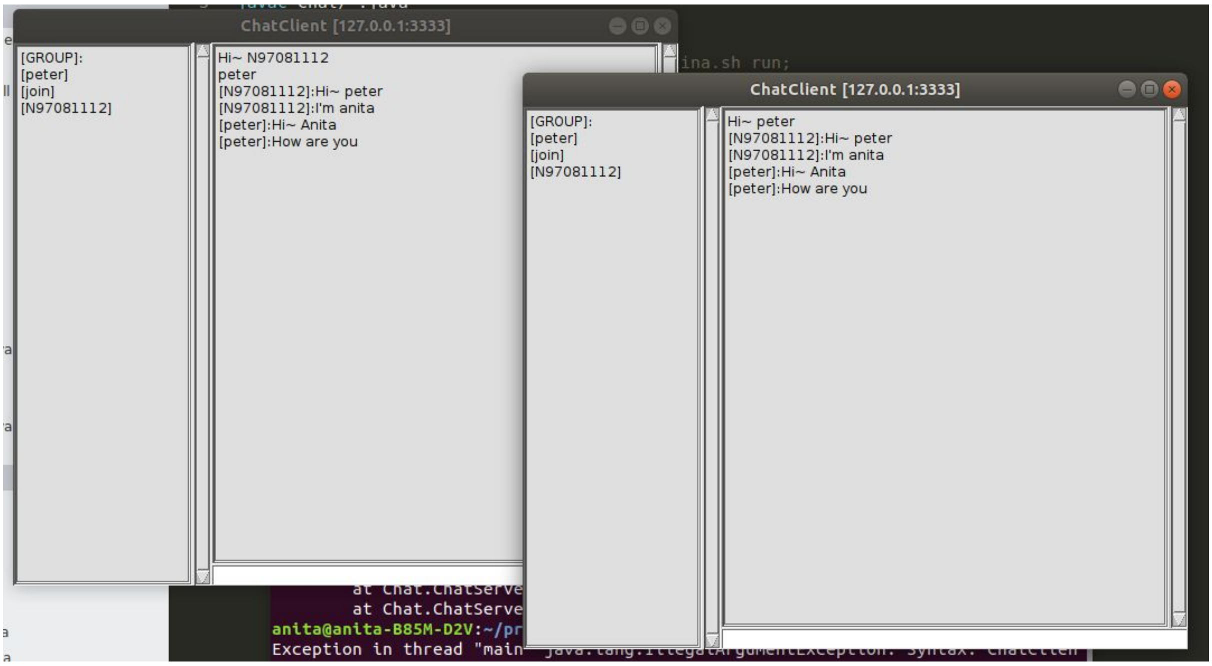
```
$ java Chat.ChatClient 127.0.0.1:3333
```

用peter登入



username: peter
password: 123456

用2 Client相互輸入訊息



```
[GROUP]:
[peter]
[join]
[N97081112]
```

```
Hi~ N97081112
peter
[N97081112]:Hi~ peter
[N97081112]:I'm anita
[peter]:Hi~ Anita
[peter]:How are you
```

```
[GROUP]:
[peter]
[join]
[N97081112]
```

```
Hi~ peter
[N97081112]:Hi~ peter
[N97081112]:I'm anita
[peter]:Hi~ Anita
[peter]:How are you
```

```
ina.sh run;
        at Chat.ChatServe
        at Chat.ChatServe
anita@anita-B85M-D2V:~/pr
Exception in thread "main" java.lang.IllegalArgumentException: Syntax: ChatClien
```

心得

Java 跟tomcat環境真的很複雜，本來以為很快可以架起來，所以沒有採用eclipse+maven去架環境（因為eclipse很肥不是很想裝）。結果光要找debug方式也找很久，指jar檔CLASSPATH 路徑也很崩潰，最後終於可以順利開發～～～ 撒花～～

雖然在想用eclipse+maven可能比較快，不過說不定是另外一個坑......