

Java Gossip: ServerSocket 類別

Socket類別主要在處理客戶端的Socket連線，如果要實作一個伺服器，可以使用ServerSocket類別，它包括了伺服器傾聽與客戶端連線的方法，您可以用數種方式來指定ServerSocket建構函式：

```
public ServerSocket(int port)
public ServerSocket(int port, int queueLength)
public ServerSocket(int port, int queueLength, InetAddress bindAddress)
```

以上的建構式皆需處理IOException, SecurityException，port是所指定要繫結（bind）的連接埠，而queueLength用來指定外來連線的佇列長度，bindAddress指定要繫結至哪一個網路介面。

ServerSocket擁有Socket類別取得相關資訊的能力，例如：

```
public InetAddress getInetAddress()
public int getLocalPort()
```

當要傾聽連線或關閉連線時，可以使用accept()與close()方法：

```
public Socket accept()
public void close()
```

這兩個方法需處理IOException，其中accept()傳回的是有關連線客戶端的Socket物件資訊，可以用它來取得客戶端的連線資訊，或關閉客戶端的連線。

下面這個程式是個簡單的Echo伺服器，您可以使用Telnet程式，或是Socket類別所實作的程式來測試它，它會將客戶端的文字指令再傳回客戶端，客戶端輸入bye可結束連線：

- EchoServer.java

```
package onlyfun.caterpillar;

import java.io.*;
import java.net.*;

public class EchoServer {
    public static void main(String[] args) {
        final int port = 7;
        ServerSocket serverSkt;
        Socket skt;
        BufferedReader sktReader;
        String message;
        PrintStream sktStream;

        try {
            serverSkt = new ServerSocket(port);
            try {
                while(true) {
                    System.out.printf("連接埠 %d 接受連線中.....\n", port);
                    skt = serverSkt.accept();
                    System.out.printf("與 %s 建立連線\n",
                        skt.getInetAddress().toString());

                    sktReader = new BufferedReader(new
                        InputStreamReader(skt.getInputStream()));

                    while((message = sktReader.readLine()) != null) {
                        if(message.equals("/bye")) {
                            System.out.println("Bye!");
                            skt.close();
                            break;
                        }

                        System.out.printf("Client: %s\n", message);
                        sktStream = new PrintStream(skt.getOutputStream());
                        sktStream.printf("echo: %s\n", message);
                    }
                }
            } catch(IOException e) {
                System.out.println(e.toString());
            }
        } catch(IOException e) {
            System.out.println(e.toString());
        }
    }
}
```

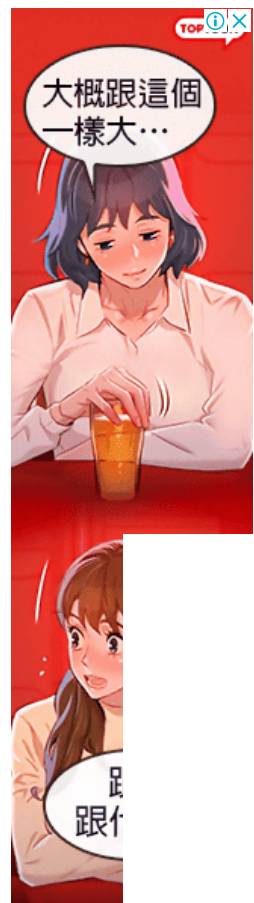
假設使用Telnet程式連線至Echo伺服器，並輸入以下的內容：

```
$ telnet localhost 7
Trying 127.0.0.1...
```



Level up
your CI/CD

Learn more



```
Connected to localhost.  
Escape character is '^]'.  
Hello! Echo Test!  
echo: Hello! Echo Test!  
哈囉！中文測試！  
echo: 哈囉！中文測試！  
/bye  
Connection closed by foreign host.
```

以下是Echo伺服器的回應：

```
$ java echoServer  
連接埠 7 接受連線中.....  
與127.0.0.1建立連線  
client say: Hello! Echo Test!  
client say: 哈囉！中文測試！  
Bye!  
連接埠 7 接受連線中.....
```