

COMP202-08B Computer Communications

Lecture 3 – Java Sockets

Matthew Luckie
mluckie@cs.waikato.ac.nz



21 July 2008

Overview of lecture

- Introduce Java
 - or reintroduce Java as the case may be
- Show, by example, how to program basic network applications with Java

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O HOKIANGA

21 July 2008

2

Hello, world.

```
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O HOKIANGA

21 July 2008

3

Running this program

```
$ javac HelloWorld.java
$ java HelloWorld
Hello World!
```

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O HOKIANGA

21 July 2008

4

Useful properties of Java

- Object oriented
- Garbage collection provided – you don't need to explicitly free resources.
- Errors are returned by the called method throwing an Exception
 - The java compiler will complain if you do not have necessary exception handling code.
- The Java compiler produces a binary that should be able to be run anywhere using the Java interpreter
- The Java API is fairly well documented by Sun
 - <http://java.sun.com/javase/6/docs/api/>

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O HOKIANGA

21 July 2008

5

Hello World, version 2.0

```
class HelloWorld2
{
    public static void main(String args[])
    {
        int i;
        System.out.println("There are " + args.length +
                           " parameters");
        for(i=0; i<args.length; i++)
        {
            System.out.println(" " + args[i]);
        }
    }
}
```

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O HOKIANGA

21 July 2008

6

Hello World, version 2.0

```
$ javac HelloWorld2.java
$ java HelloWorld2 COMP 202 08B
```

There are 3 parameters

COMP
202
08B

© THE UNIVERSITY OF Waikato • TE WHARE WAIKATO O WAIKATO

21 July 2008

7

From earlier lectures

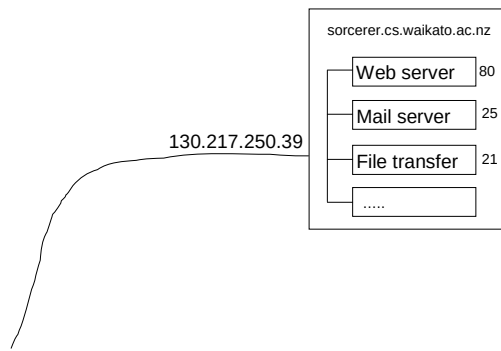
- DNS names map to IP addresses
- IP addresses identify systems on the Internet
- Applications run on a port

© THE UNIVERSITY OF Waikato • TE WHARE WAIKATO O WAIKATO

21 July 2008

8

From earlier lectures



© THE UNIVERSITY OF Waikato • TE WHARE WAIKATO O WAIKATO

21 July 2008

9

Dealing with IPv4 addresses in Java

- `java.net.InetAddress` class
- Useful methods:
 - `getByName` : resolve address with DNS
 - `getHostAddress` : return String containing IP address

© THE UNIVERSITY OF Waikato • TE WHARE WAIKATO O WAIKATO

21 July 2008

10

An example

```
String address =
    "sorcerer.cs.waikato.ac.nz";
try {
    InetAddress IP =
        InetAddress.getByName(address);
    System.out.println(address + " : " +
        IP.getHostAddress());
}
catch(UnknownHostException e) {
    System.err.println(address + " : unknown
        host");
}
```

© THE UNIVERSITY OF Waikato • TE WHARE WAIKATO O WAIKATO

21 July 2008

11

Output

sorcerer.cs.waikato.ac.nz : 130.217.250.39

© THE UNIVERSITY OF Waikato • TE WHARE WAIKATO O WAIKATO

21 July 2008

12

Doing the reverse

- It is also possible to obtain a DNS name of an IP address (if one exists) using `InetAddress::getHostName()`
- Sometimes if an IP address has a name, this method won't return it
 - if the name lookup takes too long to complete
- If the IP address does not have a name, this method returns the IP address looked up in a String

A second example

```
String address = "130.217.250.39";
try {
    InetAddress IP =
        InetAddress.getByName(address);
    System.out.println(address + " : " +
        IP.getHostName());
}
catch(UnknownHostException e) {
    System.err.println(address + " : unknown
    host");
}
```

Output

130.217.250.39 : sorcerer.cs.waikato.ac.nz

Networked hello world

- The rest of this lecture is concerned with what is known as a Socket
- A Socket is used to establish connections between networked machines on the Internet.
- `java.net.Socket`

Establishing a connection

- This code connects to the HTTP port on `www.wand.net.nz`

```
InetAddress wandIP =
    InetAddress.getByName("www.wand.net.nz");
Socket mysocket = new Socket(wandIP, 80);
```

Now what?

- We've established a connection, but now we need to be able to send and receive messages on it.
- There are multiple ways to do this in Java.

Input and Output streams

- At the lowest level, messages are made up in units of bytes
- You can obtain bytes by obtaining the corresponding input and output streams, and then using the read and write methods available

```
InputStream in = mysocket.getInputStream();
OutputStream out = mysocket.getOutputStream();
```
- But you probably don't want to do this yet.

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O WAIKATO

21 July 2008

29

Input and Output lines of text

- To start with, we want to deal with reading and writing text lines from Sockets
 - `java.io.BufferedReader`
 - `java.io.PrintWriter`
- `BufferedReader` returns whole lines of text per call
- `PrintWriter` assembles lines of text and writes to the Socket when a whole line of text is available

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O WAIKATO

21 July 2008

30

Writing lines of text

- `PrintWriter` is fairly easy to use

```
PrintWriter out = new
    PrintWriter(mysocket.getOutputStream(), true);
out.println("hello world");
out.close();
```

- The constructor takes an optional boolean parameter that controls whether or not it flushes on a line basis; you want to use `true` as the parameter.

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O WAIKATO

21 July 2008

31

Reading lines of text

- `BufferedReader` is a bit more complicated
 - its constructor takes a `Reader` object, but `Socket.getInputStream()` returns an `InputStream`.
 - The `InputStreamReader` class extends the `Reader` class, and it takes an `InputStream` as a parameter in its constructor, so we'll use that

```
InputStreamReader isr = new
    InputStreamReader(mysocket.getInputStream());
BufferedReader reader = new
    BufferedReader(isr);
```

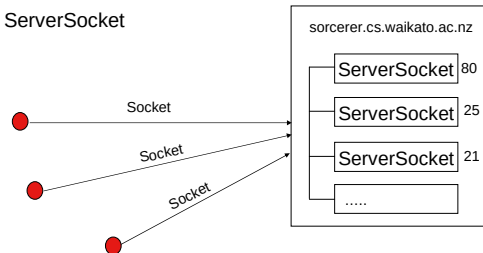
© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O WAIKATO

21 July 2008

32

ServerSocket

- There is a special type of socket for accepting new connections to the system in Java.
- `ServerSocket`



© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O WAIKATO

21 July 2008

33

ServerSocket

```
ServerSocket server = new ServerSocket(1234);
Socket client = server.accept();
```

© THE UNIVERSITY OF WAIKATO - TE WHARE WAIKATO O WAIKATO

21 July 2008

34

Wrapping it all up: HelloWorldServer.java

Note: THIS CODE IS MISSING NECESSARY EXCEPTION HANDLING

```
ServerSocket ss = new ServerSocket(1234);
while(true) {
    Socket client = ss.accept();
    PrintWriter writer = new
        PrintWriter(client.getOutputStream(), true);
    BufferedReader reader = new BufferedReader(
        new InputStreamReader(client.getInputStream()));
    String line = reader.readLine();
    writer.println("You said: " + line);
    client.close();
}
```

© THE UNIVERSITY OF WINNIPEG - TE WAIHARE WHAKANGA OWINNIPEG

21 July 2008

25

Wrapping it all up: HelloWorldClient.java

Note: THIS CODE IS MISSING NECESSARY EXCEPTION HANDLING

```
InetAddress IP = InetAddress.getByName(args[0]);
Socket me = new Socket(IP, 1234);
PrintWriter writer = new
    PrintWriter(me.getOutputStream(), true);
BufferedReader reader = new BufferedReader(
    new InputStreamReader(me.getInputStream()));
writer.println("Hello, World");
String response = reader.readLine();
me.close();
System.out.println("The other system said:\n" +
    response);
```

© THE UNIVERSITY OF WINNIPEG - TE WAIHARE WHAKANGA OWINNIPEG

21 July 2008

26

In summary

- The InetAddress class deals with resolving names to addresses
- The Socket class represents an individual connection established over the network
- The ServerSocket class is a special type of socket used to allow new connections to be made
- The PrintWriter class allows us to write individual lines of text over a socket
- The BufferedReader class allows us to read lines of text from a socket

© THE UNIVERSITY OF WINNIPEG - TE WAIHARE WHAKANGA OWINNIPEG

21 July 2008

27