

# Compte-Rendu TP de Java n°2

## Ce n'est pas une chaussette

On implémente le code suivant, notez que nous utilisons le port 8080 au lieu du port 80, car notre programme n'a pas les droits pour écouter le port 80.

Dans le fichier WebServer.java :

```
1  package TP_2;
2
3  import java.io.BufferedReader;
4  import java.io.BufferedWriter;
5  import java.io.InputStream;
6  import java.io.InputStreamReader;
7  import java.io.OutputStream;
8  import java.io.OutputStreamWriter;
9  import java.net.ServerSocket;
10 import java.net.Socket;
11
12 public class WebServer {
13
14     public WebServer(int _port){
15         run(_port);
16     }
17     private void readRequest(Socket socket){
18         try {
19             BufferedReader input = new BufferedReader(new InputStreamReader(socket.getInputStream()));
20             try {
21                 String response = "";
22                 while(! (response = input.readLine()).equals("")){
23                     System.out.println(response);
24                 }
25             } catch (Exception e2) {
26                 System.out.println(e2);
27                 System.out.println("RR-e2");
28             }
29         } catch (Exception e1) {
30             System.out.println(e1);
31             System.out.println("RR-e1");
32         }
33     }
```

```
34     private void sendResponse(Socket socket){
35         try {
36             BufferedWriter output = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));
37             try {
38                 //pour avoir une page html sur le navigateur
39                 output.write("HTTP/1.1 200 OK\r\n");
40                 output.write("Content-Type: text/html\r\n");
41                 output.write("\r\n");
42                 output.write("<html><body><h1>It works!</h1></body></html>\r\n");
43             } catch (Exception e3) {
44                 output.flush();
45                 System.out.println(e3);
46                 System.out.println("SR-e3");
47             }
48         } catch (Exception e2) {
49             System.out.println(e2);
50             System.out.println("SR-e2");
51         }
52     } catch (Exception e1) {
53         System.out.println(e1);
54         System.out.println("SR-e1");
55     }
56 }
57
58 public void run(int number){
59     try{
60         ServerSocket serverSocket = new ServerSocket(number);
61         while(true) {
62             Socket clientSocket = serverSocket.accept();
63             readRequest(clientSocket);
64             sendResponse(clientSocket);
65             clientSocket.close();
66         }
67     } catch (Exception except){
68         System.out.println(except);
69         System.out.println("Numéro de Port invalide");
70     }
71 }
72 }
```

Notez par ailleurs que j'ai mis beaucoup de temps à faire fonctionner correctement le programme, le problème venant finalement de la condition du while dans la fonction readResponse.

Dans le fichier WebServerApplication.java :

```
1 package TP_2;
2
3
4 public class WebServerApplication {
5     public static void main(String[] args){
6         WebServer firstWebServ = new WebServer(_port:8080);
7     }
8 }
9
10
```

## Reorganisation

### HttpRequest

On écrit le code ci-dessous pour la classe HttpRequest :

```
1 import java.io.BufferedReader;
2 import java.net.Socket;
3
4 public class HttpRequest {
5     private String method;
6     private String url;
7     private readClientRequest(Socket socket){
8
9     }
10
11     public HttpRequest(Socket socket){
12         BufferedReader input = new BufferedReader(new InputStreamReader(socket.getInputStream()));
13         //on remarque que les informations sont séparés par une espace
14         String[] request = input.readLine().split(" ");
15         method = request[0];
16         url = request[1];
17     }
18
19     public String getMethod(){
20
21         return method;
22     }
23     public String getUrl(){
24         return url;
25     }
26
27 }
```

## HttpResponse

```
1  import java.io.BufferedWriter;
2  import java.net.Socket;
3
4
5  public class HttpResponse {
6      private BufferedWriter output;
7
8      public void HttpResponse(Socket socket){
9
10     }
11     public void ok(String message){
12
13     }
14     public void notFound(String message){
15
16     }
17 }💡
18
```

Je n'ai pas pu aller plus loin , manque de temps / d'efficacité de ma part, ainsi qu'une migraine surprise lors de la seconde séance sur ce TP.