

华东师范大学数据科学与工程学院实验报告

课程名称：计算机网络与编程

年级：2021

上机实践成绩：

指导教师：张召

姓名：温兆和

学号：10205501432

上机实践名称：**Web 服务器实现（期末大项目）**

上机实践日期：2022.06.12

上机实践编号：15 16 17

组号：001-432

上机实践时间：09: 00

一、实验目的

综合运用本学期所学的知识，实现一个 Web 服务器。

二、实验任务

实现一个仅能处理一个请求的 Web 服务器；

实现一个多线程的 Web 服务器。

三、使用环境

IntelliJ IDEA 2020.3.2

JDK 11.0.6

四、实验过程

Task1. 实现一个简单的 Web 服务器，它仅能处理一个请求。

代码：

```
package WEIZHI;
import com.sun.net.httpserver.Request;

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

public class TCPServer {
    private static final int PORT = 8081;
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket();
        serverSocket.setReuseAddress(true);
        serverSocket.setReceiveBufferSize(64 * 1024 * 1024);
        serverSocket.bind(new InetSocketAddress("127.0.0.1", PORT), 50);
        System.out.println("Blocked and waiting for the client connection...");
        Socket socket = serverSocket.accept();
        InputStream is = socket.getInputStream();
        InputStreamReader isr = new InputStreamReader(is);
        BufferedReader br = new BufferedReader(isr);
        OutputStream os = socket.getOutputStream();
        String str1 = "/index.html";
        String str2 = "/shutdown";
        String info = Request(is);
        System.out.println("Received: "+info);
        if(!info.equals(str2))
        {
            if(info.equals(str1)){
                os.write(("HTTP/1.1 200\n"
                    + "Content-Type: text/html\n"
                    + "\n"
                    + "<html><head></head>"
                    + "<body><h1> ID Number: 10205501432"
```

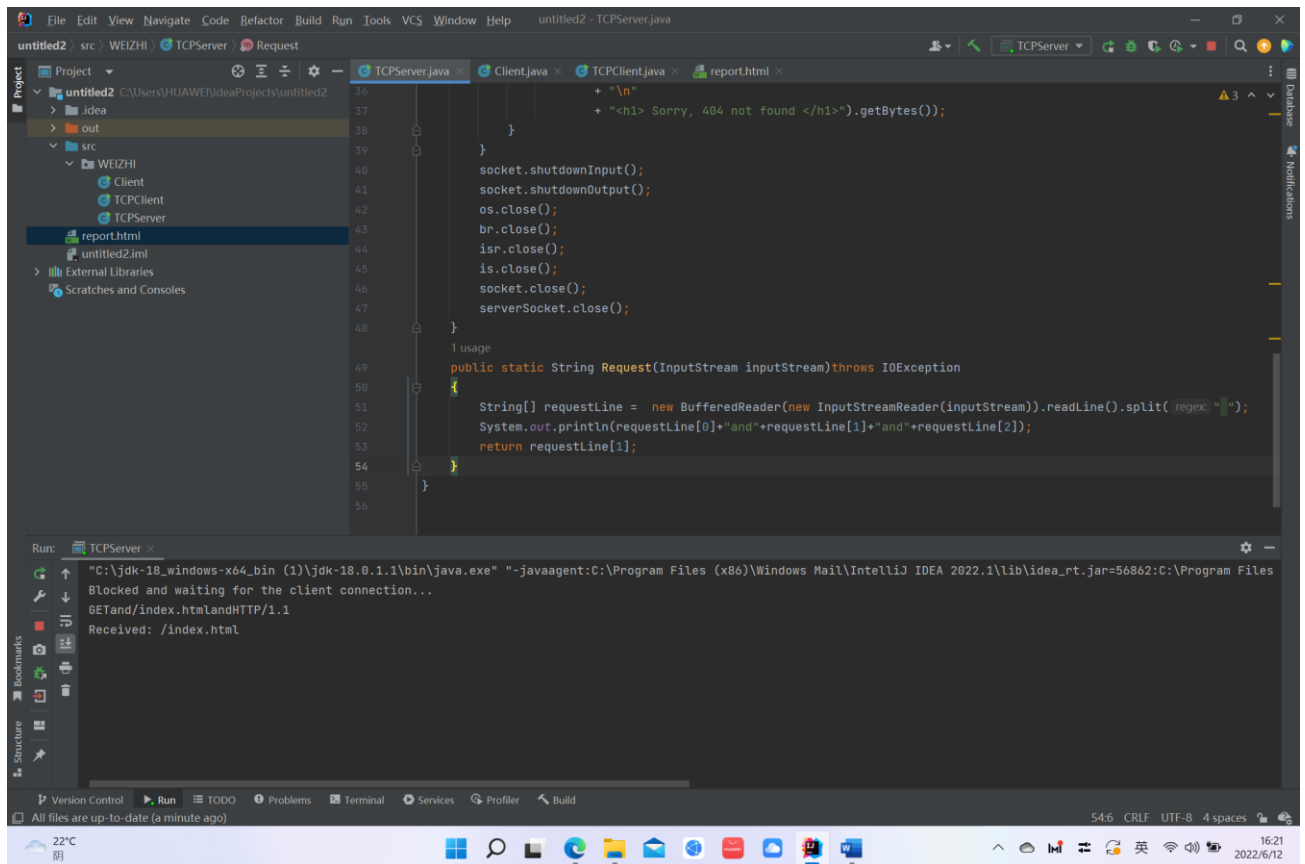
```
</h1></body></html>").getBytes());
    }
    else {
        os.write(("HTTP/1.1 404\n"
            + "Content-Type: text/html\n"
            + "\n"
            + "<h1> Sorry, 404 not found </h1>").getBytes());
    }
}
socket.shutdownInput();
socket.shutdownOutput();
os.close();
br.close();
isr.close();
is.close();
socket.close();
serverSocket.close();
}
public static String Request(InputStream inputStream) throws IOException
{
    String[] requestLine = new BufferedReader(new
InputStreamReader(inputStream)).readLine().split(" ");
    System.out.println(requestLine[0]+"and"+requestLine[1]+"and"+requestLine[2]);
    return requestLine[1];
}
}
```

结果 1: (输入 localhost:8081/index.html 后显示学号信息)



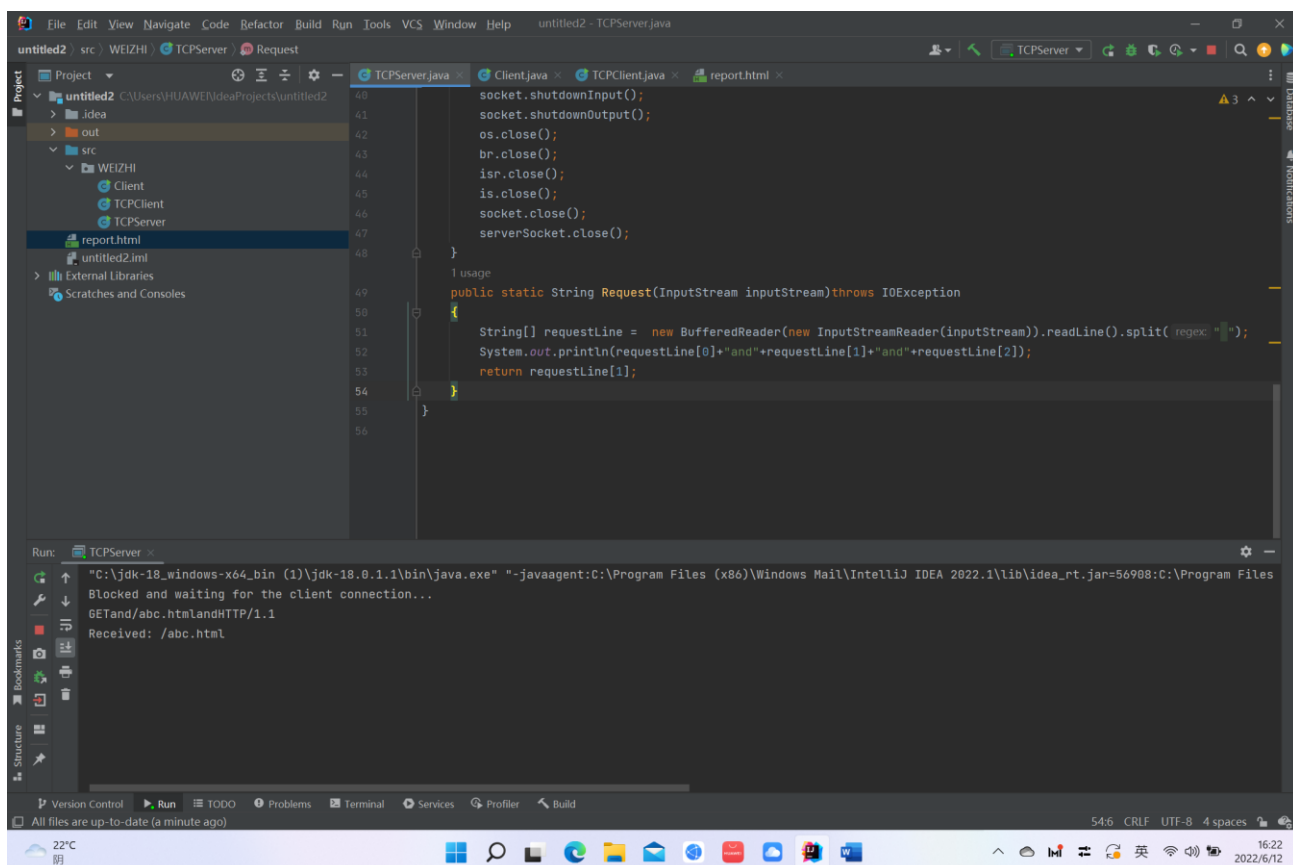
ID Number: 10205501432





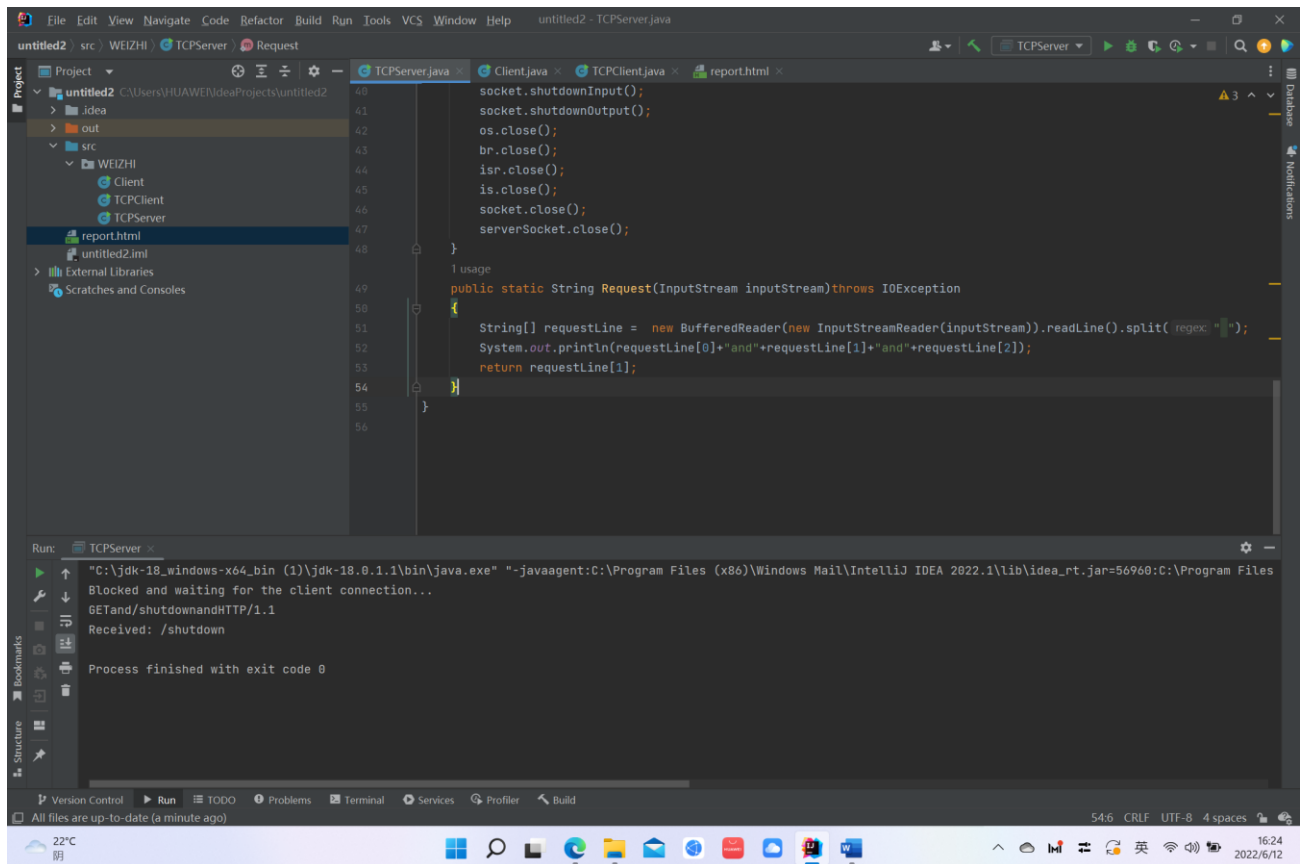
结果二：（输入 localhost:8081 下任何无效路径，显示 404 not found）



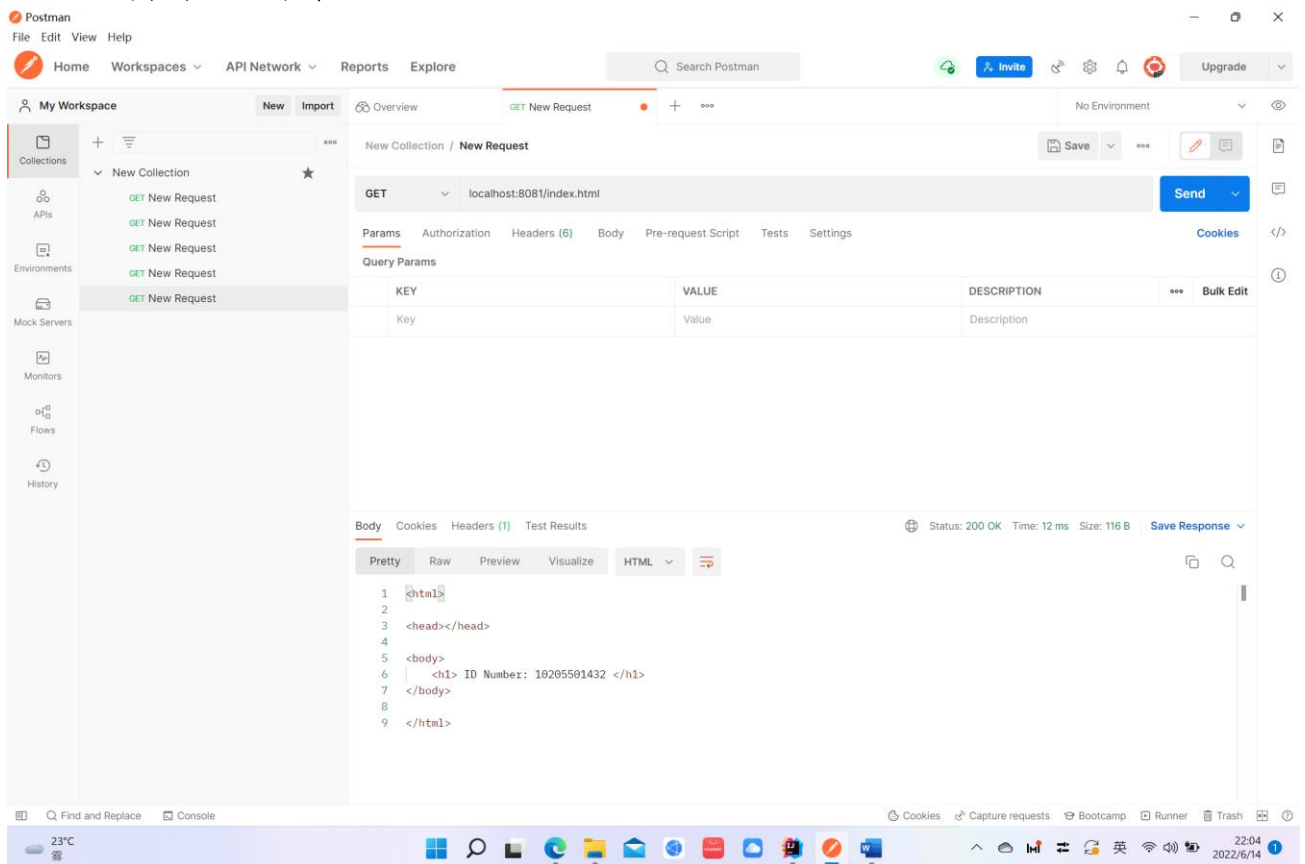


结果三：（输入 localhost:8081/shutdown 后关闭服务器）





Postman 测试 GET 命令:



The image displays two screenshots of software development environments. The top screenshot shows the IntelliJ IDEA IDE with a Java project named 'untitled2'. The project structure includes a 'src' directory with files 'WEIZHI', 'Client', 'DAILITCPServer.java', 'TCPClient.java', and 'report.html'. The 'DAILITCPServer.java' file is open, showing a TCP server implementation. The code includes a 'Request' method that reads a request line and returns the second part of the request. The bottom screenshot shows the Postman application with a 'New Request' dialog. The request is a GET request to 'localhost:8081/abc.html'. The response is a 404 Not Found status with a message 'Sorry, 404 not found </h1>'. The response body is displayed in HTML format.

IntelliJ IDEA Screenshot:

```

36         + "\n"
37         + "<h1> Sorry, 404 not found </h1>").getBytes();
38     }
39 }
40
41 socket.shutdownInput();
42 socket.shutdownOutput();
43 os.close();
44 br.close();
45 isr.close();
46 is.close();
47 socket.close();
48 serverSocket.close();
49 }
50
51 1 usage
52 public static String Request(InputStream inputStream) throws IOException
53 {
54     String[] requestLine = new BufferedReader(new InputStreamReader(inputStream)).readLine().split(" ");
55     System.out.println(requestLine[0]+"and"+requestLine[1]+"and"+requestLine[2]);
56     return requestLine[1];
57 }
58 }
59

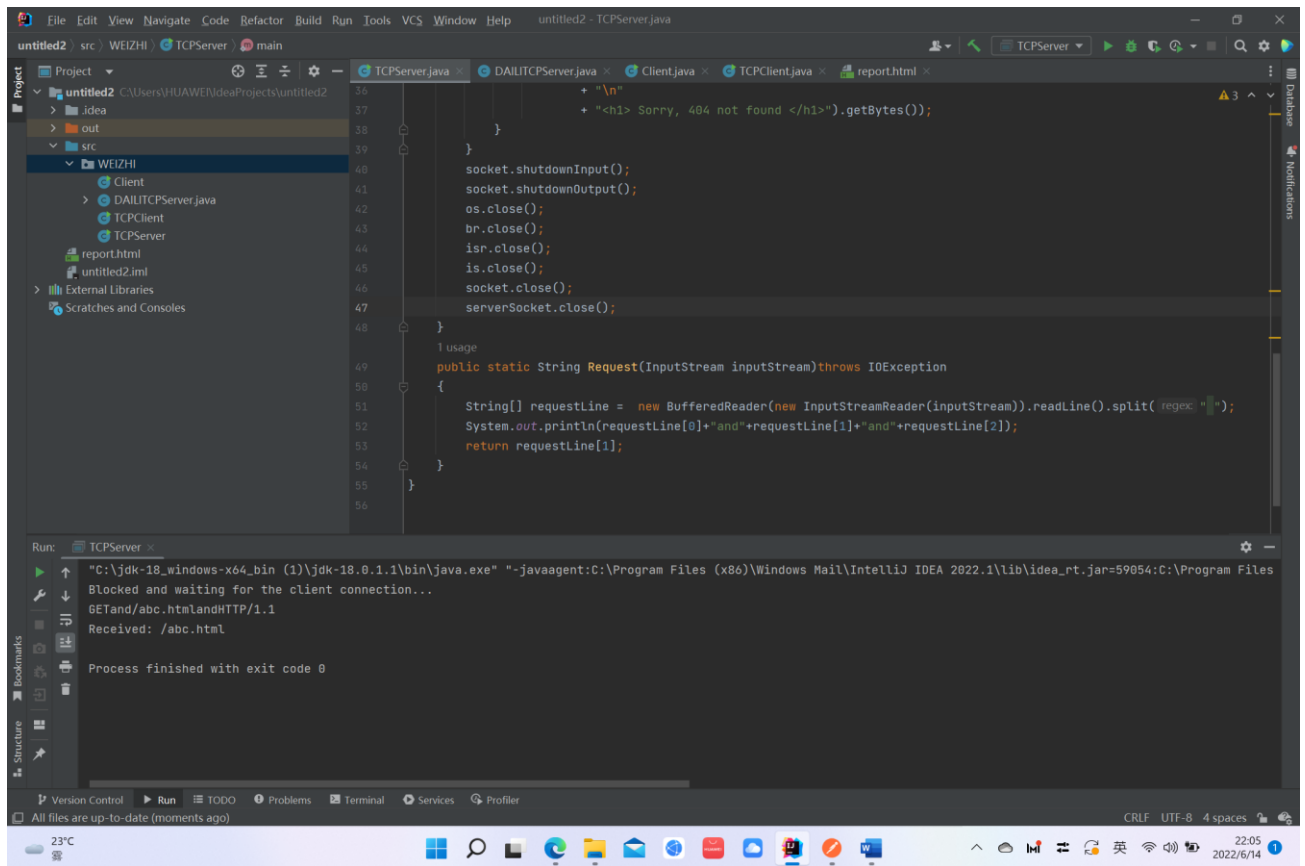
```

Postman Screenshot:

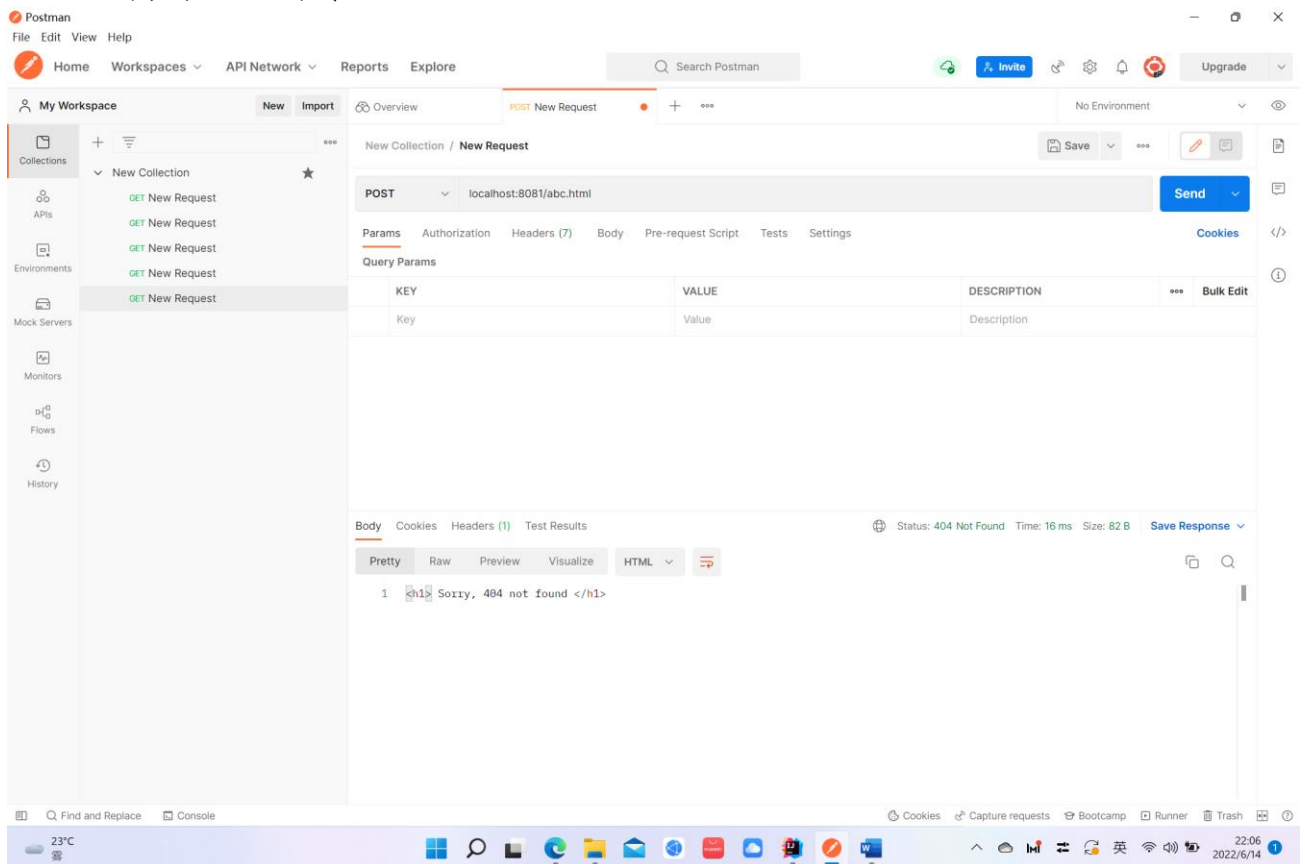
GET localhost:8081/abc.html

Status: 404 Not Found Time: 16 ms Size: 82 B

Body: <h1> Sorry, 404 not found </h1>



Postman 测试 POST 命令:



The image displays two screenshots related to a Java TCP server experiment.

Top Screenshot: IntelliJ IDEA

The IDE shows a project named "untitled2" with a package structure including "WEIZHI", "Client", "DALITCPServer.java", "TCPClient", and "TCPServer". The main file, "TCPServer.java", contains the following code:

```

36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
    }
    + "\n"
    + "<h1> Sorry, 404 not found </h1>").getBytes());
    }
    socket.shutdownInput();
    socket.shutdownOutput();
    os.close();
    br.close();
    isr.close();
    is.close();
    socket.close();
    serverSocket.close();
}
1 usage
public static String Request(InputStream inputStream) throws IOException
{
    String[] requestLine = new BufferedReader(new InputStreamReader(inputStream)).readLine().split(" ");
    System.out.println(requestLine[0]+"and"+requestLine[1]+"and"+requestLine[2]);
    return requestLine[1];
}
    }
    
```

The Run console shows the following output:

```

Run: TCPServer
"C:\jdk-18_windows-x64_bin (1)\jdk-18.0.1.1\bin\java.exe" "-javaagent:C:\Program Files (x86)\Windows Mail\IntelliJ IDEA 2022.1\lib\idea_rt.jar=59064:C:\Program Files
Blocked and waiting for the client connection...
POSTand/abc.htmlandHTTP/1.1
Received: /abc.html
Process finished with exit code 0
    
```

Bottom Screenshot: Postman

The Postman interface shows a "New Request" for a POST method to "localhost:8081/index.html". The "Query Params" table is empty.

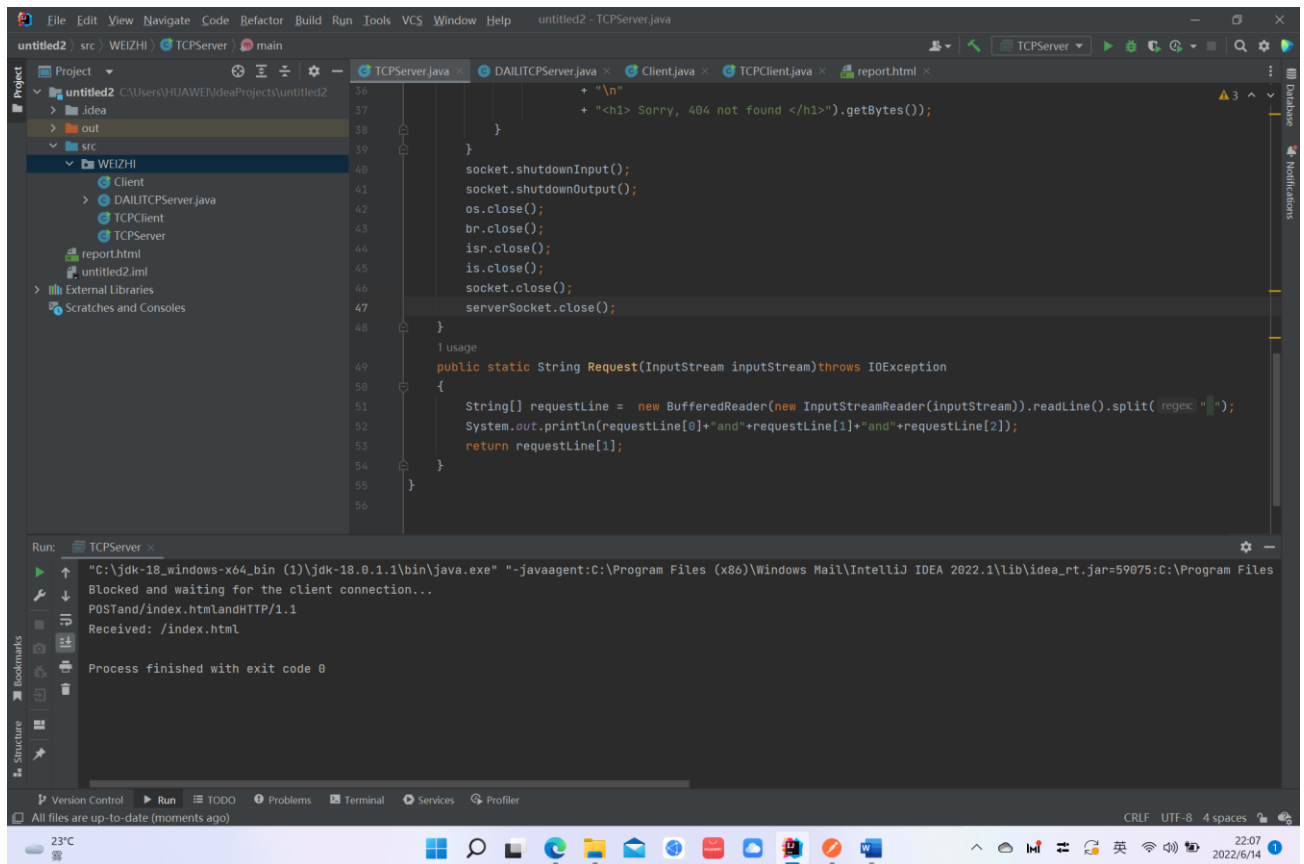
KEY	VALUE	DESCRIPTION
Key	Value	Description

The "Body" tab shows the response in HTML format:

```

1 <html>
2
3 <head></head>
4
5 <body>
6 | <h1> ID Number: 10205501432 </h1>
7 </body>
8
9 </html>
    
```

The status bar indicates "Status: 200 OK", "Time: 16 ms", and "Size: 116 B".



Task2. 实现一个简单的多线程 Web 代理服务器，让浏览器经过代理来请求 Web 对象。

代码：

```
package WEIZHI;
import java.io.*;
import java.net.*;

interface Responsemachine
{
    String response(String message);
}

class BEIDALILDEClientHandler implements Responsemachine
{
    private String str1 = "/index.html";
    private String str3 = "/shutdown";
    private String str2 = null;
    @Override
    public String response(String message) {
        if(!message.equals(str3))
        {
            if(message.equals(str1))
            {
                str2="HTTP/1.1 200\nContent-Type: text/html\n\n<h1> ID Number: 10205501432 </h1>";
            }
            else
            {
                str2="HTTP/1.1 404\nContent-Type: text/html\n\n<h1> Sorry, 404 not found </h1>";
            }
        }
        return str2;
    }
}
```

```
}  
}  
class ClientHandler extends Thread implements ResponseMachine{  
    private final Socket socket;  
    private ResponseMachine target;  
    ClientHandler(Socket socket, ResponseMachine target) {  
        this.socket = socket;  
        this.target = target;  
    }  
    @Override  
    public void run()  
    {  
        super.run();  
        try {  
            InputStream inputStream = socket.getInputStream();  
            OutputStream outputStream = socket.getOutputStream();  
            String info = Request(inputStream);  
            String ans = this.target.response(info);  
            if(!ans.equals(null))  
            {  
                outputStream.write(ans.getBytes());  
            }  
            outputStream.close();  
            inputStream.close();  
        } catch (IOException e) {  
            throw new RuntimeException(e);  
        } finally {  
            try {  
                socket.close();  
            } catch (IOException e) {  
                throw new RuntimeException(e);  
            }  
        }  
    }  
}  
    @Override  
    public String response(String message) {  
        return this.target.response(message);  
    }  
    public static String Request(InputStream inputStream) throws IOException  
    {  
        String[] requestLine = new BufferedReader(new  
        InputStreamReader(inputStream)).readLine().split(" ");  
        System.out.println(requestLine[0]+"and"+requestLine[1]+"and"+requestLine[2]);  
        return requestLine[1];  
    }  
}  
  
public class DAILITCPServer {  
    private static final int PORT = 8081;  
    public static void main(String[] args) throws IOException {  
        ServerSocket server = new ServerSocket();  
        server.setReuseAddress(true);  
        server.setReceiveBufferSize(64 * 1024 * 1024);  
        server.bind(new InetSocketAddress("127.0.0.1", PORT), 50);  
        int clietNum = 0;  
        while(true)  
        {  
            System.out.println("Blocked and waiting for the client connect...");
```

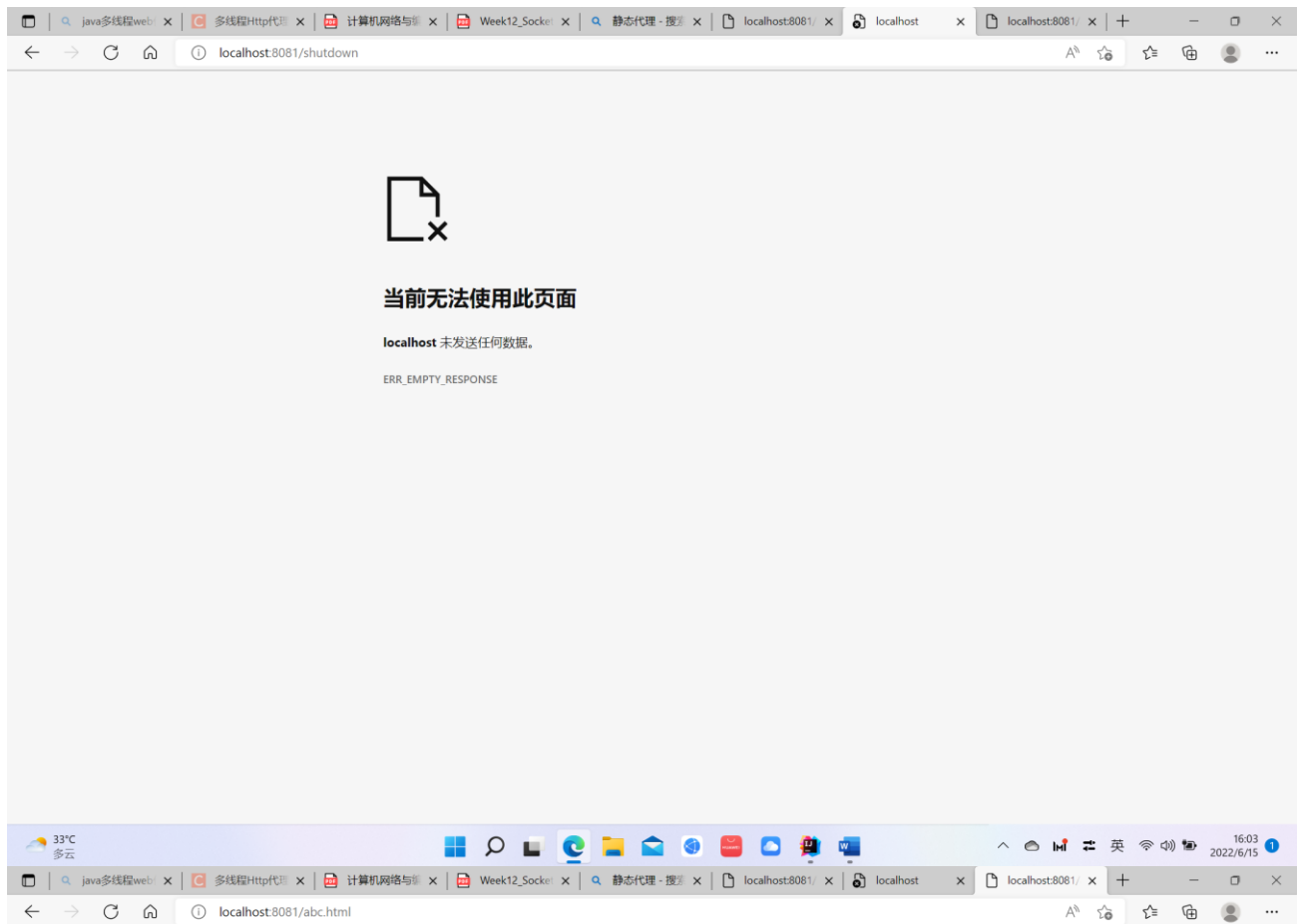
```
Socket client = server.accept();
BEIDALILDEClientHandler rm = new BEIDALILDEClientHandler();
clietNum++;
ClientHandler clientHandler = new ClientHandler(client, rm);
clientHandler.start();
}}
}
```

结果:



ID Number: 10205501432





Sorry, 404 not found



```

Run: DAILITCPServer
"C:\jdk-18_windows-x64_bin (1)\jdk-18.0.1\bin\java.exe" "-javaagent:C:\Program Files (x86)\Windows Mail\IntelliJ IDEA 2022.1\lib\idea_rt.jar=61241:C:\Program Files
Blocked and waiting for the client connect...
Blocked and waiting for the client connect...
GETand/index.htmlandHTTP/1.1
Blocked and waiting for the client connect...
Blocked and waiting for the client connect...
GETand/shutdownandHTTP/1.1
GETand/shutdownandHTTP/1.1
Exception in thread "Thread-1" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "ans" is null
    at WEIZHI.ClientHandler.run(DAILITCPServer.java:48)
Exception in thread "Thread-2" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "ans" is null
    at WEIZHI.ClientHandler.run(DAILITCPServer.java:48)
Blocked and waiting for the client connect...
Blocked and waiting for the client connect...
GETand/shutdownandHTTP/1.1
Blocked and waiting for the client connect...
Blocked and waiting for the client connect...
GETand/shutdownandHTTP/1.1
Exception in thread "Thread-3" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "ans" is null
    at WEIZHI.ClientHandler.run(DAILITCPServer.java:48)
Blocked and waiting for the client connect...
Blocked and waiting for the client connect...
GETand/shutdownandHTTP/1.1
Exception in thread "Thread-4" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "ans" is null
    at WEIZHI.ClientHandler.run(DAILITCPServer.java:48)
Exception in thread "Thread-5" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "ans" is null
    at WEIZHI.ClientHandler.run(DAILITCPServer.java:48)
Blocked and waiting for the client connect...
Blocked and waiting for the client connect...
GETand/shutdownandHTTP/1.1
Exception in thread "Thread-6" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "ans" is null
    at WEIZHI.ClientHandler.run(DAILITCPServer.java:48)
Blocked and waiting for the client connect...
GETand/abc.htmlandHTTP/1.1
Blocked and waiting for the client connect...

Process finished with exit code 130
    
```

部分 Postman 测试结果:

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

My Workspace

New Collection

GET localhost:8081/index.html

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

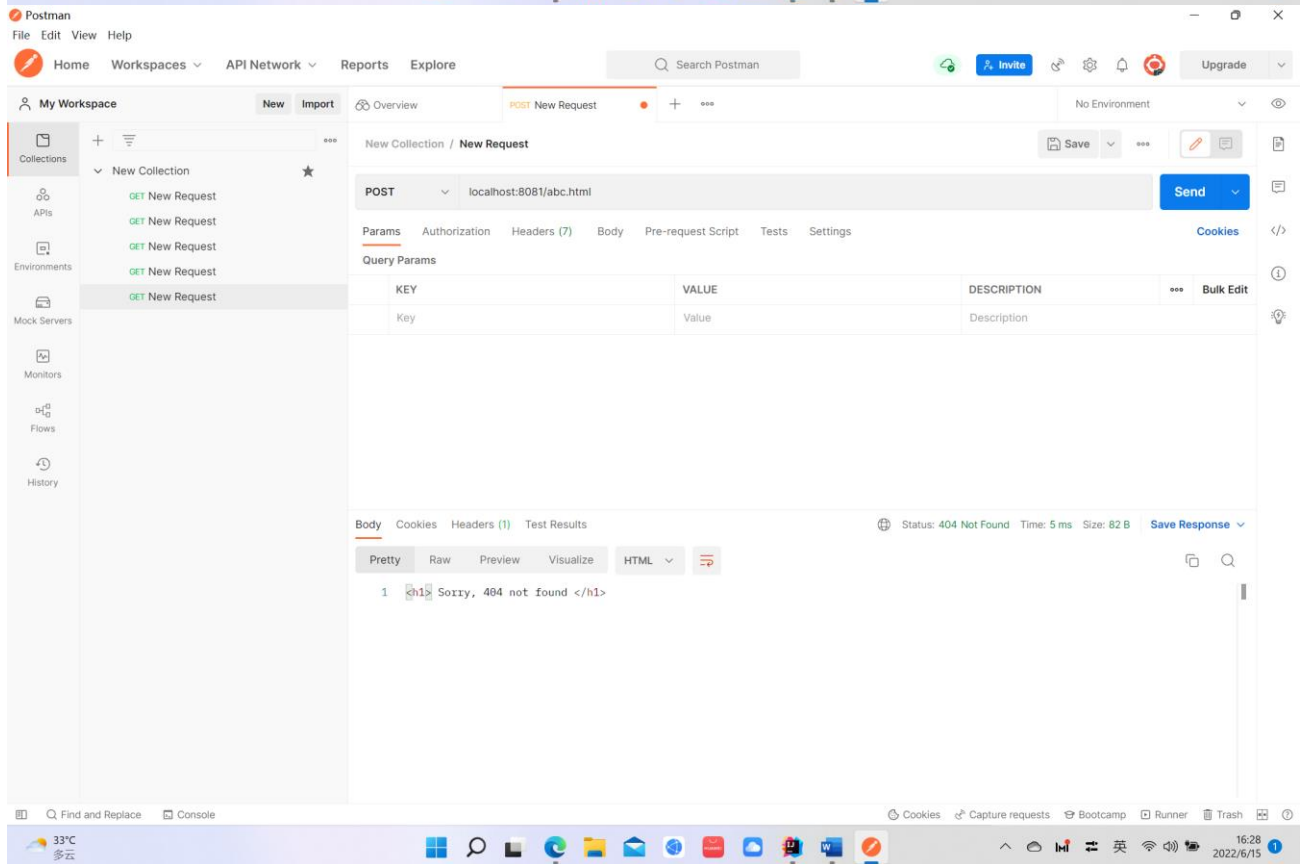
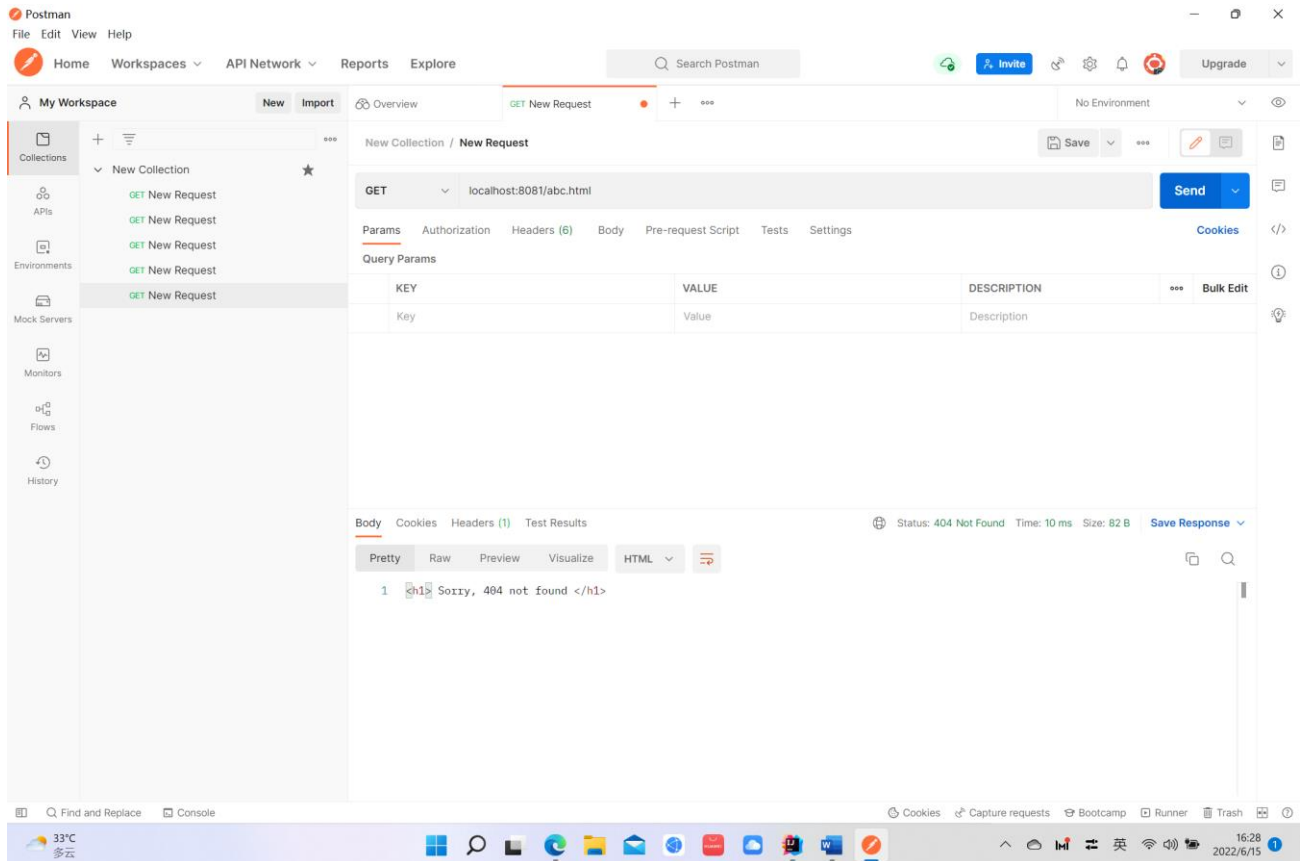
KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (1) Test Results

Status: 200 OK Time: 23 ms Size: 77 B

Pretty Raw Preview Visualize HTML

1 <h1> ID Number: 10205501432 </h1>

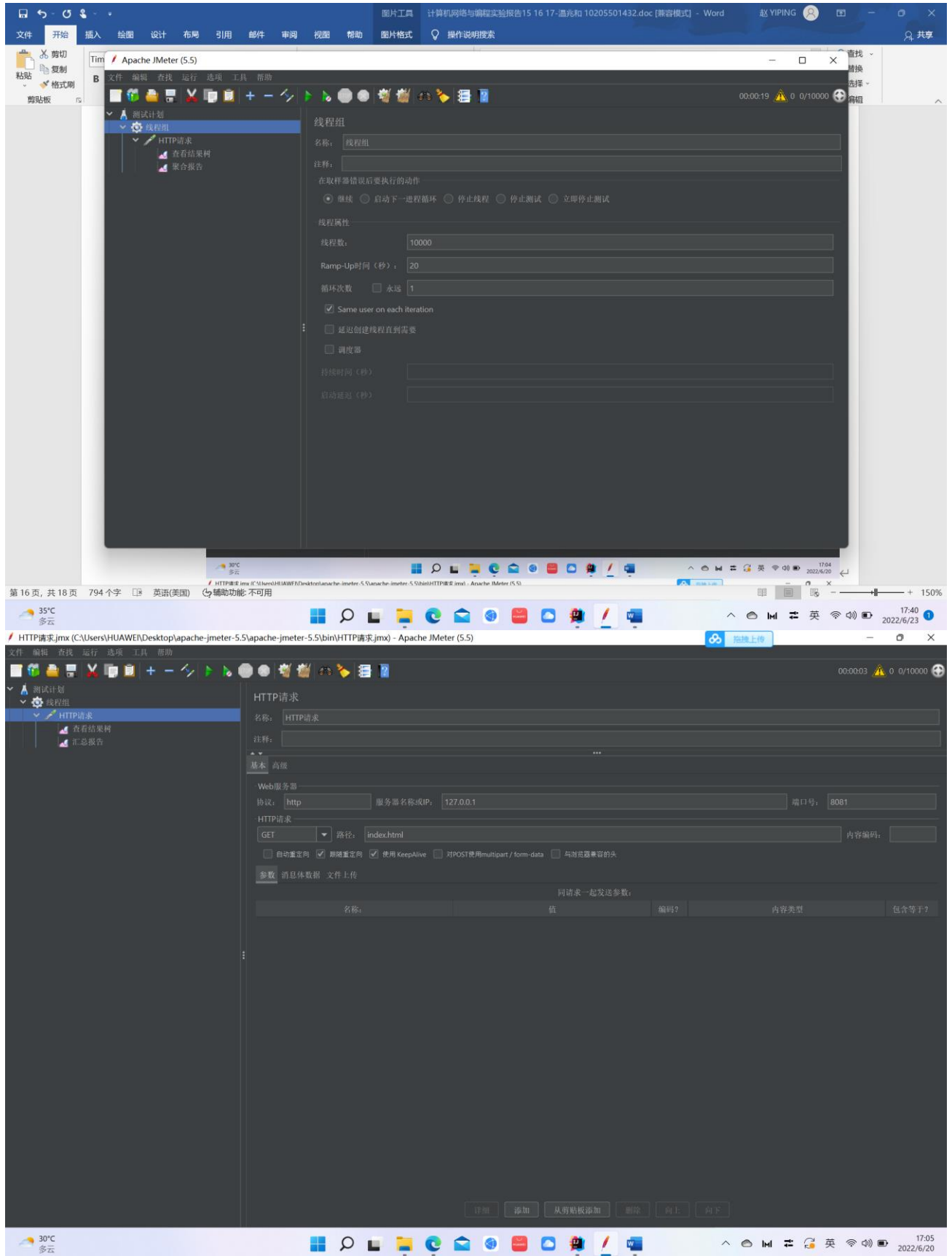


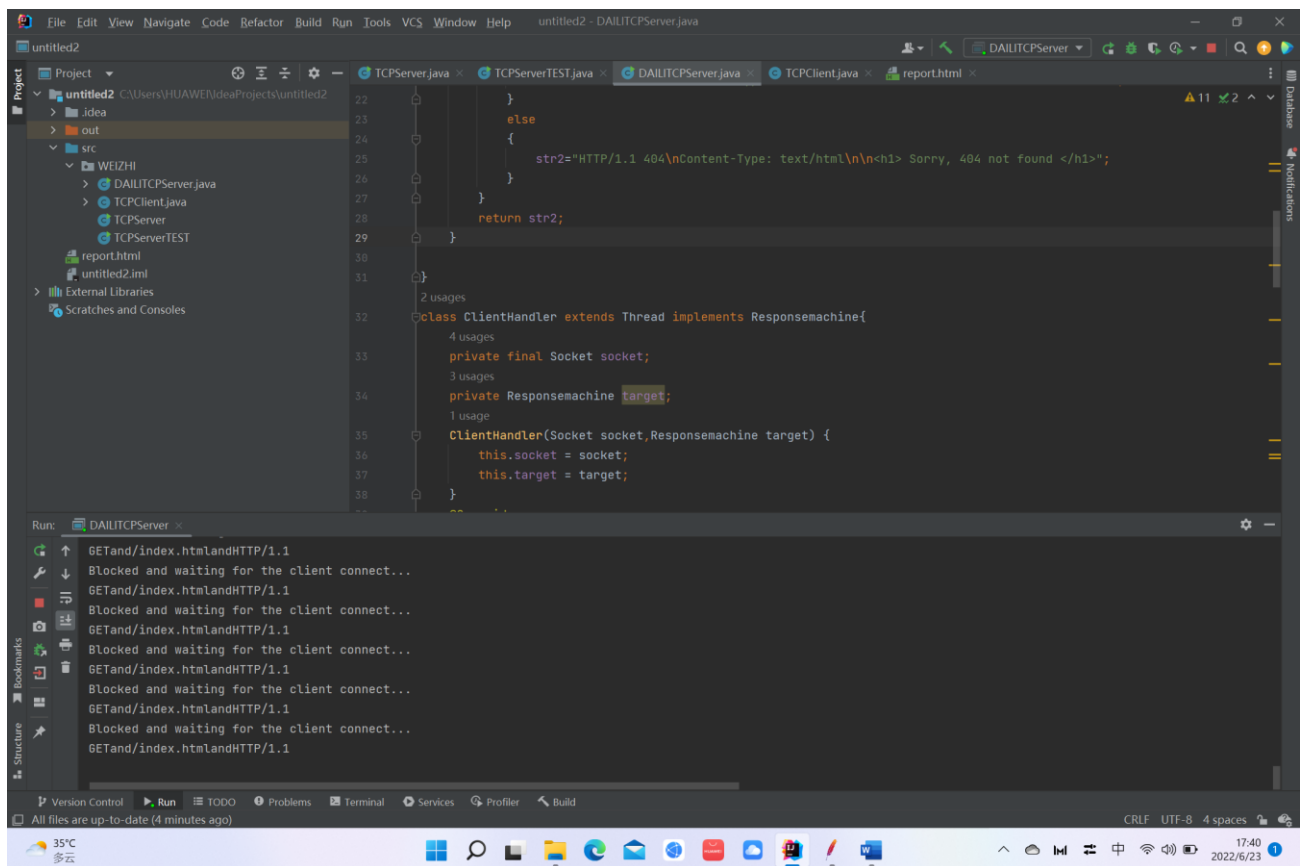
The image displays two screenshots from a computer screen. The top screenshot shows the Postman application interface. A new POST request is configured with the URL `localhost:8081/index.html`. The response body, viewed in 'Pretty' format, shows an XML-like structure: `<h1> ID Number: 18205501432 </h1>`. The status is 200 OK, and the response size is 77 B.

The bottom screenshot shows the IntelliJ IDEA IDE. The project structure on the left includes a package named `WEIZHI` with files `DAILITCPServer.java`, `TCPClient.java`, and `TCPServer.java`. The main editor displays the code for `DAILITCPServer.java`, which defines an interface `ResponseMachine` and a class `BEIDALILDEClientHandler` that implements it. The code includes imports for `java.io.*` and `java.net.*`, and uses `String` variables to handle requests.

At the bottom of the IDE, the 'Run' console shows the output of the program. It indicates that the server is blocked and waiting for client connections, followed by several GET and POST requests to `localhost:8081/index.html`. The process eventually finishes with an exit code of 130.

并发测试结果：





五、总结

在这几周的实验中，我们综合运用本学期的知识，分别实现了一个单线程的和一个多线程的 Web 服务器，为以后分布式系统的开发打下了基础。