WEB ARCHITECTURES
Assignment 1

Riccardo Gennaro

30 September 2022

1 First part: simple web server modification

1.1 Introduction: problem statement

The problem that this project addresses consists in enabling a client, via HTTP, access to a server-side service, namely an external process used to reverse a string given via a GET request parameter. The reversed string will be returned to the client browser and shown. To do so, the client uses a browser and access the service located at the URL
`http://localhost:8080/process/reverse?par1="yourInput"`,
where `par1` is the GET parameter containing the input to reverse. Also the server must be able to handle wrong spelled URL paths, requests different from GET, and parameters with names different from `par1`.

1.2 Proposed solution

In order to enable the client-server HTTP communication, TinyHttpd, the class seen during lesson, was used to open a socket on port 8080 on server side. The server listen on this port and wait for an HTTP request.

To handle the HTTP request, the HTTP daemon used during lesson was modified to address the problem of this assignment.

The main daemon modifications consist in the request parsing and in the implementation of a call to an external process, the one used to reverse the string.

- `Request parsing:` the parsing of the request is handled by the constructor of the class `RequestParser` that run the same checks implemented in the original TinyHttpdConnection class, but also checks if the path corresponds to `process/reverse`, and if the parameter has name equal to `par1`. If not, depending on the case, status codes `404 Not Found` or `400 Bad Request` are returned. Once this checks are completed, the value of the parameter is extracted from the request and parsed in order to handle strings containing spaces.

- `Call to external process:` this call is handled by the constructor of the class `RunReverse` that uses the class `java.lang.ProcessBuilder` to start the program (`Reverse.java`) used to reverse the input parameter. In order to retrieve the output of `Reverse.java`, the output stream is captured by the parent process (our RunReverse.java class), parsed and stored in a variable.

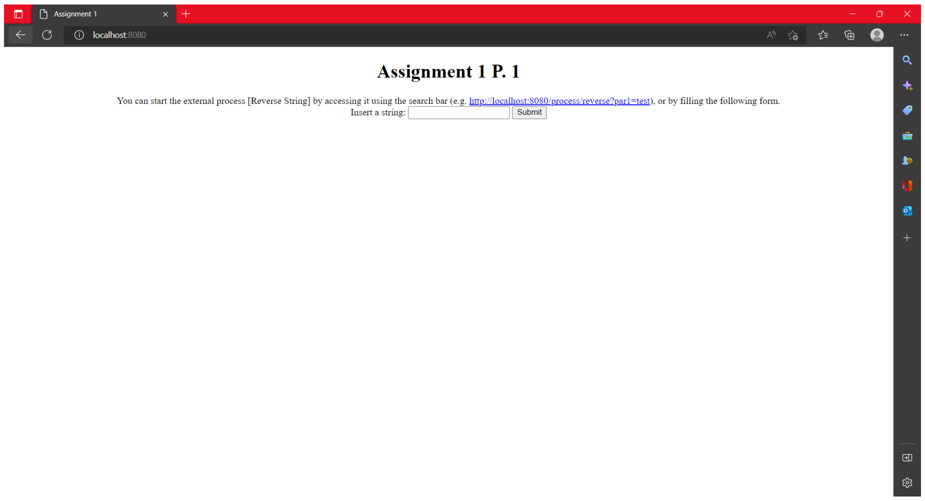
If the request is found to be valid, a response is crated and sent to the client.

1.3 Screenshots

Following, various screenshot of the running application on the client browser.

1.3.1 Index Page

Image 1 shows the index.html, accessible via the URL `http://localhost:8080`. It contains a form in which it is possible to input the string to reverse. It is also possible to input the string as a parameter named `par1` via the URL.



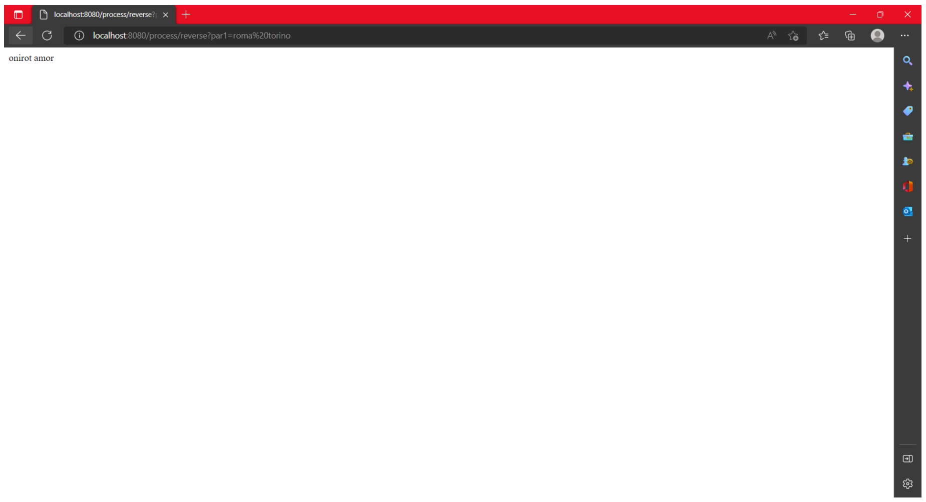
Assignment 1 P. 1

You can start the external process [Reverse String] by accessing it using the search bar (e.g. http://localhost:8080/process/reverse?par1=test), or by filling the following form.
Insert a string:

Figure 1: index.html

1.3.2 Output Page

After having submitted the form, or having sent a request via the search bar, a page is shown with the reversed input.
Image 2, depicts a test request via URL `http://localhost:8080/process/reverse?par1=roma torino`.

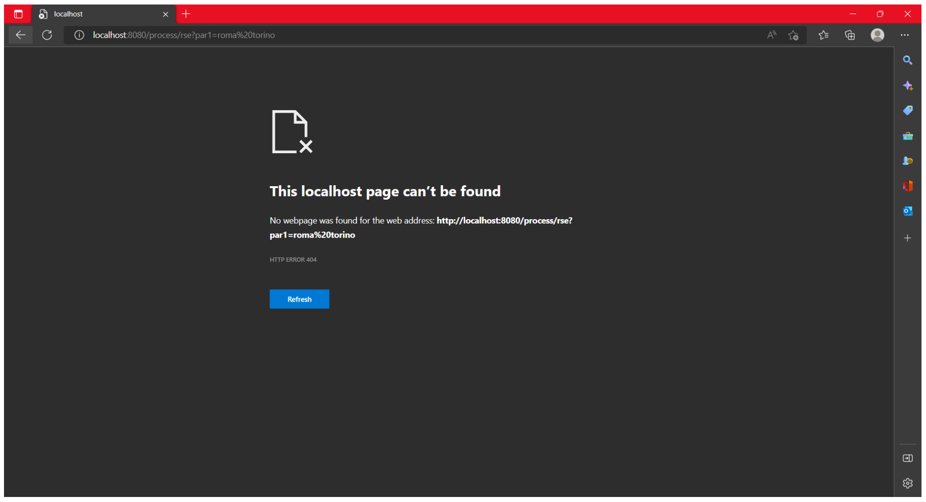

localhost:8080/process/reverse?par1=roma%20torino

```
onirot amor
```

Figure 2: Output page

1.3.3 Error Pages

Following, the pages shown after having tried to input a path different from `process/reverse` (image 3), and after a sending a parameter with the wrong name (image 4).

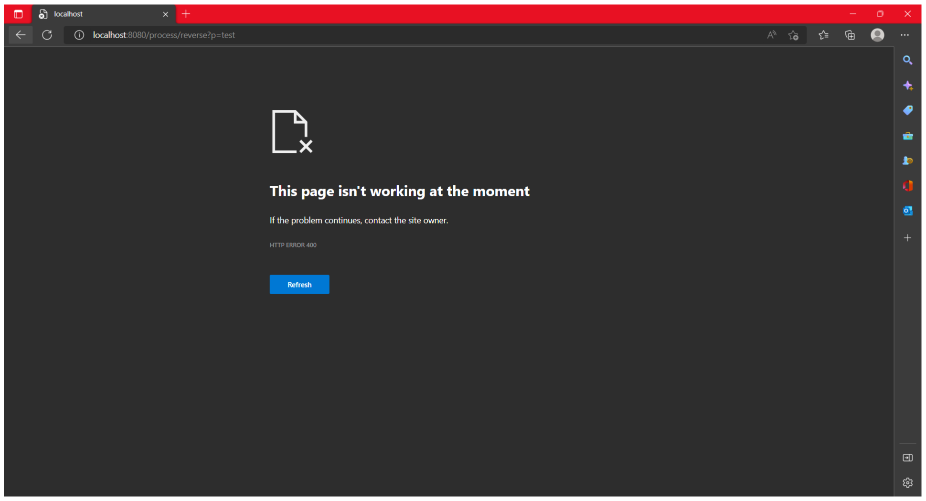

localhost:8080/process/rse?par1=roma%20torino

This localhost page can't be found

No webpage was found for the web address: **http://localhost:8080/process/rse?par1=roma%20torino**

HTTP ERROR 404

Figure 3: Not Found error



localhost:8080/process/reverse?p=test

This page isn't working at the moment

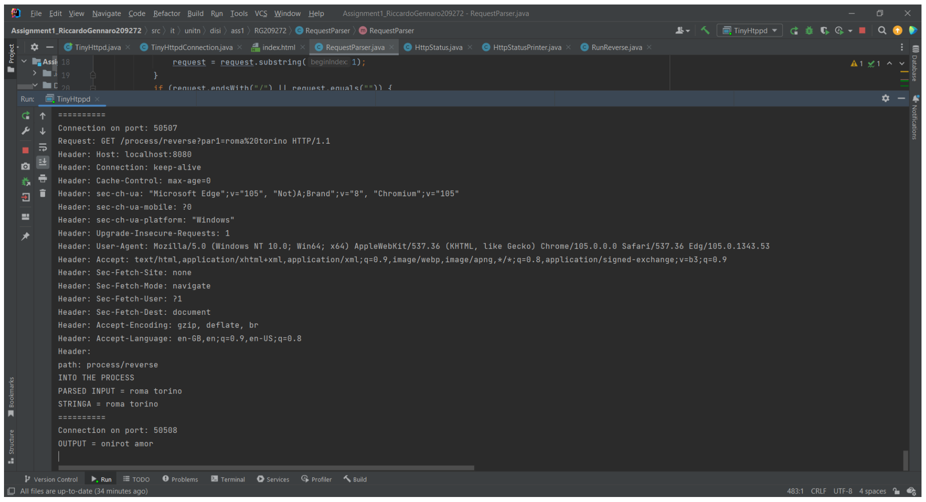
If the problem continues, contact the site owner.

HTTP ERROR 400

Figure 4: Bad Request error

1.4 IntelliJ Run Window

The server-side run terminal after a valid request with parameter `par1` = ”roma torino”.



```
==========
Connection on port: 50507
Request: GET /process/reverse?par1=roma%20torino HTTP/1.1
Header: Host: localhost:8080
Header: Connection: keep-alive
Header: Cache-Control: max-age=0
Header: sec-ch-ua: "Microsoft Edge";v="105", "Not)A;Brand";v="8", "Chromium";v="105"
Header: sec-ch-ua-mobile: ?0
Header: sec-ch-ua-platform: "Windows"
Header: Upgrade-Insecure-Requests: 1
Header: User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36 Edg/105.0.1343.53
Header: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Header: Sec-Fetch-Site: none
Header: Sec-Fetch-Mode: navigate
Header: Sec-Fetch-User: ?1
Header: Sec-Fetch-Dest: document
Header: Accept-Encoding: gzip, deflate, br
Header: Accept-Language: en-GB,en;q=0.9,en-US;q=0.8
Header:
path: process/reverse
INTO THE PROCESS
PARSED INPUT = roma torino
STRINGA = roma torino
==========
Connection on port: 50508
OUTPUT = onirot amor
```

Figure 5: Intellij run window

1.5 Comments and notes

The only problem encountered during the development was the implementation of the handling of the HTTP errors and their return points inside TinyHttpd-Connection. To solve this problem a not so elegant approach was used: after having printed an HTTP error in `RequestParser.java`, a flag is set and made accessible to `TinyHttpdConnection.java` in order to return if the request is not valid.

2 Second Part: accessing .bat through Apache Common Gateway Interface

2.1 Solution Proposed

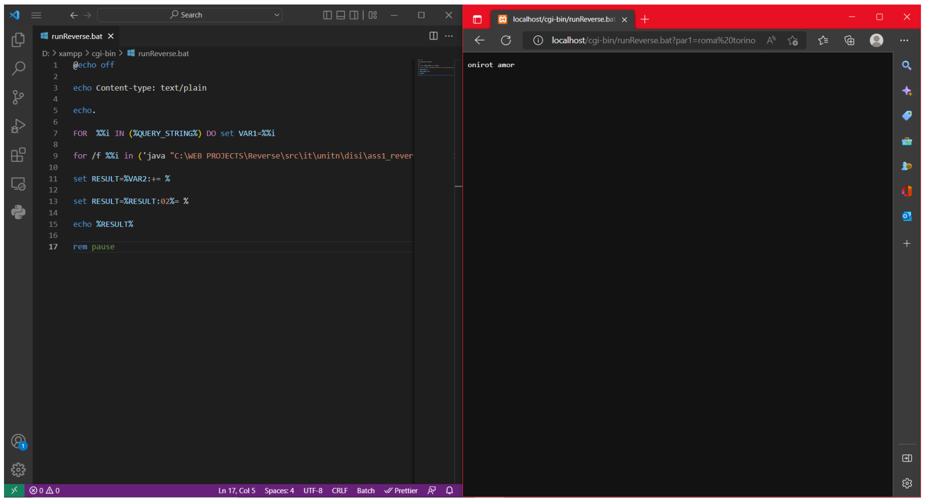
A batch file is written in order to retrieve the GET parameter saved in `QUERY_STRING` and store it in `VAR1` (an environmental variable of the Apache web server).
Subsequently, the `Reverse.java` used in the first part is run passing `VAR1` as argument. The reversed string, output of the java application, is then modified in order to remove `+` and `%20` contained in GET parameters with with spaces in them.

The resulting string is then send to the client.

In order to run the batch file from the client side, the .bat must be stored in the Apache cgi-bin directory, and accessed by the client via URL `http://localhost:80/process/reverse?par1="yourInput"`.

2.2 Screenshots

Following the screenshot of the code and the browser after serchong for `http://localhost:80/process/reverse?par1=roma torino`.



runReverse.bat
D: > xampp > cgi-bin > runReverse.bat
localhost/cgi-bin/runReverse.bat?par1=roma%20torino

```
@echo off

echo Content-type: text/plain

echo.

FOR  %%i IN (%QUERY_STRING%) DO set VAR1=%%i

for /f %%i in ('java "C:\WEB PROJECTS\Reverse\src\it\unitn\disi\ass1_rever

set RESULT=%VAR2:+= %

set RESULT=%RESULT:02%= %

echo %RESULT%

rem pause
```

```
onirot amor
```

Figure 6: Accessing process through Common Gateway Interface