

Homework, Task_5_HTTP

Agenda:

- HTTP request

- HTTP respond

- code snippets

Task 1

- Install any application to work with the HTTP Protocol
- On any web-application containing a form to correct Post Parameters Value when sending a request
- Get 2 results: one sent directly from the site and the second modified through the installed software;
- The results are illustrated with screenshots.

On the web page account.mail.ru I filled out registration form/account name on web page account.mail.ru with existed mail. Server responded with code 200, respond's body is :

```
{"status":200,"htmlencoded":false,"email":"qwerty@mail.ru","body": {"exists":true,"alternatives": "[]"}}
```

Variable *exist* has a value *true*

Screenshot#1.1. POST request from browser

The screenshot shows a browser window with the following details:

- Request URL:** `https://account.mail.ru/api/v1/user/exists`
- Request Method:** `POST`
- Status Code:** `200 OK`
- Response Body:** `{"exists":true}`
- Registration Form Details:**
 - First name: [redacted]
 - Last name: [redacted]
 - Birthdate: [redacted]
 - Sex: Male (radio button selected)
 - Имя аккаунта: `qwerty` (@mail.ru dropdown)
 - Message: `An email account with that address already exists.`
 - Password: [redacted]
 - Alternate email address (optional): [redacted]
 - Message: `An alternative email is required for account recovery. I have a mobile phone`

Content of POST request

```

POST /api/v1/user/exists HTTP/1.1
Host: account.mail.ru
Pragma: no-cache
Origin: https://account.mail.ru
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ru;q=0.8,fr;q=0.7,fy;q=0.6
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: application/json, text/javascript, */*; q=0.01
Cache-Control: no-cache
X-Requested-With: XMLHttpRequest
Cookie: p=0/oAAJF8PAAA; act=e9e0cf2225be43919f480f55643d8281; mrcu=ABB05D314E4F06CEAFDF7CA62D25; c=UE4xXQEAAhTAAUAAAACQAQ;
s=fver=0|ww=1415|wh=695|t=1|dp=2;
i=AQAcRzFdCQATAAgxB4BAR8BARKCARsCADkCAdoCAeoCAVoIAxMIwJAWMJAWCJaA4JAVoKAWMKAwSAUILAbsBCAQBAgABkwIlcyZrAAFtAAFvAAFxAA
F0AAF2AAF4AAAG+AAHCAAhmAHqAAH0AAEQAQErAQAETQFBaqFCQFuAQEEAgEFAgELAgEQAgERAgEVAgFrAgHhAgFbQFbQF1BQG
gBQGkBQGIBQGpBQGIBgFATwHeBAgEAQEEAeECCQEB4gQKBaGuQddBgUCAQBxBwUCAeu9BwgEASABAQ==;
b=rYFAKByQQAMoGNDDsYcwgBAAAI9kACMdMzAnnVvsbAQAA; o:=78:AA==; VID=36mKcF0Rvg1q00000H0Y941q::6648834:0-24bab16;
tmr_detect=0%7C1563539976936
Connection: keep-alive
Referer: https://account.mail.ru/signup/simple
X-Request-Id: 55fc7c-f26-772e-82fe-725cac21f258

email=qwerty%40mail.ru&name=%7B%22first%22%3A%22%22%2C%22last%22%3A%22%22%7D&birthday=%7B%22day%22%3A%22%22%2C%22month%22%3A%22%
22%2C%22year%22%3A%22%22%7D&htmlencoded=false&utm=%7B%22source%22%3A%22%22%2C%22medium%22%3A%22%22%2C%22campaign%22%3A%22%
2C%22term%22%3A%22%22%2C%22content%22%3A%22%22%7D&referrer=https%3A%2F%2Faccount.mail.ru%2Flogin%2F%3Fmode%3Dsimple%26v%3D2.2.0%26a
ccount host%3Daccount.mail.ru%26type%3Dlogin%26initialWidth%3D420%26allow_external%3D1%26success_redirect%3Dhttps%253A%252F%252Fe.mail.ru%252Fmess
ges%252Finbox%253Fback%253D1%26opener%3Dmail.login%26rebranding2018%3D1%26modal%3D1%26parent_url%3Dhttps%253A%252F%252Fe.mail.ru%252Flogi
n

```

I used Postman Version 7.2.2 as a tool for work with HTTP protocol. I changed account name to unexist: **dennistikhomirov**. Server responded with code 200, respond's body is :

```
{
  "status": 200,
  "htmlencoded": false,
  "email": "dennistikhomirov@mail.ru",
  "body": {
    "exists": false,
    "alternatives": [
      "dennistikhomirov@bk.ru", "dennistikhomirov00@mail", "dennistikhomirov@inbox.ru",
      "dennistikhomirov2019@mail.ru", "dennistikhomirov@list.ru", "dennistikhomirov2020@mail.ru",
      "dennistikhomirov.00@mail.ru", "dennistikhomirov.2019@mail.ru"
    ]
}
```

Variable *exist* has a value *false*

Screenshot#1.2. POST request from Postman

The screenshot shows the Postman interface with a successful POST request to `https://account.mail.ru/api/v1/user/exists`. The response status is `200 OK`, time `184 ms`, and size `842 B`. The response body is a JSON object:

```

{
  "status": 200,
  "htmlencoded": false,
  "email": "dennistikhomirov@mail.ru",
  "body": {
    "exists": false,
    "alternatives": [
      "dennistikhomirov@bk.ru",
      "dennistikhomirov00@mail.ru",
      "dennistikhomirov@inbox.ru",
      "dennistikhomirov2019@mail.ru",
      "dennistikhomirov@list.ru",
      "dennistikhomirov2020@mail.ru",
      "dennistikhomirov.00@mail.ru",
      "dennistikhomirov.2019@mail.ru"
    ]
}

```

Task 2

Customer reported the issue. It's unable to open <https://affinion1.datalex.com/travel>, but <https://affinion1.datalex.com/travel/> works fine. What's the root cause by the log below?

```
CONNECT affinion1.datalex.com:443 HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; InfoPath.2; .NET CLR
3.0.04506.648)
Host: affinion1.datalex.com
Content-Length: 0
Proxy-Connection: Keep-Alive
Pragma: no-cache

HTTP/1.1 200 Connection Established Timestamp: 16:21:39:8089
This is a HTTPS CONNECT Tunnel. Secure traffic flows through this
connection.
Secure Protocol: Tls
Cipher: Rc4 128bits
Hash Algorithm: Md5 128bits Key Exchange: RsaSign 1024bits
== Client Certificate =====
None.
== Server Certificate =====
[Subject]
CN=testcert, OU=Security, O=Cisco Systems Inc, L=San Jose,
S=California, C=US
[Issuer]
CN=testca.cisco.com, OU=Security, O=Cisco Systems Inc, L=San Jose,
S=California, C=US
[Serial Number]
01
[Not Before]
11/11/2003 11:56:23 PM
[Not After]
11/8/2013 11:56:23 PM
[Thumbprint] 99300D5B4323A5B8FBA1DF39402051355646EAE2

-----
GET /travel HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/xaml+xml,
application/vnd.ms-xpsdocument, application/x-ms-xbap, application/x-
ms-application, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; InfoPath.2; .NET CLR
3.0.04506.648)
Host: affinion1.datalex.com
Connection: Keep-Alive
Cookie: JSESSIONID=B240C1F1AF396E6BBA26C88AEA012292; AF-
COOKIE=r2572537665
```

```

HTTP/1.0 301 Moved Permanently
Set-Cookie: AF-COOKIE=r2572537665; path=/ Date: Fri, 12 Sep 2008
13:21:39 GMT
Server: Apache/2.0.46 (Red Hat)
Location: http://affinion1.datalex.com/travel/ Content-Length: 334
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://affinion1.datalex.com/travel/">here</a>.</p>
<hr />
<address>Apache/2.0.46 (Red Hat) Server at affinion1.datalex.com Port
80</address>
</body></html>
-----
```

GET /travel/ HTTP/1.1

Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg,
 application/x-shockwave-flash, application/xaml+xml,
 application/vnd.ms-expsdocument, application/x-ms-xbap, application/x-
 ms-application, application/vnd.ms-excel, application/vnd.ms-
 powerpoint, application/msword, */*

Accept-Language: en-us

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
 .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; InfoPath.2; .NET CLR
 3.0.04506.648)

Host: affinion1.datalex.com

Proxy-Connection: Keep-Alive

Cookie: brand_url=mycompany.test.trilegiantloyalty.com/ip-mc;
 JSESSIONID=B240C1F1AF396E6BBA26C88AEA012292; AF-COOKIE=r2572537665

```

HTTP/1.1 502 Fiddler - Connection Failed Content-Type: text/html
Connection: close
Timestamp: 16:22:00:9968
[Fiddler] Connection to affinion1.datalex.com failed.<BR>Exception
Text: A connection attempt failed because the connected party did not
properly respond after a period of time, or established connection
failed because connected host has failed to respond 216.203.24.99:80
-----
```

ANSWER.

After first GET request user gets the response

HTTP/1.0 301 Moved Permanently

It means that resource was moved to new location that pointed in LOCATION header:

Location: <http://affinion1.datalex.com/travel/>

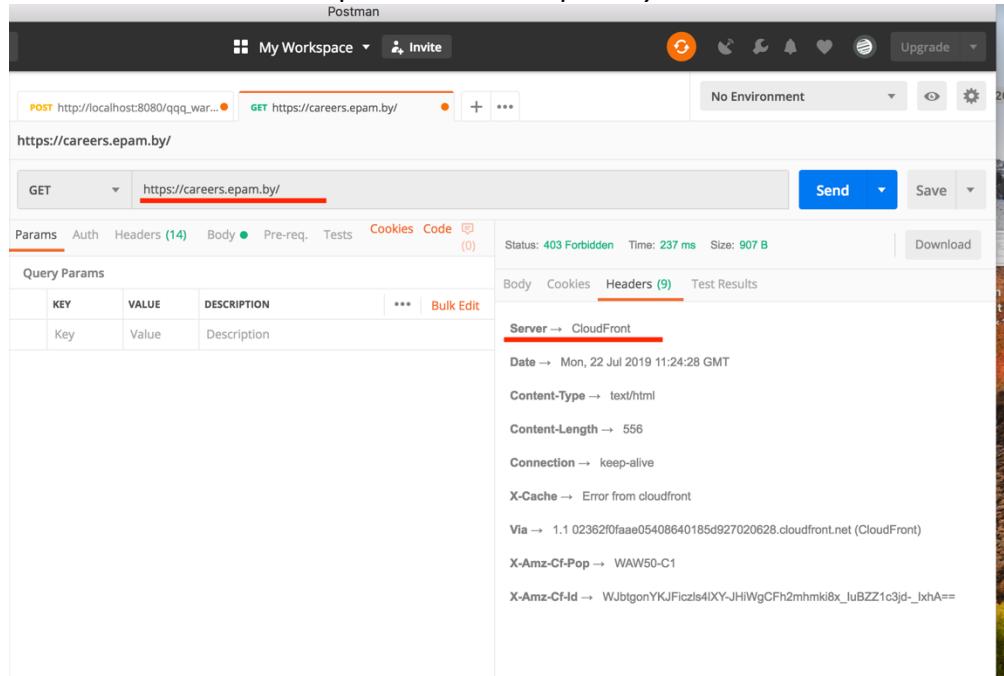
That's why request to <http://affinion1.datalex.com/travel/> should be successful. But it was failed again because host didn't respond. Probably web app was removed or server was broken.

Task 3

There are many web servers such as Apache, nginx, IIS, etc. Using HTTP, find out which web server is being used epam.by, gstu.by I wikipedia.org?

1. Request to <http://epam.by> always redirected to <https://careers.epam.by/>
<https://careers.epam.by/> uses CloudFront server

Screenshot#3.1. POST request to *careers.epam.by*



Postman

POST http://localhost:8080/qqq_war... ● GET https://careers.epam.by/ ● + ... No Environment

https://careers.epam.by/

GET https://careers.epam.by/ Send Save

Params Auth Headers (14) Body Pre-req. Tests Cookies Code (0)

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 403 Forbidden Time: 237 ms Size: 907 B

Body Cookies Headers (9) Test Results

Server → CloudFront

Date → Mon, 22 Jul 2019 11:24:28 GMT

Content-Type → text/html

Content-Length → 556

Connection → keep-alive

X-Cache → Error from cloudfront

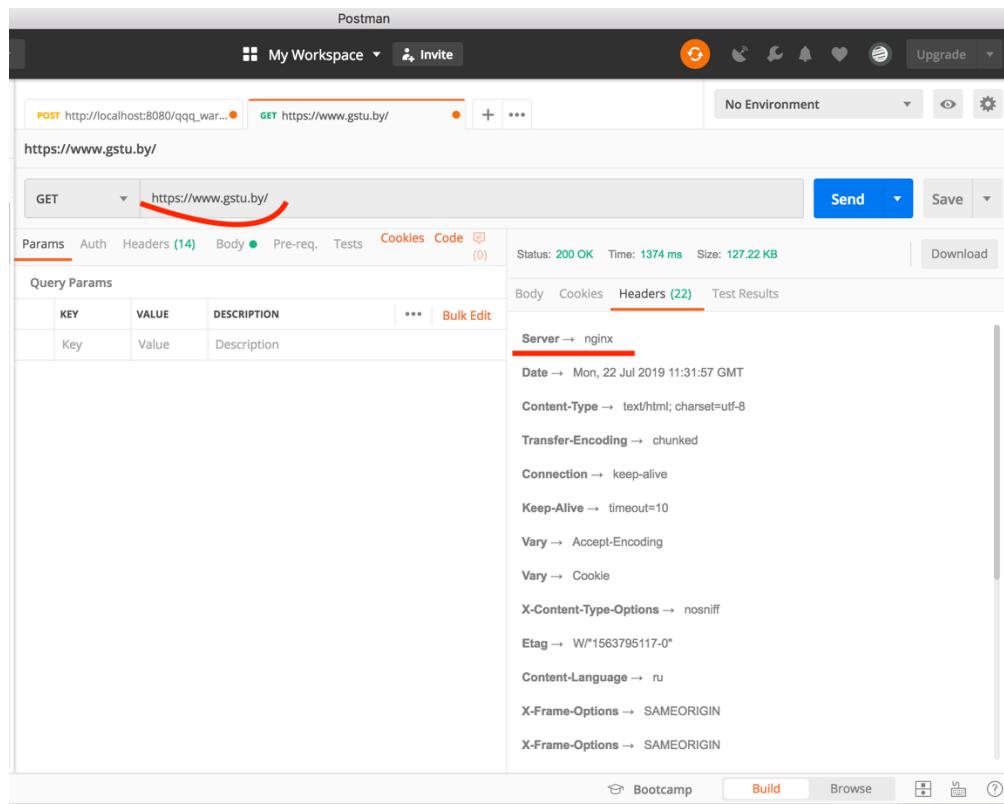
Via → 1.1 02362f0faae05408640185d927020628.cloudfront.net (CloudFront)

X-Amz-Cf-Pop → WAW50-C1

X-Amz-Cf-Id → WJbtgonYKJFiczs4IXY-JHiWgCFh2mhmk8x_luBZZ1c3jd-_lxhA==

2. Gstu.by uses [nginx](#)

Screenshot#3.2. POST request to *gstu.by*



Postman

POST http://localhost:8080/qqq_war... ● GET https://www.gstu.by/ ● + ... No Environment

https://www.gstu.by/

GET https://www.gstu.by/ Send Save

Params Auth Headers (14) Body Pre-req. Tests Cookies Code (0)

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 1374 ms Size: 127.22 KB

Body Cookies Headers (22) Test Results

Server → nginx

Date → Mon, 22 Jul 2019 11:31:57 GMT

Content-Type → text/html; charset=utf-8

Transfer-Encoding → chunked

Connection → keep-alive

Keep-Alive → timeout=10

Vary → Accept-Encoding

Vary → Cookie

X-Content-Type-Options → nosniff

Etag → W/"1563795117-0"

Content-Language → ru

X-Frame-Options → SAMEORIGIN

X-Frame-Options → SAMEORIGIN

3. Wikipedia.org uses mw1328.eqiad.wmnet

Screenshot#3.1. POST request to *Wikipedia.org*

The screenshot shows the Postman application interface. A GET request is made to <https://www.wikipedia.org/>. The response status is 200 OK, time is 360 ms, and size is 19.12 KB. The Headers tab is selected, showing various HTTP headers. The 'Server' header is highlighted with a red box and contains the value 'mw1328.eqiad.wmnet'. Other visible headers include Date, Content-Type, Content-Length, Connection, Cache-Control, ETag, Last-Modified, Backend-Timing, Content-Encoding, Vary, X-Varnish, and Age.

Header	Value
Date	Mon, 22 Jul 2019 11:34:40 GMT
Content-Type	text/html
Content-Length	18687
Connection	keep-alive
Server	mw1328.eqiad.wmnet
Cache-Control	s-maxage=86400, must-revalidate, max-age=3600
ETag	W/"129d0-58db7caf28a4a"
Last-Modified	Mon, 15 Jul 2019 12:56:46 GMT
Backend-Timing	D=333 t=1563700447331418
Content-Encoding	gzip
Vary	X-Seven, Accept-Encoding
X-Varnish	49811138 24447129, 930426636 937238243, 1073687126 838548166
Age	10285

Task 4

Give the name and version of the browser you are using. Which User-Agent sends your browser to the web servers? Is it possible to send another User-Agent using the same browser?

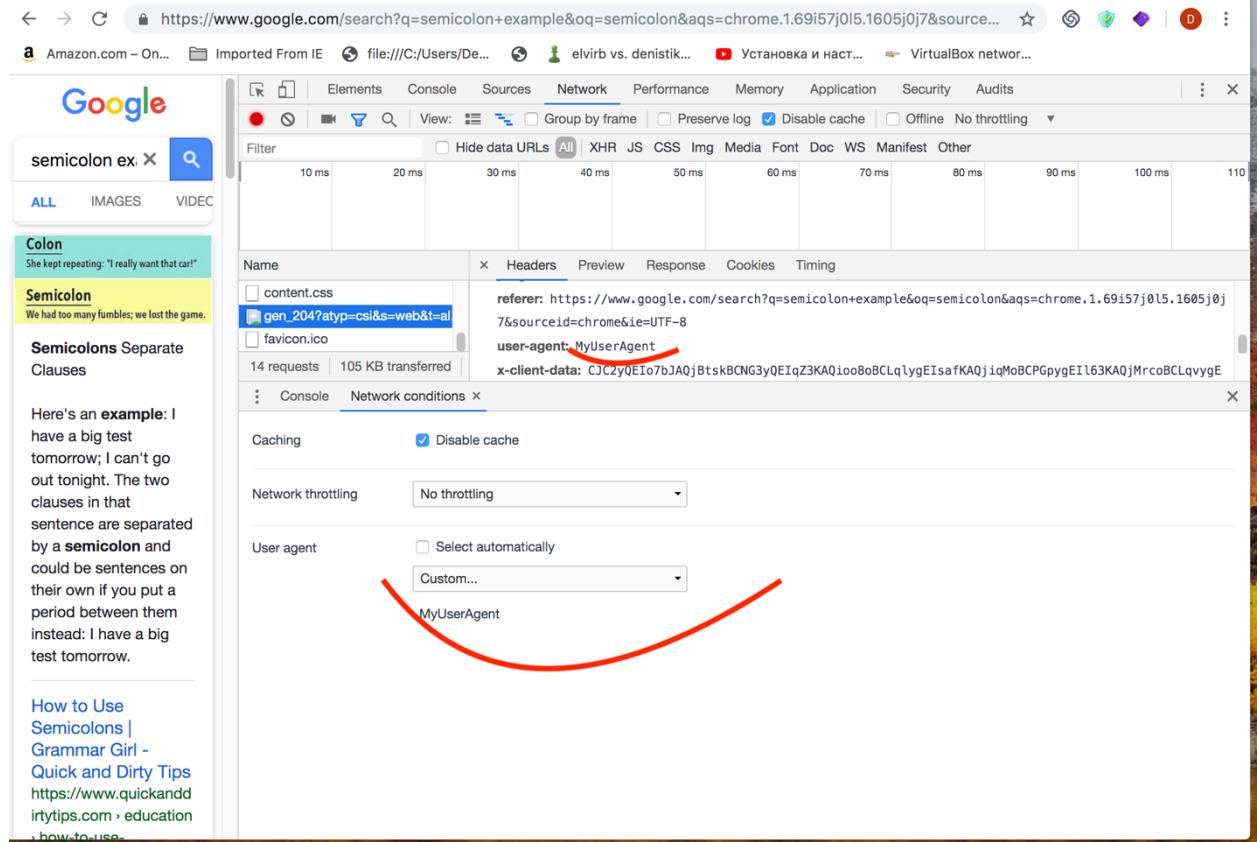
1. I am using Chrome Version 75.0.3770.142 (Official Build) (64-bit)
2. By default Chrome browser's User-Agent is:

Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/75.0.3770.142 Safari/537.36

3. Yes, it is possible to change User-Agent for Chrome. Open *DevelopersTools->3dotmenu->More tools->Network condition* and Network condition tab appears in Drawer window. Then unselect checkbox 'Select automatically' and choose desirable User-Agent.

Screenshot#4.1. Custom User-Agent in HTTP request



Task 5

Why after the 2nd request the user was not logged in?

```
GET /web/login.jsp HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/xaml+xml, 
application/vnd.ms-xpsdocument, application/x-ms-xbap, application/x-
ms-application, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */
Accept-Language: en-us 
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; InfoPath.2; .NET CLR
3.0.04506.648)
Host: localhost
Proxy-Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Date: Thu, 18 Sep 2008 16:17:36 GMT
Server: Apache-Coyote/1.1
Content-Type: text/html
Content-Length: 84
Set-Cookie: JSESSIONID=ED08EBA3C4A4ADA74C48365ED0324A4A.tomcat1;
Path=/app
```

```
Node: <b>first</b><br/>
Session: <b>null</b>
```

```
<a href="action.jsp">action.jsp</a>
```

```
GET /web/action.jsp HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/xaml+xml, 
application/vnd.ms-xpsdocument, application/x-ms-xbap, application/x-
ms-application, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */
Referer: http://localhost/web/login.jsp
Accept-Language: en-us 
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; InfoPath.2; .NET CLR
3.0.04506.648)
Host: localhost
Proxy-Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Date: Thu, 18 Sep 2008 16:17:41 GMT
Server: Apache-Coyote/1.1
Content-Type: text/html
Content-Length: 66
Set-Cookie: JSESSIONID=A0C56C2678CF7607BFF538240D7EA7E8.tomcat1;
Path=/app
```

```
You are not logged, sorry.
```

- After the first request user was successfully logged in and server respond contained SET-COOKIE header:

Set-Cookie: JSESSIONID=ED08EBA3C4A4ADA74C48365ED0324A4A.tomcat1; Path=/app
--

Set-cookie header defined by RFC6265 <https://tools.ietf.org/html/rfc6265>

- If user-agent wants to be automatically logged in without login process then User-agent MUST include COOKIE header in next HTTP request as described here <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cookie>

For example:

Cookie: JSESSIONID=ED08EBA3C4A4ADA74C48365ED0324A4A.tomcat1;

Also a URL path “/app” must exist in the requested resource before sending the Cookie header.

ANSWER. User isn't logged after 2nd request because this request doesn't contain neither COOKIE header nor resource path “/app” for these cookies .

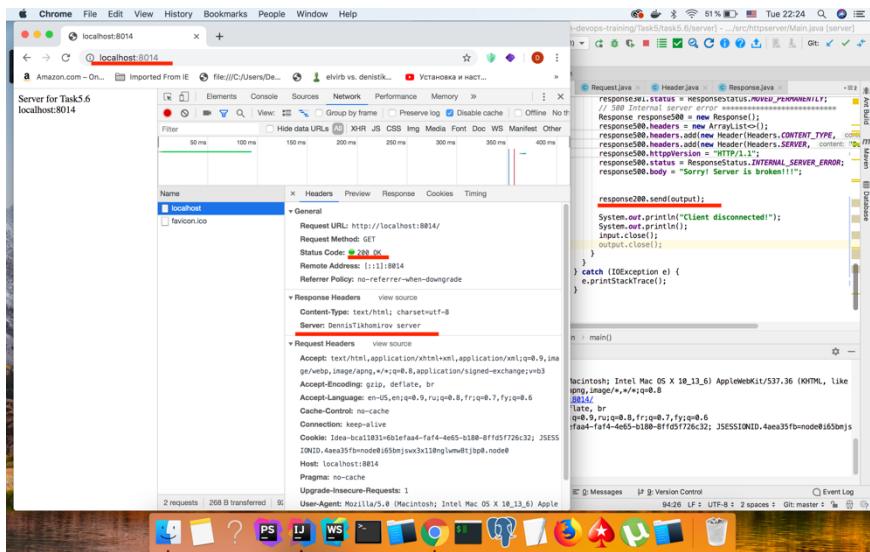
Task 6

- Create a simple console web server based on TCP/IP sockets.

Java code available here <https://github.com/dennis00010011b/epam-devops-training/tree/master/Task5/task5.6/server/src/httpserver>

- From any browser make requests to this server
- As response to use all of the status groups
- Demonstrate the content of HTTP requests and HTTP responses
- Demonstrate the results (screenshots)

6.1 Response with status 200



Request

```
GET / HTTP/1.1
Host: localhost:8014
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ru;q=0.8,fr;q=0.7,fy;q=0.6
Cookie: Idea-bca11031=6b1efaa4-faf4-4e65-b180-8ffd5f726c32;
JSESSIONID.4aea35fb=node0i65bmjswx3x110nglwmw8tjbp0.node0
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Server: DennisTikhomirov server

<p>Server for Task5.6 <br> localhost:8014</p>
```

6.2 Response with status 301

The screenshot shows a browser window with the URL `localhost:8014` and a developer tools Network tab. The Network tab lists a single request to `localhost:8014/moved/` with a status code of `301 Moved Permanently`. The response headers include `Content-Type: text/html; charset=utf-8` and `Location: localhost:8014/moved/`. The response body contains the text "Client disconnected!". To the right, an IDE (IntelliJ IDEA) displays the `Main.java` file, which contains Java code for handling a 301 response. The code includes logic to set the status to `Moved`, create a `Response` object, add a `Location` header, and send the response body.

```
response301.status = ResponseStatus.MMOVED;
// 500 Internal server error *****
Response response500 = new Response();
response500.headers = new ArrayList<>();
response500.headers.add(new Header(Headers.CONTENT_TYPE, "text/html"));
response500.headers.add(new Header(Headers.LOCATION, "localhost:8014/moved/"));
response500.httpVersion = "HTTP/1.1";
response500.status = ResponseStatus.INTERNAL_ERROR;
response500.body = "Sorry! Server is broken!";
response301.send(output);
System.out.println("Client disconnected!");
System.out.println();
input.close();
output.close();
}
} catch (IOException e) {
e.printStackTrace();
}
}

Main > main()
```

Request

```
GET / HTTP/1.1
Host: localhost:8014
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ru;q=0.8,fr;q=0.7,fy;q=0.6
Cookie: Idea-bca11031=6b1efaa4-faf4-4e65-b180-8ffd5f726c32; JSESSIONID.4aea35fb=node0i65bmjswx3x110nglwmw8tjbp0.node0
Host: localhost:8014
Pragma: no-cache
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 301 Moved Permanently
Content-Type: text/html; charset=utf-8
Location: localhost:8014/moved/
Server: DennisTikhomirov server
```

6.3 Response with status 404

The screenshot shows a browser window with the URL `localhost:8014`. The Network tab of the developer tools is open, showing a single request to `localhost` with a status of `404 Not Found`. The Java code in the background shows a `response404.send(output);` line highlighted.

```
    response404.status = ResponseStatus.MOVE_TO_ROOT;
    // 500 Internal server error *****
    Response response500 = new Response();
    response500.headers = new ArrayList<>();
    response500.headers.add(new Header(Headers.CONTENT_TYPE, "text/html"));
    response500.headers.add(new Header(Headers.CONTENT_LENGTH, String.valueOf(response500.length())));
    response500.httpVersion = "HTTP/1.1";
    response500.status = ResponseStatus.INTERNAL_SERVER_ERROR;
    response500.body = "Sorry! Server is broken!";
    response404.send(output);
    System.out.println("Client disconnected!");
    System.out.println();
    input.close();
    output.close();
}
} catch (IOException e) {
    e.printStackTrace();
}
}
```

Request

```
GET / HTTP/1.1
Host: localhost:8014
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ru;q=0.8,fr;q=0.7,fy;q=0.6
Cookie: Idea-bca11031=6b1efaa4-faf4-4e65-b180-8ffd5f726c32; JSESSIONID.4aea35fb=node0i65bmjswx3x110nglwmw8tjbp0.node0
Host: localhost:8014
Pragma: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
```

Response

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=utf-8
Server: DennisTikhomirov server

Page NOT FOUND (
```

6.4 Response with status 500

The screenshot shows a browser window at localhost:8014 displaying the error message "Sorry! Server is broken!!!". Below the message is a detailed network request and response analysis. To the right is an IDE (IntelliJ IDEA) showing the Java code for the server. The code defines a Main class with a main() method that handles requests and sends a 500 Internal Server Error response with the message "Sorry! Server is broken!!!". The code also includes logging for client disconnection.

```
response500.headers = new ArrayList<Header>();
response500.headers.add(new Header(Headers.CONTENT_TYPE, "text/html; charset=UTF-8"));
response500.headers.add(new Header(Headers.CONTENT_LENGTH, String.valueOf(response500.body.length())));
response500.httpVersion = "HTTP/1.1";
response500.status = ResponseStatus.INTERNAL_SERVER_ERROR;
response500.body = "Sorry! Server is broken!!!";

response500.send(output);
System.out.println("Client disconnected!");
System.out.println();
input.close();
output.close();
}

} catch (IOException e) {
e.printStackTrace();
}
}
```

Request

```
GET / HTTP/1.1
Host: localhost:8014
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ru;q=0.8,fr;q=0.7,fy;q=0.6
Cookie: Idea-bca11031=6b1efaa4-faf4-4e65-b180-8ffd5f726c32; JSESSIONID.4aea35fb=node0i65bmjswx3x110nglwmw8tjbp0.node0
5f726c32; JSESSIONID.4aea35fb=node0i65bmjswx3x110nglwmw8tjbp0.node0
nqlwmw8tjbp0.node0
```

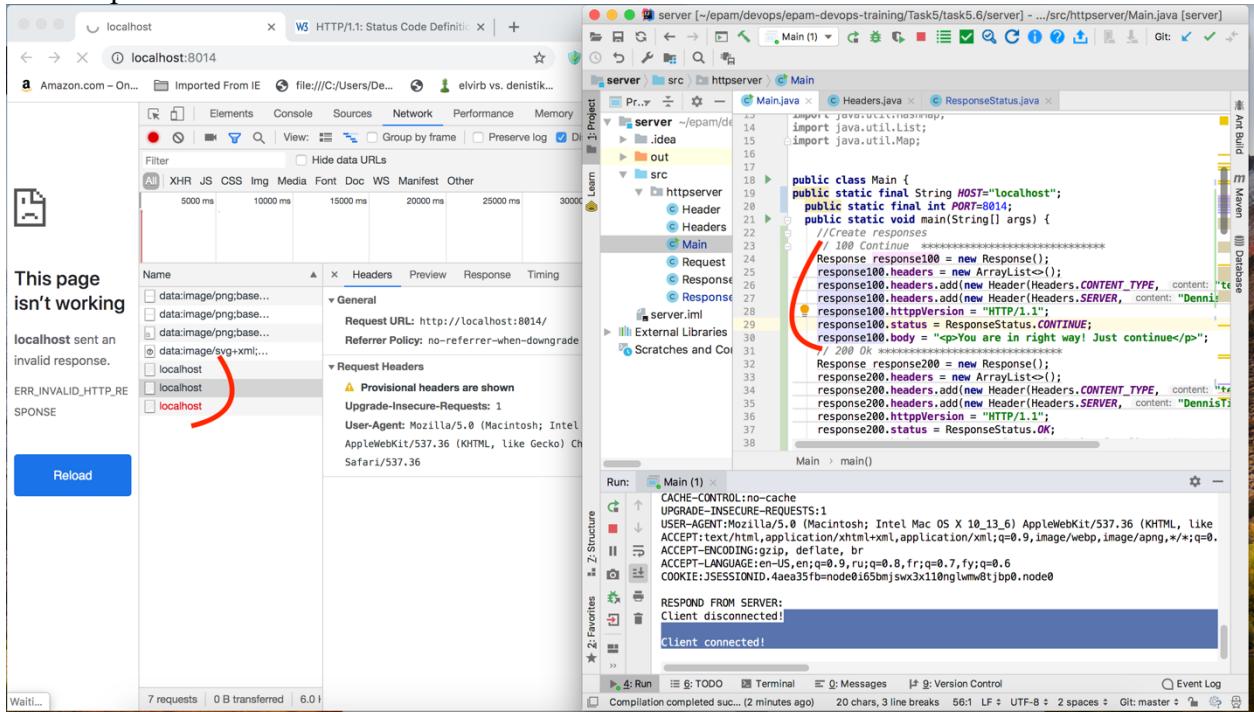
Response

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Server: DennisTikhomirov server

Sorry! Server is broken!!!
```

6.5 Response with status 100

According <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html> the client(Google Chrome) is ignoring responds with status 100, repeating requests and waiting final respond with status 200.



Request

```
GET / HTTP/1.1
Host: localhost:8014
Upgrade-Insecure-Requests:
1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
```

Response

```
HTTP/1.1 100 Continue
Content-Type: text/html; charset=utf-8
Server: DennisTikhomirov server

"<p>You are in right way! Just continue, baby</p>"
```

Task 7

- Create a simple HTML document containing a web form to enter a user name

```
<html>
<head>
    <title>$Title$</title>
</head>
<body>
    <form action="FormServlet" method="get">
        Data will be transferred with GET method
        <br>
        Enter your name and press ENTER:
        <input type="text" name="username">
        <br>
    </form>
</body>
</html>
```

- Create a backend part to process the data entered on the form. Output to the browser greeting "Hello, {username}!"

Project's code is available here <https://github.com/dennis00010011b.epam-devops-training/tree/master/Task5/task5.7>

Servlet for processing GET and POST requests for main page "/"

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "FormServlet", urlPatterns = {"/"})
public class FormServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("It was POST request");
        response.setContentType("text/html");
        try {
            request.getRequestDispatcher("/post.jsp").include(request, response);
        } catch (ServletException e) {
            e.printStackTrace();
        }
        PrintWriter out = response.getWriter();

        out.println(this.getGreeting(request));
        out.flush();
        System.out.println(request.getParameter("username"));
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("It was GET request");
        response.setContentType("text/html");

        try {
```

```

        request.getRequestDispatcher("/get.jsp").include(request, response);
    } catch (ServletException e) {
        e.printStackTrace();
    }
    PrintWriter out = response.getWriter();
    out.println(this.getGreeting(request));
    out.flush();
    System.out.println(request.getParameter("username"));
}

private String getGreeting(HttpServletRequest request) {
    String username =
        request.getParameter("username") == null ? "BRO" :
    request.getParameter("username");
    return "<h2>Hello " + username.toUpperCase() + " !</h2> " +
        "\n" + "Username was transferred with " + request.getMethod() + " method";
}

protected void doPut(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("It was PUT request");
}
}

```

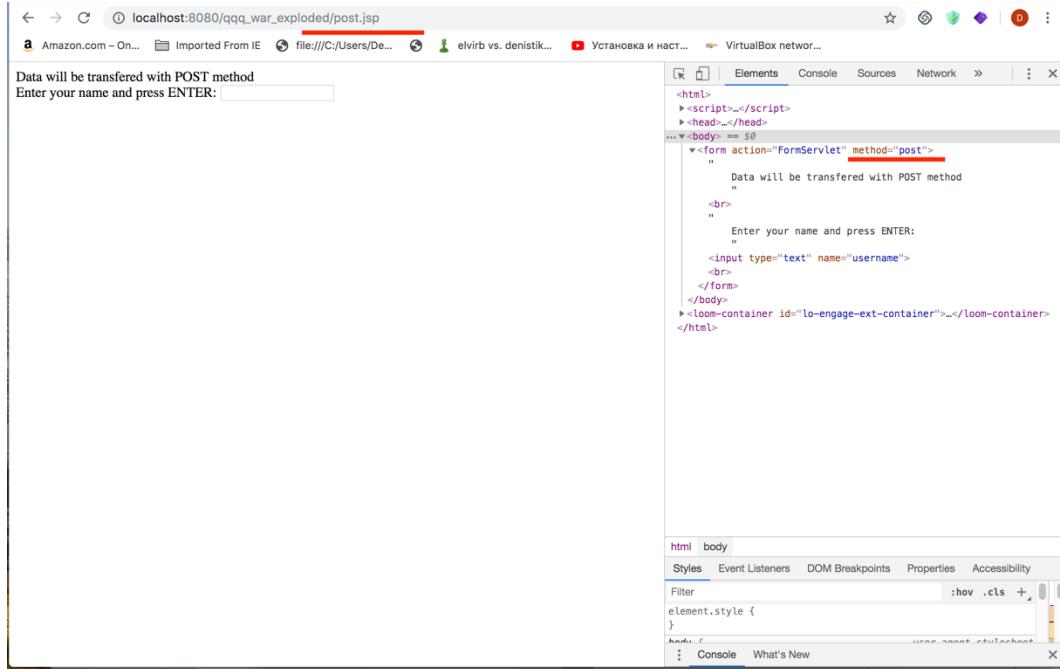
- Provide the ability to transfer parameters from the HTML form to the server in various request methods

User can select a transfer method by URL.

/post.jsp for processing with POST request

/get.jsp for processing with GET request

Screenshot #7.1. HTML form with POST request method



Screenshot #7.2. HTML form with GET request method

The screenshot shows a browser window with the URL `localhost:8080/qqq_war_exploded/get.jsp`. The page content is:

```
<html>
<script></script>
<head></head>
<body> == $0
<form action="FormServlet" method="get">
    "Data will be transferred with GET method"
    "Enter your name and press ENTER:"
    <input type="text" name="username">
    <br>
</form>
</body>
</loom-container id="lo-engage-ext-container"></loom-container>
</html>
```

The developer tools sidebar shows the HTML structure with the `method="get"` attribute highlighted.

Screenshot #7.3. Respond from server (printed "Hello \$username")

The screenshot shows a browser window with the URL `localhost:8080/qqq_war_exploded/FormServlet`. The page content is:

```
<html>
<script></script>
<head></head>
<body> == $0
<form action="FormServlet" method="post"></form>
<h2>Hello DEVOPS !</h2>
<p>Username was transferred with POST method</p>
</body>
</loom-container id="lo-engage-ext-container"></loom-container>
</html>
```

The developer tools sidebar shows the HTML structure with the `method="post"` attribute highlighted. A red arrow points from the text "Hello DEVOPS !" to the word "DEVOPS".

- Upload the created website to the web server

WAR archive placed here https://github.com/dennis00010011b.epam-devops-training/tree/master/Task5/task5.7/out/artifacts/artf_archive2

I still can't get access to a cloud that's why I can't deploy server to the web

- Test its functionality
- Display all HTTP requests and HTTP responses
- Demonstrate the results (screenshots)