**Dennis Tikhomirov DevOps training, Spring’19**

**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;**

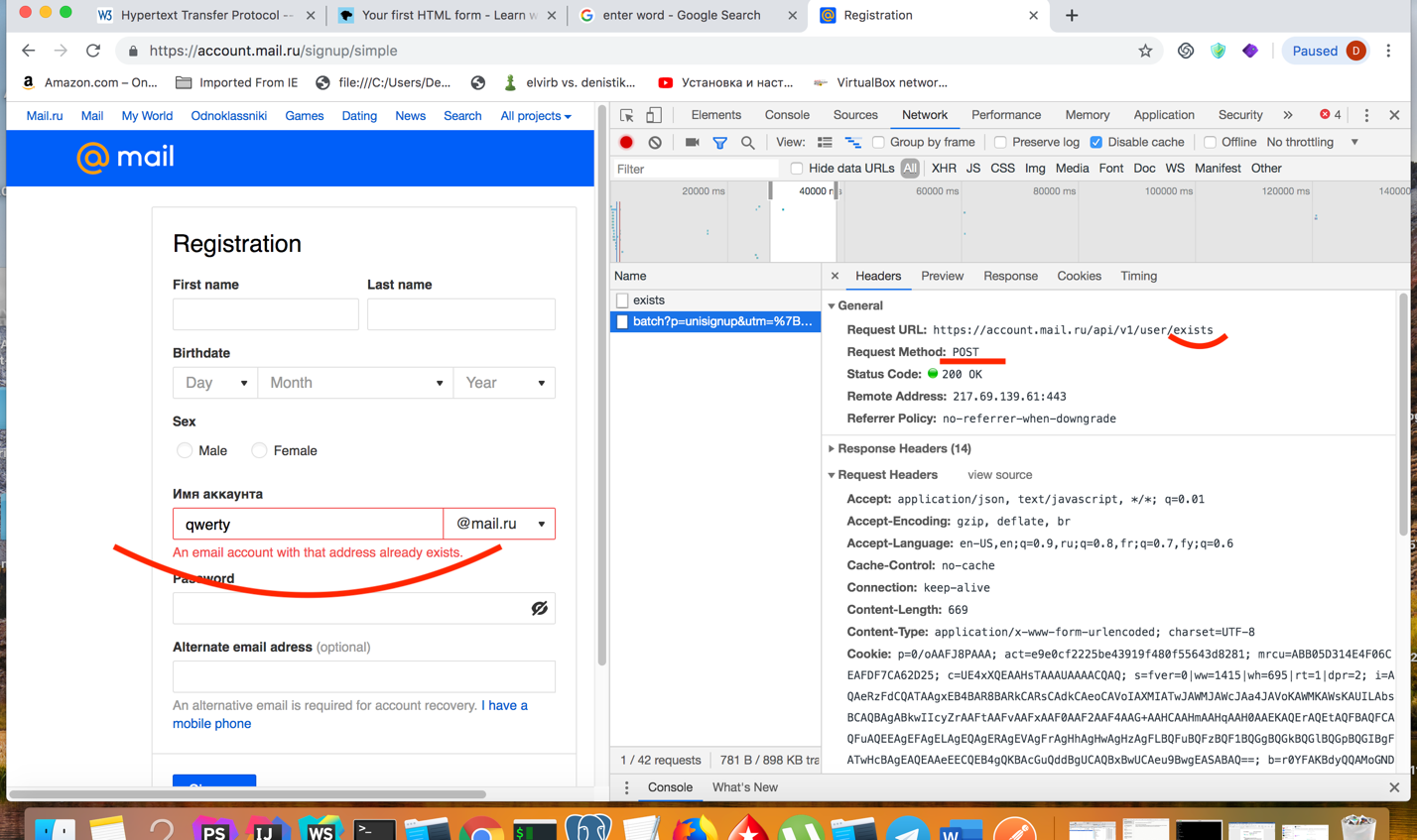
* **The results are illustrated with screenshots.**

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



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/oAAFJ8PAAA; act=e9e0cf2225be43919f480f55643d8281; mrcu=ABB05D314E4F06CEAFDF7CA62D25; c=UE4xXQEAAHsTAAAUAAAACQAQ; s=fver=0|ww=1415|wh=695|rt=1|dpr=2; i=AQAeRzFdCQATAAgxEB4BAR8BARkCARsCAdkCAeoCAVoIAXMIATwJAWMJAWcJAa4JAVoKAWMKAWsKAUILAbsBCAQBAgABkwIIcyZrAAFtAAFvAAFxAAF0AAF2AAF4AAG+AAHCAAHmAAHqAAH0AAEKAQErAQEtAQFBAQFCAQFuAQEEAgEFAgELAgEQAgERAgEVAgFrAgHhAgHwAgHzAgFLBQFuBQFzBQF1BQGgBQGkBQGlBQGpBQGIBgFATwHcBAgEAQEAAeEECQEB4gQKBAcGuQddBgUCAQBxBwUCAeu9BwgEASABAQ==; b=r0YFAKBdyQQAMoGNDDsYcwgBAAAIZ9kACMdMzAnnVvsbAQAA; o=:78:AA==; VID=36mKcF0Rvg1q00000H0Y941q::6648834:0-0-24bab16; 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%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%26account\_host%3Daccount.mail.ru%26type%3Dlogin%26initialWidth%3D420%26allow\_external%3D1%26success\_redirect%3Dhttps%253A%252F%252Fe.mail.ru%252Fmessages%252Finbox%253Fback%253D1%26opener%3Dmail.login%26rebranding2018%3D1%26modal%3D1%26parent\_url%3Dhttps%253A%252F%252Fe.mail.ru%252Flogin

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,

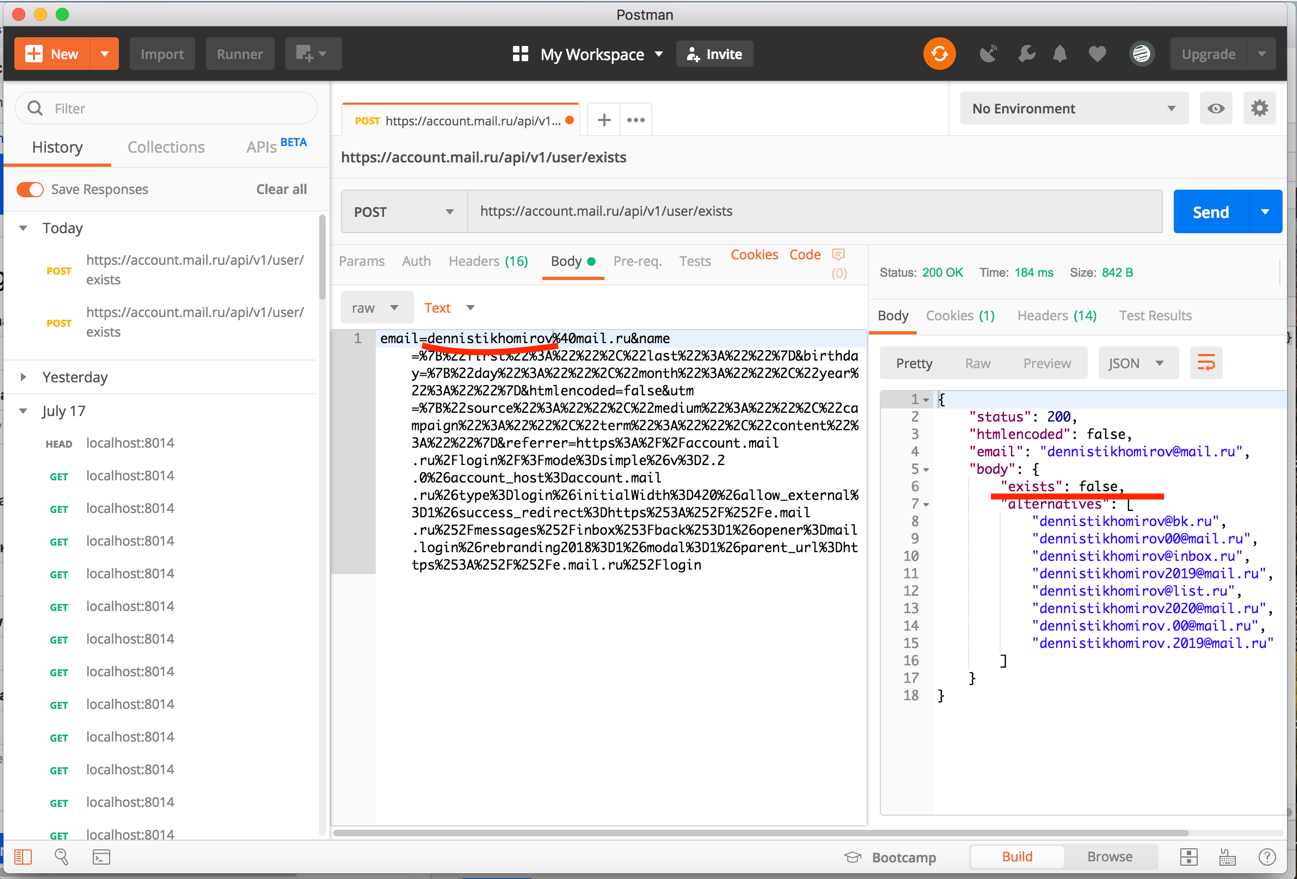
"email": "dennistikhomirov@mail.ru",

"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

**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-xbitmap, 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-xbitmap, 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  
**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

------------------------------------------------------------------

**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?**

**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?**

**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-xbitmap, 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-xbitmap, 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.

------------------------------------------------------------------

1. 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>

1. 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 Indicates 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” .

**Task 6**

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

**•  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)**

**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 transfered 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 transfered 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**

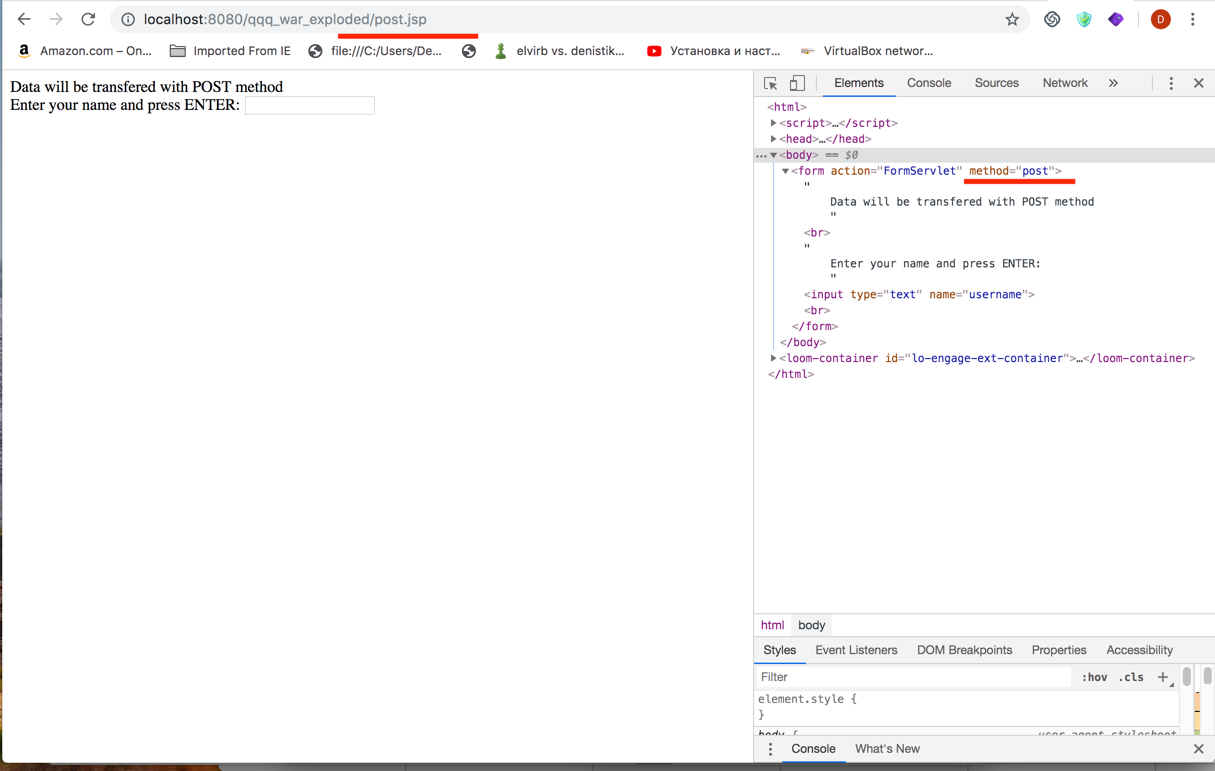
**various request methods**

User can select a transfer method by URL.

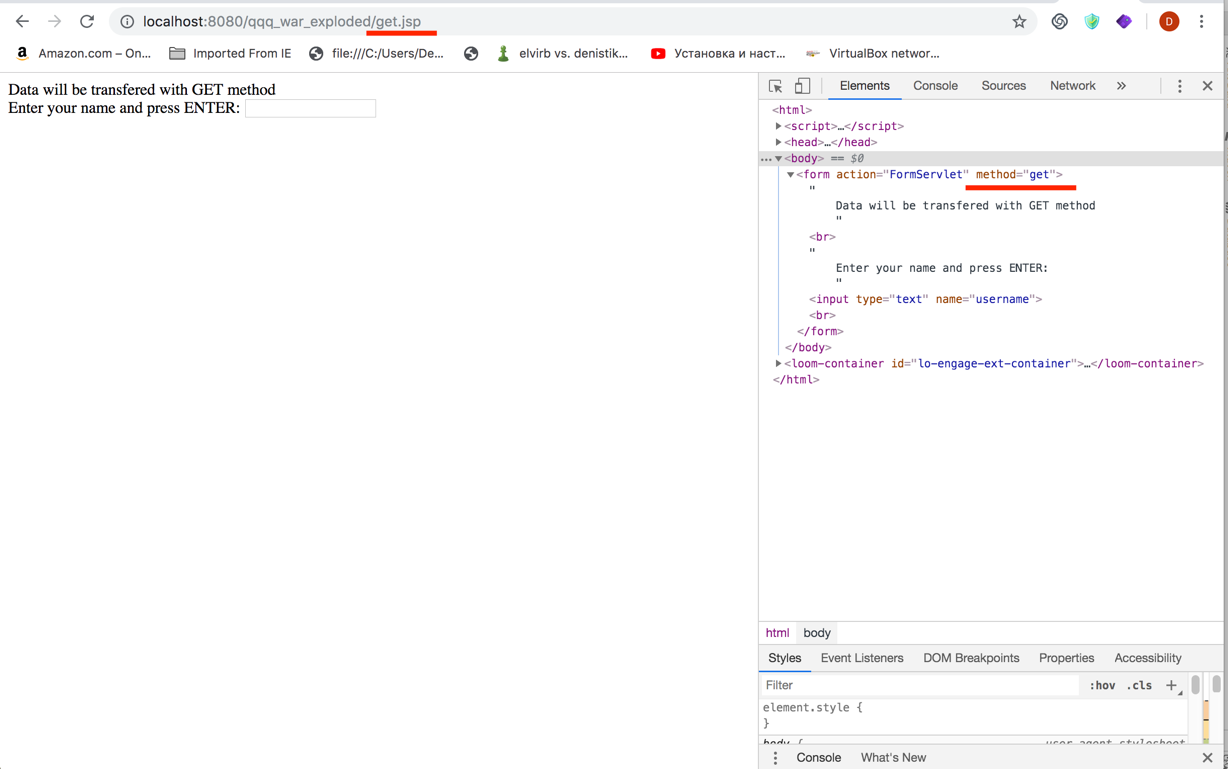
*/post.jsp* for processing with POST request

*/get.jsp* for processing with GET request

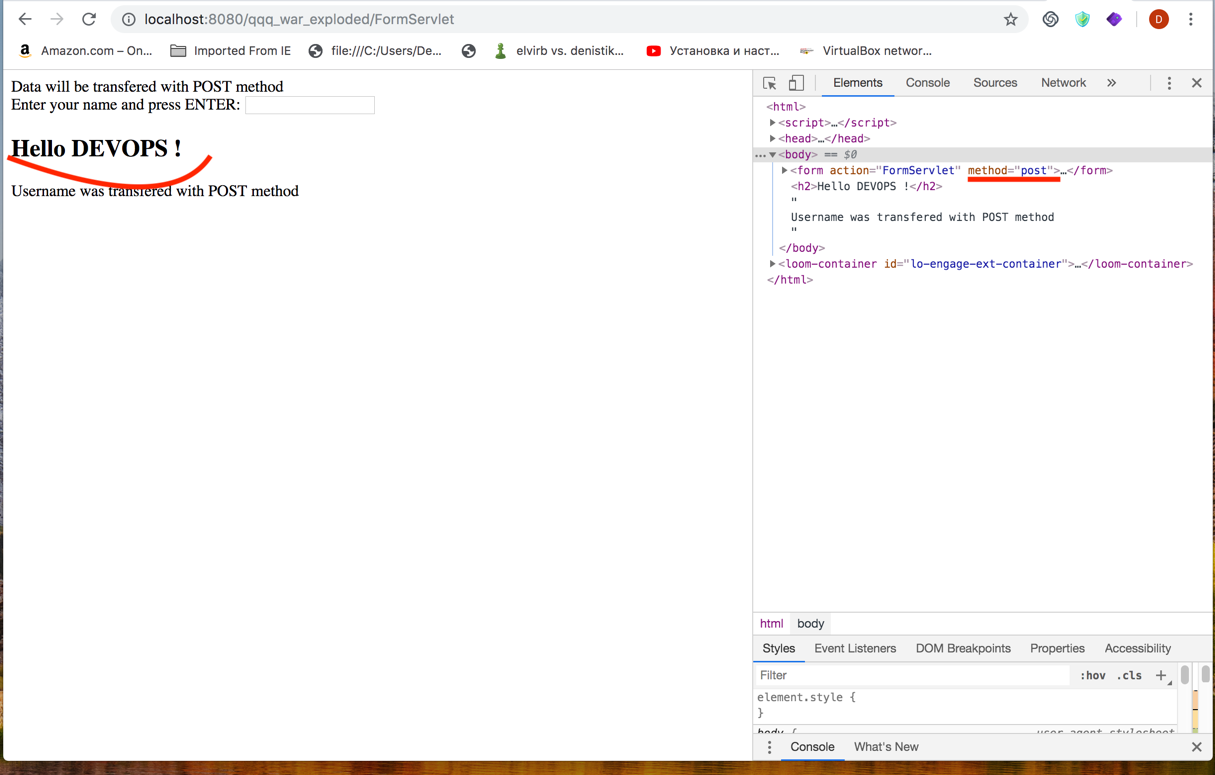
Sceenshot #7.1. HTML form with POST request method



Sceenshot #7.2. HTML form with GET request method



Sceenshot #7.3. Respond from server (printed “Hello $username”)



**•  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>

**•  Test its functionality**

**•  Display all HTTP requests and HTTP responses**

**•  Demonstrate the results (screenshots)**