**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică și Microelectronică**

**Departamentul** **Ingineria Software şi Automatică**

**Raport**

**Lucrare de laborator nr.3**

Disciplina: Programarea în rețea

Tema: HTTP Client

A îndeplinit: st.gr.TI-171 Verhovețchi Daniel

A controlat: Buldumac Oleg

Chișinău 2020

**Scopul lucrării:**

Să se creeze o aplicație client HTTP

**Răspunsuri la întrebări:**

1. Cum este formatat corpul unei cereri HTTP pentru o cerere HTTP de tip POST?

Defaul este MIME-ul ” application/x-www-form-urlencoded”. Acesta este un format pentru codificarea perechilor cheie-valoare cu chei posibil duplicate. Exemplu:

Name=Gareth+Wylie&Age=24&Formula=a+%2B+b+%3D%3D+13%25%21.

1. De unde știe un client HTTP ce tip de conținut trimite serverul HTTP?

Răspunsul de la cerere conține antetul – Content-Type, ceia ce arată tipul de conțint.

1. Cum decide un client dacă ar trebui să aibă încredere în certificatul unui server?

Clientul descarcă certificatul unui server, acesta conține cheia publică a serverului. Acest certificat este semnat cu cheia privată a unei autorități de certificare (AC). Clientul utilizează o cheie publică pentru a verifica dacă întradevăr cerficatul este emis de o AC. Certificatul conține adresa IP a serverului, iar clientul verifică dacă IP – urile coincid. Dacă IP- urile coincid și dacă certificatul încă nu a expirat, atunci clientul are încredere în certificatul respectiv.

1. Care este problema principală cu certificatele autosemnate?

Ele nu asigură securitate, astfel datele transferate devin vulnerabile.

1. Conexiunea persistentă HTTP – care sunt principalele beneficii?

Se elimină necesitatea de a forma handshake- ul de fiecare dată cînd se trimite o cerere, iar asta duce la câștig în timp și nu duce la supraîncărcarea rețelei.

1. Ce este negocierea conținutului în HTTP și cum are loc?

Negocierea conținutului este un mecanism, prin care clientul și serverul comunică pentru a determina care reprezentare a resursei existente pe server să fie transmisă către client.

1. Care sunt tipurile de negociere a conținutului HTTP și cum au loc?

Există două tipuri de negociere a conținutului: negociere proactivă și negociere reactivă. Negocierea proactivă este metoda standart de a obține resursa și constă în specificarea câtorva câmpuri în cererea către server care ar descrie reprezentarea dorită, iar serverul utilizând un algoritm, decide care reprezentare a resursei să o trimită. Negocierea reactivă constă din mai multe cereri, astfel dacă clientul are nevoie de o resursă cu reprezentare ambiguă, serverul trimite înapoi toate reprezentările, de unde clientul alege și trimite la server opțiunea dorită și serverul trimite reprezentarea specificată.

1. Ce este un ETag în HTTP și cum funcționează?

ETag este un antet din respunsul serverului care definește versiunea unei resurse. Când clientul mai are nevoie de această resursă adaugă antetul “If-None-Match” în cerere, dacă serverul deține o versiune mai nouă a resursei, o transmite și specifică versiunea în ETab.

1. Diferența dintre protocoalele fără stare și cele cu stare. Cărui tip îi aparține HTTP?

Protocoalele cu stare mențin o conexiune între server și client până când transferul de date s-a finisat, pe când protocoalele fără stare nu mențin nici o conexiune între client și server. HTTP este un protocol fără stare, dar poate simula starea prin cookies, sau sesiune.

1. Avantajele cheie ale HTTP/2 în comparație cu HTTP/1.1

* Utilizarea multiplexării
* Introducerea funcționalității server-push
* Arhivarea anteților utilizând HPACK
* Protocol binar

1. Ce este un tip MIME, din ce constă și pentru ce se folosește?

MIME (Multipurpose Internet Mail Extensions) sau simplu media type este un standart care indică tipul și formatul unui document sau fișier. Un MIME constă din două șiruri de caractere (tip, subtip) concatenate cu un slash între ele. Se folosește pentru ca clientul să proceseze corect contentul returnat de server.

1. Care este diferența dintre GET și POST?

GET este o metodă folosită pentru cereri de extragere a resurselor de pe server, POST pentru transmiterea datelor pe server.

1. Care este diferența dintre PUT și POST?

PUT se folosește pentru update, POST pentru a crea noi entități. PUT poate crea resurse dacă se cunoaște calea exactă.

1. Care sunt metodele idempotente în HTTP?

PUT, DELETE, GET, OPTIONS, HEAD.

1. Care sunt metodele sigure și nesigure în HTTP?

Sigure: OPTIONS, GET, HEAD.

Nesigure: PUT, POST, DELETE, PATCH

1. Pentru ce este nevoie de cURL?

Pentru a automatiza transferul de date dintre client și server.

1. Pentru ce este nevoie de HTTP Proxy?

Pentru securitate, anonimitate, controlul accesului la internet, viteza sporita a internetului.

1. Diferența dintre autentificare și autorizare

Autentificarea raspunde la întrebarea “Cine ești?”, autorizarea – “Ce drepturi ai?”.

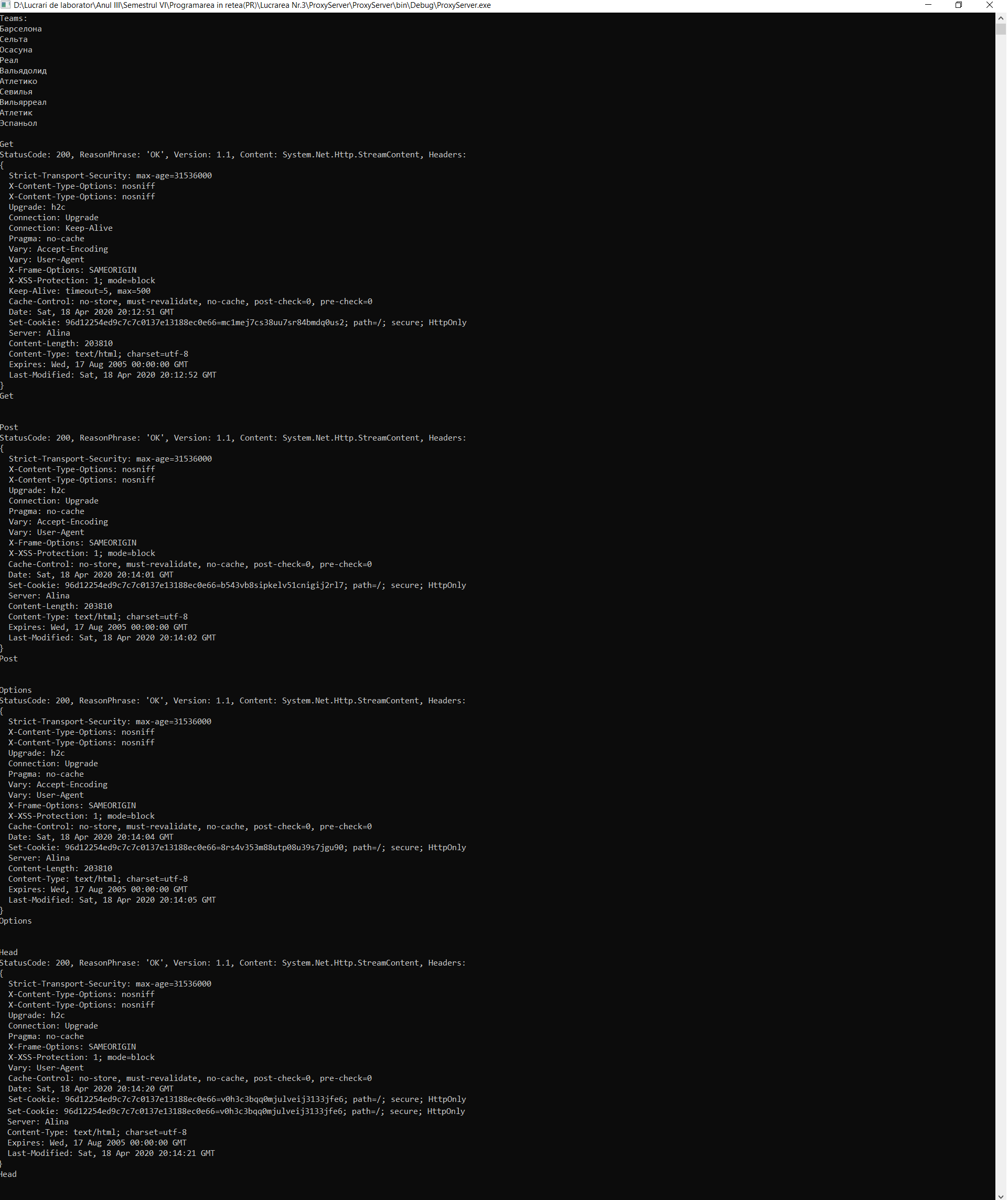
1. Care sunt metodele de autentificare HTTP?

Prin Cookies, Tokens, Semnături, Parolă pentru unică folosință, Basic HTTP (criptarea username-ului și parolei)

1. HTTP cookies – pentru ce se folosește?

Pentru păstrarea pe calculatorul unui user a configurărilor, setărilor, proprietăților și în general a datelor în formă criptată trimise de un server. La restabilirea conexiunii cu serverul respectiv, cookisurile sunt trimite înapoi la server.

Rezultate obținute:



Concluzie:

In urma efectuării acestei lucrări de laborator s-au acumulat competențe de lucru cu protocolul HTTP. De asemenea au fost căpătate cunoștințe în ceea ce privește conectarea la proxy server. Tot prin proxy au fost evectuate și request-urile la pagina web aleasă. Au fost folosite următoarele metode ale protocolului HTTP ca GET, POST, OPTIONS, HEAD.

Cu ajutorul expresiilor regulate, au fost extrase echipele din clasamentul campionatului intern a Spaniei. În final au fost analizate rezultatele și formulate concluzii.

**Github Link:** [**https://github.com/Verhovetchi/PR**](https://github.com/Verhovetchi/PR)

**Video Link:** [**https://drive.google.com/file/d/1p5FzjghnZrASfAjq0JdzI80uKoc35ea/view**](https://drive.google.com/file/d/1p5FzjghnZrASfAjq0JdzI80uKoc35ea/view)