

UNIVERSITATEA DE STAT DIN MOLDOVA
FACULTATEA MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL INFORMATICĂ

DARE DE SEAMĂ

Lucrare de laborator №1 la tema :

”Bazele HTTP”

Disciplina ***”Framework-uri Pentru Dezvoltarea WEB”***

Coordonator: Nartea Nichita

Autorul: Spînu Andrei IA2201

Chișinău, 2024

Cuprins:

1.	<u>Partea teoretică</u>	<u>3</u>
	• <u>Formularea problemei</u>	<u>3</u>
	• <u>Descrierea scopului și a principalelor etape ale lucrării</u>	<u>4</u>
2.	<u>Partea practică</u>	<u>4</u>
	• <u>Descrierea scurtă a aspectelor de implementare</u>	<u>4</u>
3.	<u>Concluzie și link către depozitul Git</u>	<u>5</u>
4.	<u>Răspunsuri la întrebările de control</u>	<u>5</u>
5.	<u>Lista surselor utilizate</u>	<u>6</u>

Formularea problemei:

Sarcina nr. 1. Analiza cererilor HTTP

1. Accesați site-ul <http://sandbox.usm.md/login>.
2. Deschideți fila Network în instrumentele pentru dezvoltatori ale browserului.
3. Introduceți date incorecte pentru autentificare (de exemplu, username: student, password: studentpass).
4. Analizați cererile care au fost trimise către server.
5. Răspundeți la următoarele întrebări:
 - Ce metodă HTTP a fost utilizată pentru a trimite cererea?
 - Ce anteturi au fost trimise în cerere?
 - Ce parametri au fost trimiși în cerere?
 - Ce cod de stare a fost returnat de server?
 - Ce anteturi au fost trimise în răspuns?
6. Repetați pașii 3-5, introducând date corecte pentru autentificare (username: admin, password: password).

Sarcina nr. 2. Crearea cererilor HTTP

1. Scrieți o cerere de tip GET către server la adresa <http://sandbox.com>, indicând în antetul User-Agent numele și prenumele dvs.
2. Scrieți o cerere de tip POST către server la adresa <http://sandbox.com/cars>, indicând în corpul cererii următorii parametri:
 - make: Toyota
 - model: Corolla
 - year: 2020
3. Scrieți o cerere de tip PUT către server la adresa <http://sandbox.com/cars/1>, indicând în antetul User-Agent numele și prenumele dvs., în antetul Content-Type valoarea application/json, iar în corpul cererii următorii parametri: json { "make": "Toyota", "model": "Corolla", "year": 2021 }
4. Scrieți unul dintre posibilele răspunsuri ale serverului la cererea anterioară. http POST /cars HTTP/1.1 Host: sandbox.com Content-Type: application/json User-Agent: John Doe model=Corolla&make=Toyota&year=2020 Presupuneți situațiile în care serverul poate returna codurile de stare HTTP 200, 201, 400, 401, 403, 404, 500.
5. Scrieți o cerere de tip DELETE la alegerea dvs. și să explicați de ce, în acest caz, este potrivit să utilizați metoda DELETE.

Sarcina nr. 3. Sarcina suplimentară. HTTP_Quest

TIP

Parcurgeți quest-ul trimițând cereri către server.

1. Trimiteți o cerere de tip POST către server la adresa `http://sandbox.usm.md/quest`, indicând în antetul User-Agent numele și prenumele dvs. (De exemplu, User-Agent: John Doe).
`http POST /quest HTTP/1.1 Host: sandbox.usm.md User-Agent: John Doe`
curl: `bash curl -X POST http://sandbox.usm.md/quest -H "User-Agent: John Doe"`
2. Urmați instrucțiunile de pe server, îndeplinindu-le în ordine.
3. La finalul quest-ului, vi se va afișa un cuvânt secret, pe care va trebui să-l includeți în raport.

Descrierea scopului și a principalelor etape ale lucrării

Scopul principal al acestei lucrări este de a înțelege bazele protocolului HTTP și de a efectua operațiuni fundamentale de trimitere și analiză a cererilor către un server web. Lucrarea se concentrează pe realizarea cererilor de tip GET, POST, PUT și DELETE, analizând răspunsurile și codurile de stare returnate de server, precum și explorarea fluxului de interacțiune client-server. În plus, sarcina suplimentară presupune parcurgerea unui quest HTTP folosind instrumente dedicate, precum curl sau Postman, pentru a îmbunătăți abilitățile practice în manipularea cererilor HTTP.

Partea practică

În cadrul acestei lucrări de laborator, am revenit asupra elementelor de bază ale protocolului HTTP, ceva ce am mai întâlnit anterior, dar a fost util să le revizitez în mod practic. Prima sarcină mi-a amintit de analiza cererilor HTTP direct în browser, folosind instrumentele de dezvoltator. Am accesat pagina de login de pe `http://sandbox.usm.md/login` și am simulat atât o autentificare eșuată, cât și una reușită. Așa cum mă așteptam, cererea de autentificare a fost de tip POST, iar codurile de stare returnate au fost relevante pentru fiecare situație — 401 Unauthorized pentru autentificarea eșuată și 200 OK pentru cea reușită. Acest proces a fost familiar, însă mi-a oferit o oportunitate să verific încă o dată detaliile traficului între client și server.

Partea practică, în care am trimis diverse cereri GET, POST, PUT și DELETE folosind curl și Postman, a fost o reîmprospătare utilă. Crearea unei cereri POST pentru a adăuga un automobil, urmată de o cerere PUT pentru a-l actualiza, mi-a reamintit cât de importantă este structura corectă a datelor și anteturilor într-un mediu REST. Exemplele acestea, deși simple, mi-au consolidat cunoștințele despre cum să manipulez diferitele metode HTTP și să gestionez datele trimise către server.

Sarcina finală, un quest în care am interacționat cu serverul printr-o serie de cereri, mi-a oferit o modalitate interesantă de a repeta fluxul de lucru client-server. Deși a fost un exercițiu familiar, mi-a plăcut să folosesc din nou Postman pentru a trimite cereri și a urmări pașii quest-ului. La

final, am extras cuvântul secret, un detaliu care a pus accentul pe corectitudinea implementării fiecărei cereri.

Concluzie și link către depozitul Git

Această lucrare de laborator a fost o oportunitate utilă de a revedea și consolida noțiunile esențiale ale protocolului HTTP. Deși am fost deja familiarizat cu metodele și anteturile HTTP, exercițiile practice mi-au permis să îmi reîmprospătez cunoștințele și să îmbunătățesc înțelegerea modului în care funcționează comunicarea client-server. Prin utilizarea instrumentelor *curl* și *Postman*, am avut ocazia să implementez și să testez cereri într-un mediu controlat, verificând răspunsurile și înțelegând contextul fiecărei metode HTTP.

În concluzie, laboratorul a servit drept o reamintire eficientă și practică a bazelor HTTP, contribuind la aprofundarea competențelor necesare în dezvoltarea web.

Depozitul cu implementarea acestei lucrări de laborator poate fi accesat la următorul link:

https://github.com/andy1kspn/FPDWlabs/tree/Laborator_01

Răspunsuri la întrebările de control

1. Ce este protocolul HTTP?

HTTP este un protocol de transfer de date între client și server pe web.

2. Pentru ce folosim antetul User-Agent?

User-Agent identifică tipul de browser sau aplicație folosită de client în cererile către server.

3. Care este diferența dintre metodele PUT și PATCH?

PUT actualizează o resursă complet, în timp ce PATCH modifică doar anumite părți ale resursei.

Lista surselor utilizate

1. https://moodle.usm.md/pluginfile.php/727758/mod_resource/content/1/Client-Server.%20Ce%20este%20HTTP_.pdf
2. https://github.com/MSU-Courses/frameworks-for-web-development/blob/main/ro/01_http_basics/01_04_curl.md?plain=1