

Лабораторна робота №7

Протокол передачі гіпертексту HTTP

Мета: Вивчити принципи роботи та використання протоколу HTTP, формування HTTP запитів та структуру HTTP відповіді.

Теоретичні відомості

HTTP (HyperText Transfer Protocol – протокол передавання гіпертексту) було розроблено як основу World Wide Web. Його робота ґрунтується на принципі взаємодії клієнт-сервер: програма-клієнт установлює TCP з'єднання із сервером (стандартний номер для порту http-серверу 80) та надсилає йому запит HTTP. Сервер оброблює цей запит та надсилає клієнту HTTP відповідь. Документи, що регламентують використання протоколу HTTP, можна знайти за наступним посиланням: <http://www.w3.org/Protocols/>.

Структура HTTP-запиту

HTTP-запит складається із заголовку запиту та його тіла, які розділяються пустим рядком (Рис.1). Тіло запиту може бути відсутнім.



Рис.1. Загальна структура запиту HTTP

Рядок запиту (Request Line) вказує :

- метод передавання;
- URL-адресу, до якої необхідно звернутись;
- версію протоколу HTTP.

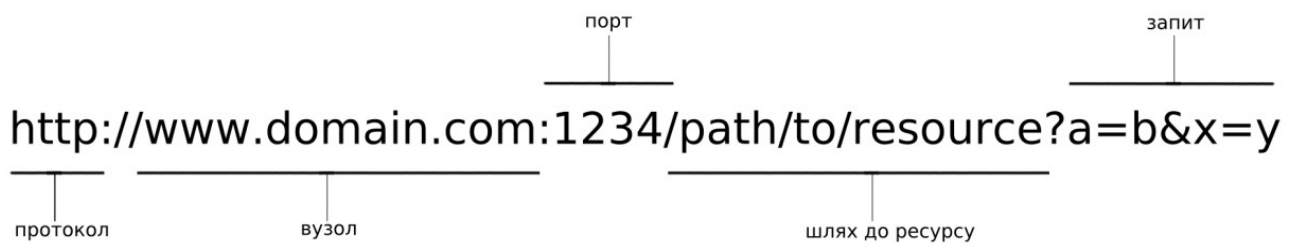
Методами передавання можуть бути:

- GET, якщо необхідно запитати дані разом із HTTP-заголовками;
- HEAD, якщо необхідні HTTP-заголовки без даних;
- POST, якщо необхідно відправити дані на сервер.

Для запиту за методом POST обов'язковими параметрами є тип тіла запиту (Content-Type) та довжина тіла запиту у байтах (Content-Length).

URL (англ. Uniform Resource Locator) це єдиний вказівник на ресурс.

Структура URL-адреси наступна:



де протокол – тип протоколу, за яким робиться запит; вузол – ім'я серверу в системі DNS; порт – порт, який прослуховує серверна програма; шлях до ресурсу – шлях до певного ресурсу або обробника запитів на стороні серверу; запит – параметри або вирази запиту.

Для HTTP-серверів в якості протоколів в URL можна вказувати або `http` для звичайних з'єднань, або `https` – для більш безпечного обміну даними. За замовчуванням HTTP-сервери використовують порт за номером 80.

Заголовки (Message Headers) описують тіло повідомлення, передають різноманітні параметри, інші відомості та інформацію. Параметри передаються у наступному форматі:

Ім'я_параметру: значення_параметру

Наступні параметри частіше використовуються у HTTP-запиті:

Connection (з'єднання) – може приймати значення `Keep-Alive` або `close`.

`Keep-Alive` ("залишити живим") позначає, що після видачі даного документу з'єднання із сервером не розривається, і можна відправляти інші запити. Більшість браузерів працюють саме у режимі `Keep-Alive`, тому що він дозволяє за одне з'єднання із сервером отримати HTML-сторінку та, наприклад, рисунки до неї. Якщо режим `Keep-Alive` встановлено, то він зберігається до першої помилки або до явного вказування на черговому запиті `Connection: close`.

`Close` ("закрити") – з'єднання закривається після відповіді на даний запит.

User-Agent – його значенням є "кодове позначення" браузера, наприклад:

`Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0`

У цьому параметрі також часто вказується операційна система клієнта.

Accept – список типів, що підтримуються браузером, які наводяться у порядку їх уподобання цим браузером. Наприклад, для `Firefox/72.0` цей параметр може мати наступні значення:

`Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`

Referer – URL, з якого перейшли на даний ресурс.

Host – ім'я вузла, на якому знаходиться ресурс у запиті. Це корисно для випадку, коли на сервері є декілька віртуальних серверів з однією IP адресою. У цьому випадку ім'я віртуального серверу визначається за допомогою цього поля.

Accept-Language – мова, що підтримується. Цей параметр має значення для серверу, який може видавати один і той самий документ у різних мовних версіях. Наприклад: `Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3`

Приклад простого HTTP-запиту:

`GET / HTTP/1.1` – перший рядок заголовку запиту;

`Host: www.znu.edu.ua` – другий рядок заголовку запиту

- пустий рядок

Структура HTTP-відповіді

Формат HTTP-відповіді є схожим на формат HTTP-запиту: він також має заголовок та тіло, які розмежовуються пустим рядком. Заголовок також складається з основного рядка та рядків з параметрами. Основний рядок запиту складається із 3-х полів, що поділяються пробілами: версія протоколу, код помилки (кодове позначення стану виконання запиту) та словесний опис помилки. Коди 2xx позначають успішне виконання запиту, 3xx — відповідає тій чи іншій формі перенаправлення, 4xx - позначає помилку з вини клієнта, 5xx — помилка на сервері або у сценарії.

Нижче наведено параметри HTTP-відповіді, що використовуються найчастіше.

Date – дата обробки запиту.

Server – назва веб-серверу.

ETag є міткою ресурсу для статичних документів. Зазвичай вона подібна контрольній сумі або підпису файлу.

X-Powered-By – для динамічних документів вказує на додаток, який було використано для формування документу. Наприклад: X-Powered-By: PHP/5.0.3

Connection – подібний до відповідного параметру запиту.

Content-Type (тип вмісту) – має позначення типу вмісту відповіді. У залежності від значення Content-Type браузер сприймає відповідь як HTML-сторінку, картинку gif або jpeg, як файл, який необхідно зберегти на диску, або щось інше, що потребує певних визначених дій від нього. Широковживаними типами вмісту є: text/html - текст у форматі HTML (веб-сторінка); text/plain - простий текст; image/jpeg - картинка у форматі JPEG; image/gif — картинка у форматі GIF; application/octet-stream - потік даних для запису на диск. Часто в параметрі Content-Type через крапку з комою також вказується тип кодування документу.

Наприклад:

```
Content-Type: text/html; charset=windows-1251
```

Content-Length - довжина вмісту відповіді у байтах.

Last-Modified – дата останньої зміни документу.

Завдання до лабораторної роботи

1. Запустіть програму Wireshark. Налаштуйте фільтр програми для захоплення пакетів з даними HTTP. Запустіть будь-який браузер, що встановлено на комп'ютері. Переведіть Wireshark у режим захоплення пакетів з HTTP (опція port 80). Укажіть у рядку адреси браузеру <http://school-izm9.in.ua/> (або іншим за вказівкою викладача) та натисніть Enter. У програмі Wireshark повинні з'явитись декілька захоплених пакетів. За заголовками HTTP визначте які з них є HTTP-запитами, а які HTTP-відповідями. Наведіть у звіті HTTP-заголовки одного сеансу «запиту-відповіді».
2. За допомогою утиліти telnet встановіть ТСП з'єднання за 80-м портом з вузлом <http://school-izm9.in.ua/> (або іншим за вказівкою викладача). Для цього запустіть утиліту telnet ("Пуск" -> "Виконати" -> "cmd" -> "telnet"), встановіть режим "локальне відлуння", для того щоб бачити набраний текст.

Для з'єднання з вузлом за 80-м портом введіть команду:

open school-izm9.in.ua 80.

Після цього можна вводити HTTP-запит, приклад якого показано в теоретичній частині. Наведіть скрин-шоти виконання завдань у звіті.

3. Використайте дані, що отримано за допомогою програми Wireshark, сформулюйте HTTP-запит до серверу <http://school-izm9.in.ua/> (можна використовувати декілька перших рядків заголовку). Після отримання HTTP-відповіді її можна скопіювати у буфер. Для цього виділіть текст за допомогою лівої клавіші миші та натисніть праву кнопку миші в межах вікна. У результаті виділений текст скопіюється у буфер і його можна буде вставити в будь-який редактор за допомогою комбінації клавіш **Ctrl+V**. Після встановлення з'єднання відправте HTTP-запит на отримання HTTP-заголовків для головної сторінки сайту без отримання даних. Необхідно, щоб код відповіді був або 2** (три цифри, де перша 2), або 3**, або 5**. Якщо код починається з цифри 4, то було допущено помилку. У першому рядку запиту слід вказувати відносну адресу головної сторінки (тобто /), у заголовку запиту обов'язково потрібно передати параметр Host. У звіті наведіть **Screenshot** копії HTTP-запитів і відповідей серверу.
4. Повторіть запити для двох будь-яких web серверів. Наведіть аналіз отриманих HTTP-відповідей та заповніть таблицю:

	Сервер 1	Сервер 2
URL адреса сервера		
Код відповіді		
Web-сервер (назва, версія)		
Операційна система сервера		
Статична чи динамічна сторінка Для статичної сторінки вказати Etag, для динамічної – додаток, який створював HTML-сторінку у відповідь		
Кодування документу (якщо є)		
Чи закриває з'єднання сервер одразу, чи очікує нових відповідей?		

Контрольні запитання

- 1 Які порти прослуховує Web-сервер?
- 2 Яка структура HTTP-запиту до серверу?
- 3 Задля якої мети у HTTP-запиті використовується параметр Connection?
- 4 Яку інформацію несе у HTTP-запиті параметр Host ?
- 5 Який вигляд має простий HTTP-запит?
- 6 Яка структура HTTP-відповіді від серверу?
- 7 Які коди можуть бути у HTTP-відповіді та що вони означають?