Інформаційно-комунікаційні систем. Частина ІІ

Тема 3

Програмне забезпечення комп’ ютерних мереж

План лекції

1. Протокол динамічної конфігурації вузла, DHCP
2. Система доменних імен, DNS
3. Мережевий протокол для доступу до текстового інтерфейсу, Telnet
4. Протокол передачі файлів, FTP
5. Протокол захищеного віддаленого керування SSH
6. Простий протокол пересилання пошти, SMTP
7. Поштовий офісний протокол, POP3
8. Протокол доступу до інтернет-повідомлень, IMAP
9. **Протокол передачі гіпертексту, HTTP**
10. Простий протокол керування мережею, SNMP

9. Протокол передачі гіпертексту, HTTP

- **Характеристики**
  - **Протокол передачі гіпертесту (HyperText Transfer Protocol, HTTP) –**
    - **потокол який забезпечує діалог між браузерами та веб-серверами:**
      - **Браузер – програмний засіб, який дозволяє взаємодіяти з даними (тест, зображення тощо) на гіпертекстовій веб-сторінці**
      - **Веб-сервер (HTTP-сервер) – це сервер, що приймає HTTP-запити від клієнта та повертає HTTP-відповіді.**
    - **RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0**
    - **RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1**
  - **В якості транспортного протоколу використовується – TCP.**
  - **Протокол HTTP за стандартом використовує:**
    - **80-й порт TCP для обміну даними;**
    - **443-й порт TCP для захищеного обміну даними (HTTP over SSL)**
    - **Може бути використаний будь-який інший порт.**

Формат стрічки підключення до веб-серверу

- **Уніфікований ідентифікатор ресурсів**
  - **RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax;**
  - **структура уніфікованого ідентифікатора ресурсів:**

<схема>://<логін>: <пароль>@<вузол>: <порт>/<URL-шлях>?<параметри>#<якір>

  - **<схема> - протокол прикладного рівня, наприклад: file, ftp, http, https тощо;**
  - **<логін> - ім'я користувача, яке використовується при доступі;**
  - **<пароль> - пароль конкретного користувача;**
  - **<вузол> - доменне ім'я або IP-адреса вузла;**
  - **<порт> - TCP/UDP-порт вузла, який використовується при підключенні;**
  - **<URL-шлях> - статична або динамічна адреса до файлу;**
  - **<параметри> - стрічка запиту, що передається на сервер;**
  - **<якір> - ідентифікатор якоря, який посилається на певну частину документу.**

- **Приклади, доступ до ресурсу через протокол:**
  - **HTTP - http://example.com/pub/files/school.html**
  - **HTTP - http://test:test@example.com:8080/pub/files/school.html**
  - **HTTPS - https://example.com/pub/files/school.html**
  - **FTP - ftp://test:test@ftp.examle.com/**

Типи з’єднань та HTTP-методи

- **Протокол HTTP підтримує постійні та непостійні з’єдннння:**
  - **При непостійному з'єднанні протокол TCP одержує тільки один об'єкт**
  - **При постійному з'єднанні - всі об'єкти в межах одного TCP-з’єднання**
    - **Даний метод для HTTP 1.1 використовується по замовчуванню**
    - **Клієнти і сервери з підтримкою HTTP 1.1 можна налаштувати і для непостійного з'єднання.**
- **Метод HTTP - послідовність з будь-яких символів, крім керуючих, яка вказує на основну операцію над ресурсом.**
- **Зазвичай метод являє собою коротке англійське слово, записане заголовними буквами.**
  - **Наприклад, OPTIONS, GET, HEAD, PUT, POST, PATCH, DELETE, TRACE, LINK, UNLINK**
  - **Назви методу чутливі до регістру.**

HTTP-методи

- **HTTP 1.0**
  - **GET – згідно стандарту HTTP, запити типу GET вважаються ідемпотентним - багаторазове повторення одного і того ж запиту GET повинно приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами).**
    - **Це дозволяє кешувати відповіді на запити GET.**
    - **Крім звичайного методу GET, розрізняють ще умовний GET і частковий GET.**
      - **Умовні запити GET містять заголовки If-Modified-Since, If-Match, If-Range і подібні.**
      - **Часткові GET містяться в запиті Range. Порядок виконання подібних запитів визначений стандартами окремо.**
  - **POST - застосовується для передачі даних користувача заданому ресурсу.**
    - **Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються серверу методом POST і він поміщає їх на сторінку.**
      - **При цьому передані дані (у прикладі з блогами - текст коментаря) включаються в тіло запиту.**
      - **Аналогічно за допомогою методу POST зазвичай завантажуються файли.**
      - **На відміну від методу GET, метод POST не рахується ідемпотентним, тобто багаторазове повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожної відправки коментаря з'являтиметься одна копія цього коментаря).**
      - **Якщо результатат виконання команди - 200 (Ok) і 204 (No Content), то в тіло відповіді слід включити повідомлення про підсумок виконання запиту.**
      - **Якщо був створений ресурс, то серверу слід повернути відповідь - 201 (Created) із зазначенням URI нового ресурсу в заголовку Location.**
      - **Результат виконання методу POST не кешується.**

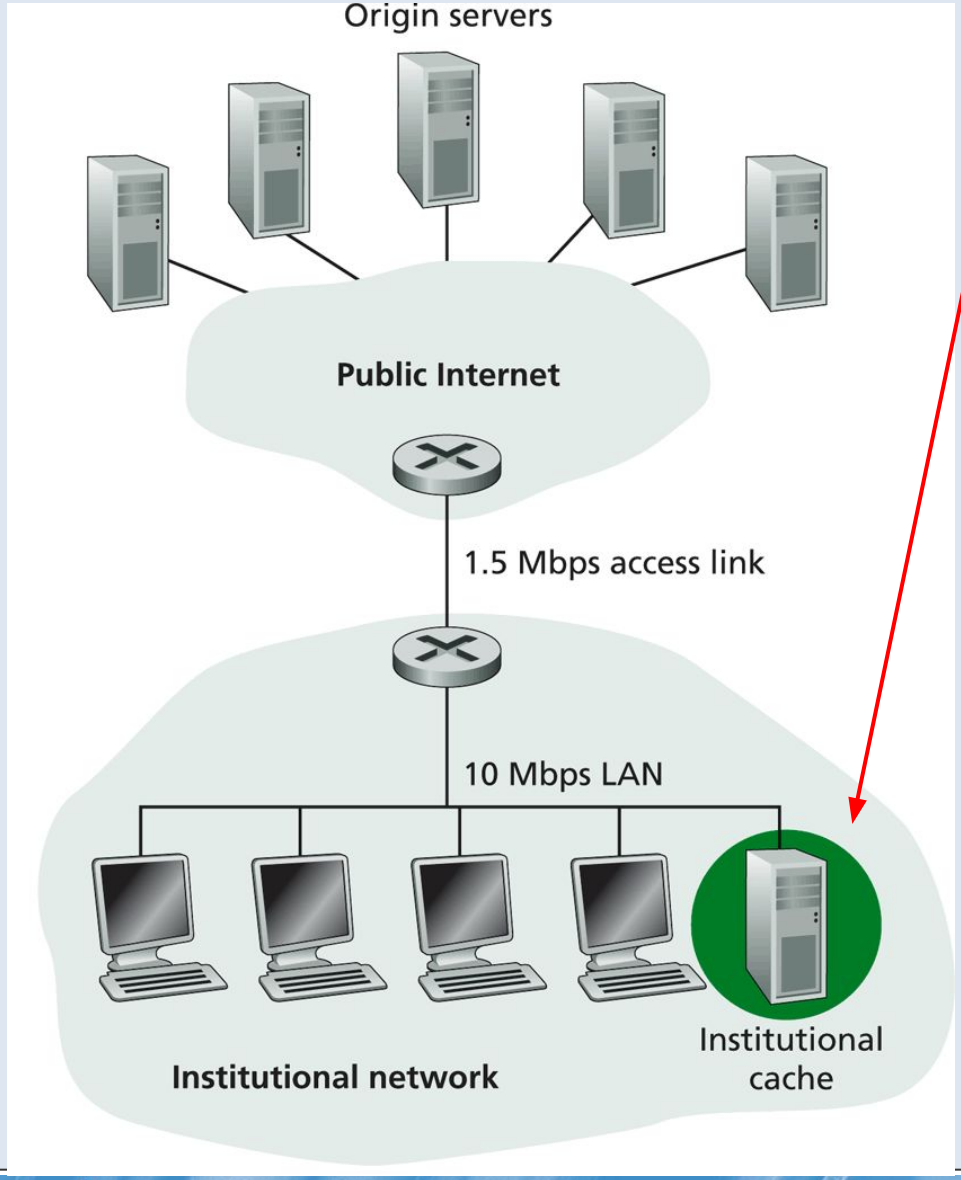
HTTP-методи

- **HTTP 1.0**
  - **HEAD – аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло.**
    - **Запит HEAD зазвичай застосовується для витягання метаданих, перевірки наявності ресурсу (URL) і щоб дізнатися, чи не змінився він з моменту останнього звернення.**
    - **Заголовки відповіді можуть кешуватися. При розбіжності метаданих ресурсу з відповідною інформацією в кеші копія ресурсу позначається як застаріла.**
- **HTTP 1.1**
  - **GET, POST, HEAD**
  - **PUT - застосовується для завантаження вмісту запиту - на вказаний в запиті URI.**
    - **Якщо по заданому URI не існувало ресурсу, то сервер створює його і повертає статус 201 (Created).**
    - **Якщо був змінений ресурс, то сервер повертає 200 (Ok) або 204 (No Content).**
    - **Сервер не повинен ігнорувати некоректні заголовки Content-*, які передані клієнтом разом із повідомленням. Якщо якийсь з цих заголовків не може бути розпізнаний або не допустимий при поточних умовах, то пвертається код помилки 501 (Not Implemented).**
    - **Фундаментальна відмінність методів POST і PUT полягає в розумінні призначень URI ресурсів. Метод POST припускає, що за вказаною URI виконується обробка переданого клієнтом вмісту. Використовуючи PUT, клієнт передбачає, що завантажуваний вміст розташований за даним URI ресурсу.**
    - **Повідомлення відповідей сервера на метод PUT не кешується.**
  - **DELETE - Видалення об'єкта на сервері за вказаною URL**

HTTP-методи

- **HTTP 1.1**
  - **PATCH – аналогічний PUT, але застосовується тільки до фрагменту ресурсу.**
  - **OPTIONS - використовується для визначення можливостей веб-серверу або параметрів з'єднання для конкретного ресурсу.**
    - **Передбачається, що запит клієнта може містити тіло повідомлення, в якому вказаний перелік відомостей, що його цікавить.**
    - **Результат виконання цього методу не кешується.**
  - **TRACE – повертає отриманий запит так, що клієнт може побачити, які проміжні сервера додають або змінюють дані в запиті.**
  - **LINK – встановлює зв'язок зазначеного ресурсу з іншими.**
  - **UNLINK – прибирає зв'язок зазначеного ресурсу з іншими.**

Використання проксі-серверів



- **Проксі-сервер - це транзитний сервер, що перенаправляє HTTP-трафік.**
- **Проксі-сервер реалізує дві основні функції**
  - **посередника між клієнтами та серверам;**
    - **контроль процесу обміну даними між клієнтами та серверами;**
    - **застосовується як міжмережевий екран і засіб управління HTTP-трафіком;**
    - **використовують для анонімізації запитів.**
  - **кешування даних:**
    - **виконує кешування запитів/відповідей клієнтів з метою підвищення відклику;**
    - **зменшує завантаженість каналу.**

Коди HTTP-відповідей

- **1xx – Інформаційні**
  - **100 Continue:**
    - **Сервер задоволений початковими відомостями про запит.**
    - **Клієнт може продовжувати пересилати заголовки.**
  - **101 Switching Protocols:**
    - **Сервер пропонує перейти на більш відповідний для зазначеного ресурсу протокол.**
    - **Список пропонованих протоколів сервер обов'язково вказує в поле заголовка Update. Якщо клієнта це зацікавить, то він посилає новий запит із зазначенням іншого протоколу.**
  - **102 Processing**
    - **Запит прийнятий, але на його обробку знадобиться тривалий час. Використовується сервером, щоб клієнт не розірвав з'єднання через перевищення часу очікування. Клієнт при отриманні такої відповіді повинен скинути таймер і чекати наступної команди в звичайному режимі.**
- **2xx – Успішна доставка (Successful)**
  - **Повідомлення цього класу інформують про випадки успішного приймання та обробки запиту клієнта.**
  - **200 OK:**
    - **Успішний запит. Якщо клієнтом були запитані будь-які дані, то вони знаходяться в заголовку і/або тілі повідомлення.**

Коди HTTP-відповідей

- **2xx – Успішна доставка (Successful)**
  - **201 Created (Створено)**
    - **В результаті успішного виконання запиту було створено новий ресурс.**
  - **202 Accepted (Прийнято)**
    - **Запит був прийнятий на обробку, але обробка не завершена.**
  - **203 Non-Authoritative Information (Не надійна інформація)**
    - **Аналогічно відповіді 200, але в цьому випадку передана інформація була взята не з первинного джерела (резервної копії, іншого сервера тощо). І тому може бути неактуальною.**
  - **204 No Content (Ніякого вмісту)**
    - **Сервер успішно обробив запит, але у відповіді були передані тільки заголовки без тіла повідомлення. Клієнт не повинен оновлювати вміст документа, але може застосувати до нього отримані метадані.**
  - **205 Reset Content (Скидання вмісту)**
    - **Сервер зобов'язує клієнта запитати введені користувачем дані. Тіла повідомлення сервер при цьому не передає і документ оновлювати не обов'язково.**
  - **206 Partial Content (Частковий вміст)**
    - **Сервер вдало виконав запит клієнта, але передав тільки частину документу.**

Коди HTTP-відповідей

- **3xx – Переадресація (Redirection)**
  - **Цей клас статусних кодів вказує, що для виконання запиту, потрібні подальші дії агента користувача.**
  - **300 Multiple Choices (Множинний вибір)**
    - **За вказаною URI існує кілька варіантів надання ресурсу по типу MIME.**
    - **Сервер передає з повідомленням список альтернативних варіантів, надаючи можливість клієнтові або користувачеві зробити вибір.**
  - **301 Moved Permanently (Постійно переміщений)**
    - **Запитаний документ був остаточно перенесено на новий URI, вказаний в полі Location заголовку.**
    - **При запитах не шляхом HEAD сервер повинен передати в тілі повідомлення гіпертекстове пояснення.**
    - **При використанні всіх методів, крім GET і POST, попередньо слід повідомити користувача про зміну посилання.**
  - **302 Moved Temporarily (Тимчасово переміщений)**
    - **Запитаний документ був тимчасово перенесений на інший URI, вказаний в заголовку в поле Location.**
    - **При всіх методах крім HEAD сервер повинен передати в тілі гіпертекстове пояснення.**
    - **При використанні всіх відмінних від GET і POST методів попередньо слід повідомити користувача про зміну URI.**

Коди HTTP-відповідей

- **3xx – Переадресація (Redirection)**
  - **303 See Other (Дивись інші)**
    - **Документ по даному URI потрібно запросити за адресою в полі Location заголовка з використанням методу GET не дивлячись навіть на те, що перший запит виконано методом POST.**
    - **Якщо використовується не метод HEAD, то серверу слід включити в тіло повідомлення короткий гіпертекстове опис.**
  - **304 Not Modified (Не модифіковано)**
    - **Сервер повертає такий код, якщо клієнт запросив документ методом GET, в заголовку використовував поле Date і документ не змінився з зазначеного моменту. При цьому повідомлення сервера не повинно містити тіла.**
  - **305 Use Proxy (Використовуйте проксі)**
    - **Запит до запитуваного ресурсі повинен здійснюватися через проксі-сервер, URI якого вказаний в полі Location заголовка.**
    - **Даний код відповіді можуть використовувати тільки рідні HTTP-сервера (Не проксі).**
  - **306 Reserved (Зарезервовано)**
    - **На даний момент зарезервований код.**

Коди HTTP-відповідей

- **3xx – Переадресація (Redirection)**
  - **307 Temporary Redirect (Тимчасове перенаправлення)**
    - **Запитуваний ресурс короткий час доступний тільки по іншому URI (вказується в полі Location заголовка). Якщо був посланий не метод HEAD, то серверу слід включити в тіло повідомлення короткий гіпертекстове опис.**
    - **При використанні всіх методів крім GET і POST попередньо слід повідомити користувача про тимчасову зміну посилання.**
- **4xx – Помилка клієнта (Client Error)**
  - **Клас кодів 4xx призначений для вказівки помилок з боку клієнта. При використанні всіх методів, крім HEAD, сервер повинен повернути в тілі повідомлення гіпертекстове пояснення для користувача.**
  - **400 Bad Request (Поганий запит)**
    - **Запит не зрозумілий сервером через наявність синтаксичної помилки. Клієнту слід повторно звернутися до ресурсу зі зміненим запитом.**
  - **401 Unauthorized (Неавторизований доступ)**
    - **Запит вимагає ідентифікації користувача. Клієнт повинен запитати ім'я і пароль у користувача і передати їх у записі WWW-Authenticate заголовка в наступному запиті. У разі введення помилкових даних сервер знову поверне цей же статус.**
  - **402 Payment Required (Необхідна оплата (зарезервовано)**
    - **Передбачається використовувати в майбутньому. На даний момент не використовується.**

Коди HTTP-відповідей

- 4xx – Помилка клієнта (Client Error)
  - 403 Forbidden (Заборонено доступ)
    - Сервер відмовляється виконувати запит через обмеження в доступі.
  - 404 Not Found (Не знайдено)
    - Сервер не знайшов відповідного ресурсу за вказаною URI.
  - 405 Method Not Allowed (Метод не підтримується)
    - Зазначений клієнтом метод (GET, POST, HEAD) не можна застосувати до ресурсу.
  - 406 Not Acceptable (Не прийнятно)
    - Запитаний URI не може задовольнити переданим в заголовку характеристикам.
  - 407 Proxy Authentication Required (Необхідна авторизація на проксі)
    - Відповідь аналогічна коду 401 за винятком того, що автентифікація проводиться для проксі-сервера.
  - 408 Request Timeout (Час очікування запиту сплинув)
    - Час очікування сервером передачі запиту від клієнта минув.
  - 409 Conflict (Конфлікт)
    - Запит не можна виконати через конфлікт доступу до ресурсу.
    - Наприклад, коли два клієнта намагаються змінити ресурс за допомогою методу PUT.

Коди HTTP-відповідей

- **4xx – Переадресація (Помилка клієнта)**
  - **410 Gone (Вилучений)**
  - **411 Length Required (Необхідно вказати довжину)**
  - **412 Precondition Failed (Умова «хибна»)**
  - **413 Request Entity Too Large (Запрошувані дані занадто великі)**
  - **414 Request-URI Too Long (Запитуваний URI занадто довгий)**
  - **415 Unsupported Media Type (Тип даних не підтримується сервером)**
  - **416 Requested Range Not Satisfiable (Не можна досягти запитуваного діпазону)**
  - **417 Expectation Failed (Помилка в полі Expectation)**
  - **422 Unprocessable Entity (Необроблюваний екзмепляр)**
  - **423 Locked (Заблоковано)**
  - **424 Failed Dependency (Невиконувана залежність)**
  - **426 Upgrade Required (Необхідно оновлення)**

Рекомендовані ресурси для самостійного ознайомлення з кодами відповідей:

https://www.ietf.org/rfc/rfc2616.txt

http://www.lib.ru/WEBMASTER/rfc2068/

http://www.webdav.org/specs/rfc4918.html

Коди HTTP-відповідей

- **5xx – Помилка серверу (Server Error)**
  - **Для всіх ситуацій, крім використання методу HEAD, сервер повинен включати в тіло повідомлення пояснення, яке клієнт відобразить користувачеві.**
  - **500 Internal Server Error (Внутрішня помилка сервера)**
    - **Будь-яка внутрішня помилка сервера, яка не входить в рамки інших помилок класу 5xx.**
  - **501 Not Implemented (Відсутня реалізація)**
    - **Сервер не підтримує можливостей, які необхідні для обробки запиту.**
    - **Типова відповідь для випадків, коли сервер не розуміє вказаний у запиті метод.**
  - **502 Bad Gateway (Поганий шлюз)**
    - **Сервер в ролі шлюзу або проксі-серверу отримав повідомлення про невдалий результат виконання проміжної операції.**
  - **503 Service Unavailable (Сервіс недоступний)**
    - **Сервер тимчасово не має можливості обробляти запити з технічних причин (обслуговування, перевантаження тощо).**
  - **504 Gateway Timeout (Шлюз не відповідає)**
    - **Сервер в ролі шлюзу або проксі-серверу не дочекався відповіді від висхідного сервера для завершення поточного запиту.**

Коди HTTP-відповідей

- **5xx – Помилка серверу (Server Error)**
  - **505 HTTP Version Not Supported (Версія HTTP не підтримується)**
    - **Сервер не підтримує або відмовляється підтримувати зазначену в запиті версію протоколу HTTP.**
  - **507 Insufficient Storage (Закінчилося місце)**
    - **Бракує місця для виконання поточного запиту. Проблема може бути тимчасовою.**
  - **510 Not Extended (Відсутнє розширення)**
    - **На сервері відсутнє розширення, яке планує використовувати клієнт.**
    - **Сервер може додатково передати інформацію про доступні йому розширення.**

Структура HTTP-запиту

- **Заголовок повідомлення (Request Message Header): HTTP-запит складається**
  - **Рядок запиту (Request Line) - визначає тип запиту (GET, POST, HEAD тощо);**

    request-method-name request-URI HTTP-version

    - **request-method-nam: метод доступу до ресурсу, наприклад, GET, POST, HEAD, OPTIONS тощо**
    - **request-URI : вказується ресурсу, який запитується;**
    - **HTTP-version: версія HTTP протоколу - HTTP/1.0 та HTTP/1.1;**
    - **Наприклад:**
      - **GET /test.html HTTP/1.1**
      - **HEAD /query.html HTTP/1.0**
      - **POST /index.html HTTP/1.1**
  - **Заголовки запиту (Request Headers) - характеризують тіло повідомлення, параметри передачі та інші відомості (див. наступний слайд);**

Структура HTTP-запиту

- **Заголовок повідомлення (Request Message Header): HTTP-запит складається**
  - **Заголовки запиту (Request Headers) - характеризують тіло повідомлення, параметри передачі, інші відомості та вказуються у вигляді пар - параметр передачі (request-header-name) та значення параметру (request-header-value#):**

    request-header-name: request-header-value1, request-header-value2, ...

    - **Наприклад:**
      - **Host: www.xyz.com**
      - **Connection: Keep-Alive**
      - **Accept: image/gif, image/jpeg, */***
      - **Accept-Language: us-en, fr, cn**
- **Проміжний пустий рядок (Blank Line) – розділяє заголовок та тіло повідомлення**
- **Тіло повідомлення (Request Message Body) - безпосередньо дані запиту.**

Формат HTTP-пакету: відповідь

- **Заголовок повідомлення (Response Message Header): HTTP-відповідь складається з**
  - **Лінії статусу (Status Line) – визначає результат виконання запиту;**

    HTTP-version status-code reason-phrase

    - **HTTP-version: версія HTTP протоколу - HTTP/1.0 та HTTP/1.1;**
    - **status-code: 3-х значний код відповіді;**
    - **reason-phrase: коротке пояснення значення коду;**
    - **Наприклад:**
      - **HTTP/1.1 200 OK**
      - **HTTP/1.0 404 Not Found**
      - **HTTP/1.1 403 Forbidden**
  - **Заголовки відповіді (Response Headers) - характеризують тіло повідомлення, параметри передачі та інші відомості (див. наступний слайд)**

Формат HTTP-пакету: відповідь

- **Заголовок повідомлення (Response Message Header): HTTP-відповідь складається з**
  - **Заголовки відповіді (Response Headers) - характеризують тіло повідомлення та вказуються у вигляді пар - параметр передачі (response-header-nam) та значення параметру (response-header-value#):**

    response-header-name: response-header-value1, response-header-value2, ...

    - **Наприклад:**
      - **Content-Type: text/html**
      - **Content-Length: 35**
      - **Connection: Keep-Alive**
      - **Keep-Alive: timeout=15, max=100**
- **Проміжний пустий рядок (Blank Line) – розділяє заголовок та тіло повідомлення**
- **Тіло повідомлення (Response Message Body) - безпосередньо дані відповіді.**

HTTP заголовки

- **Всі HTTP-заголовки розділяються на чотири основних групи:**
  - **General Headers (Основні заголовки) - повинні включатися в будь-яке повідомлення клієнта і сервера.**
  - **Request Headers (Заголовки запиту) - використовуються тільки в запитах клієнта.**
  - **Response Headers (Заголовки відповіді) - присутні тільки у відповідях сервера.**
  - **Entity Headers (Заголовки сутності) - супроводжують кожну сутність повідомлення.**
    - **Сутності - це корисна інформація, передана в запиті або відповіді.**
    - **Сутність складається з метаінформації (заголовки) і безпосередньо вмісту (тіло повідомлення).**

Заголовок	Група	Короткий опис
Cache-Control	General	Визначає директиви управління механізмами кешування. Для проксі-серверів.
Connection	General	Задає параметри, необхідні для конкретного з'єднання.
Date	General	Дата і час формування повідомлення
Pragma	General	Використовується для спеціальних вказівок, які можуть (опціонально) застосовується до будь-якого одержувачу по всьому ланцюжку запитів/відповідей (наприклад, pragma: no-cache).

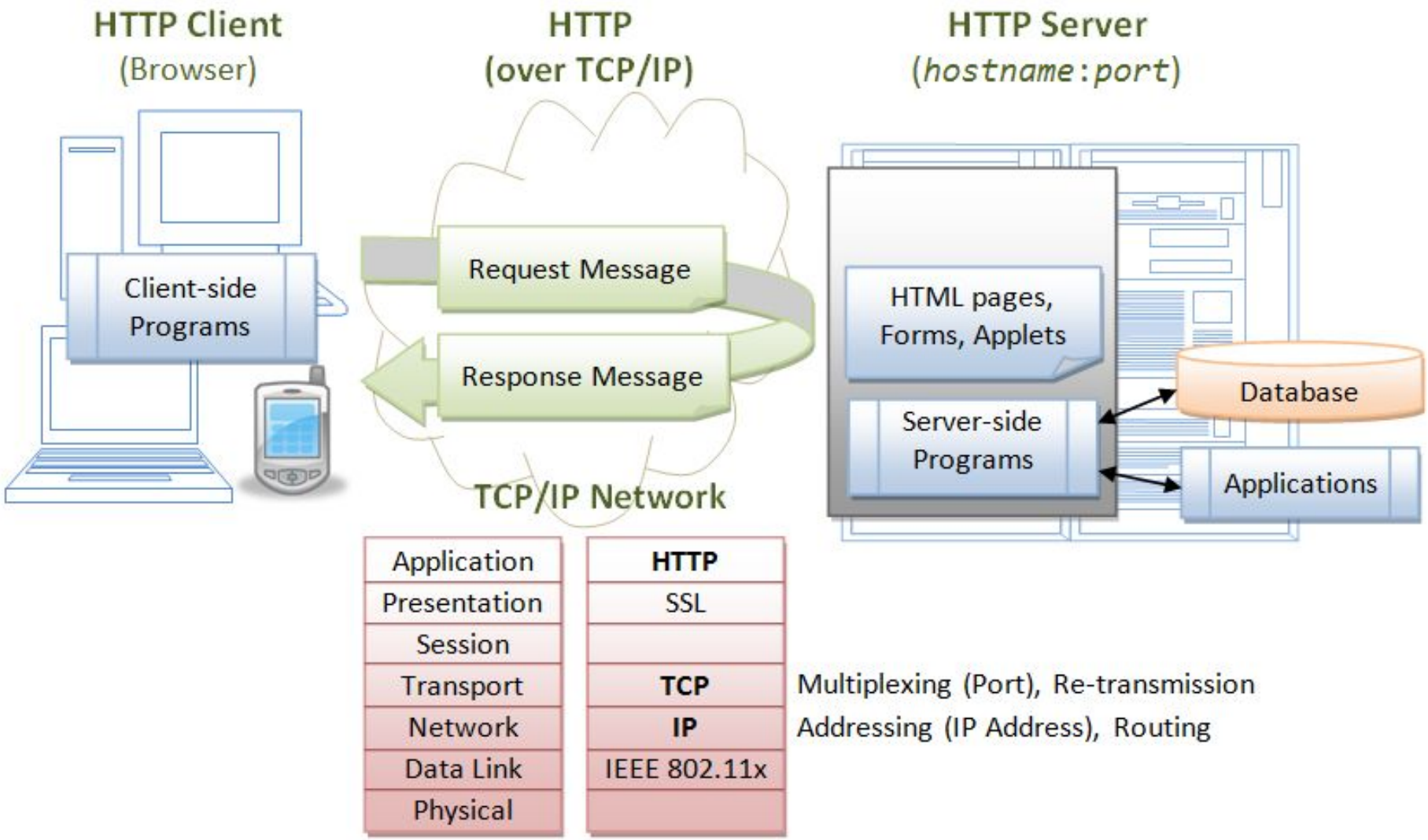
HTTP заголовки

Заголовок	Група	Короткий опис
Transfer-Encoding	General	Задає тип перетворення, застосовного до тіла повідомлення. На відміну від Content-Encoding цей заголовок поширюється на всі повідомлення, а не тільки на сутність.
Via	General	Використовується шлюзами і проксі для відображення проміжних протоколів і вузлів між клієнтом і веб-сервером.
Warning	General	Додаткова інформація про поточний статус, яка не може бути представлена в повідомленні.
Accept	Request	Визначає типи даних, які будуть очікуватися у відповіді.
Accept-Charset	Request	Визначає кодування символів (charset) для даних, які очікуються у відповіді.
Accept-Encoding	Request	Визначає допустимі формати кодування/декодування вмісту (напр, gzip)
Accept-Language	Request	Дозволені мови. Використовується для узгодження передачі.
Authorization	Request	Облікові дані клієнта, запитувача ресурс.
From	Request	Електронна адреса відправника
Host	Request	Ім'я/мережевий адресу [і порт] сервера. Якщо порт не вказаний, використовується 80.
Max-Forwards	Request	Являє механиз обмеження кількості шлюзів і проксі при використанні методів TRACE і OPTIONS.
Proxy-Authorization	Request	Використовується при запитах, що проходять через проксі, що вимагають авторизації
User-Agent	Request	Інформація про клієнтський агент (клієнті)

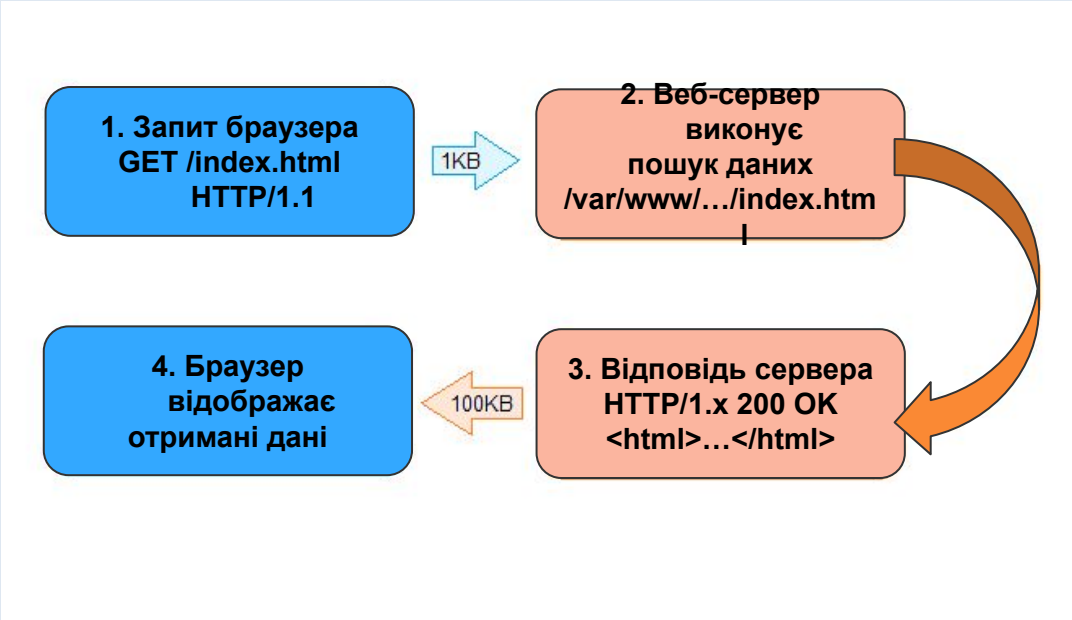
HTTP заголовки

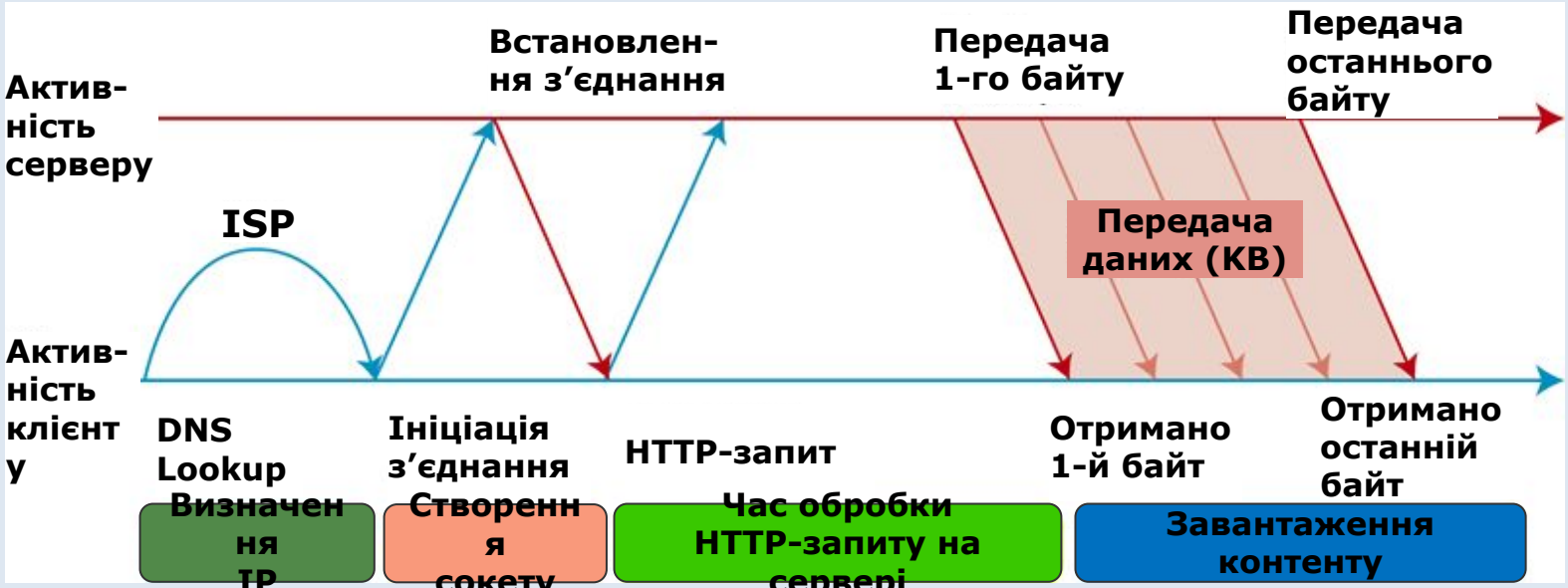
Заголовок	Група	Короткий опис
Location	Response	Адреса перенаправлення
Proxy-Authenticate	Response	Повідомлення про статус з кодом 407.
Server	Response	Інформація про програмне забезпечення сервера, яке відповідає на запит (це може бути як веб- так і проксі-сервер).
Allow	Entity	Список методів, що можуть бути застосовані до запитуваного ресурсу.
Content-Encoding	Entity	Застосовується при необхідності перекодування вмісту (наприклад, gzip / deflated).
Content-Language	Entity	Локалізація вмісту (мови)
Content-Length	Entity	Розмір тіла повідомлення (в октетах)
Content-Range	Entity	Діапазон (використовується для підтримки багатопотокового завантаження або дозавантаження)
Content-Type	Entity	Вказує тип вмісту (mime-type, наприклад text/html) .Часто показування на таблицю символів локалі (charset)
Expires	Entity	Дата/час, після якої ресурс вважається застарілим. Використовується проксі-серверами
Last-Modified	Entity	Дата/час останньої модифікації сутності

Приклад виконання HTTP-запиту

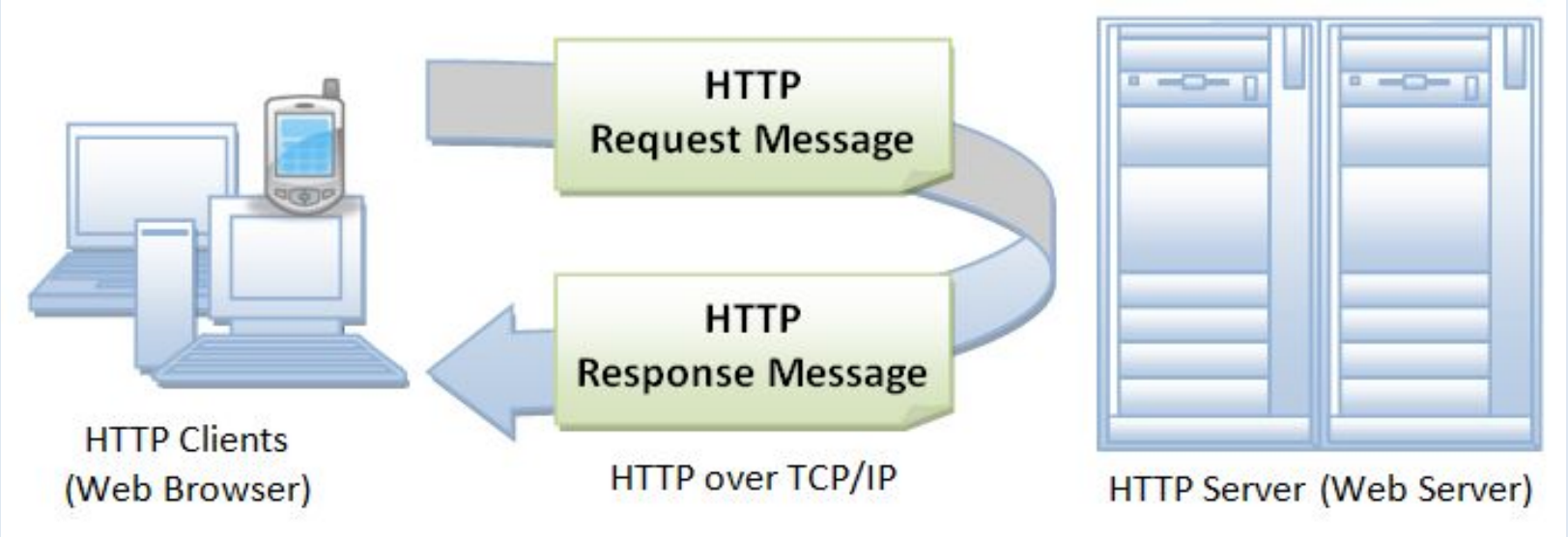


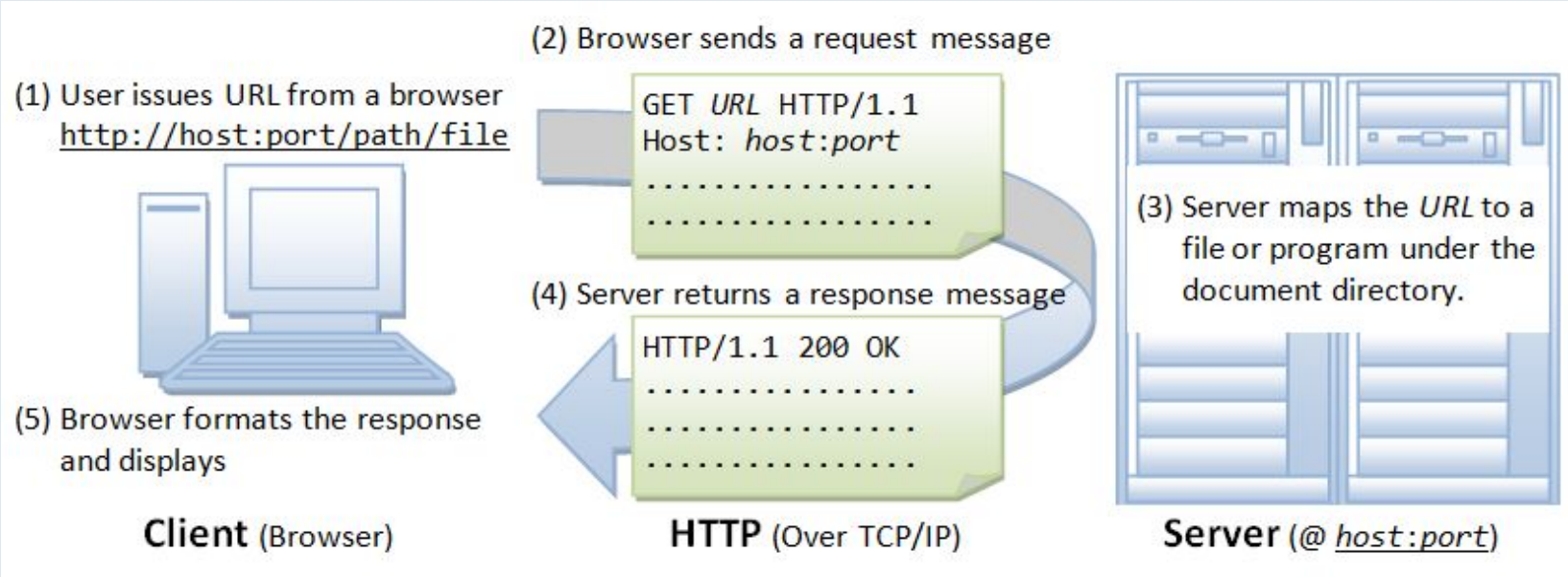
Приклад виконання HTTP-запиту





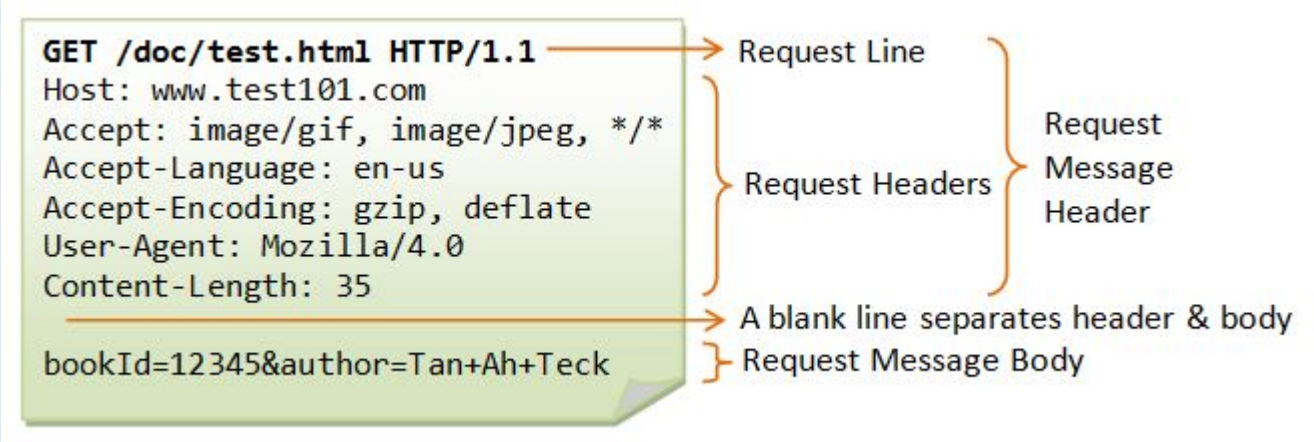
Приклад виконання HTTP-запиту



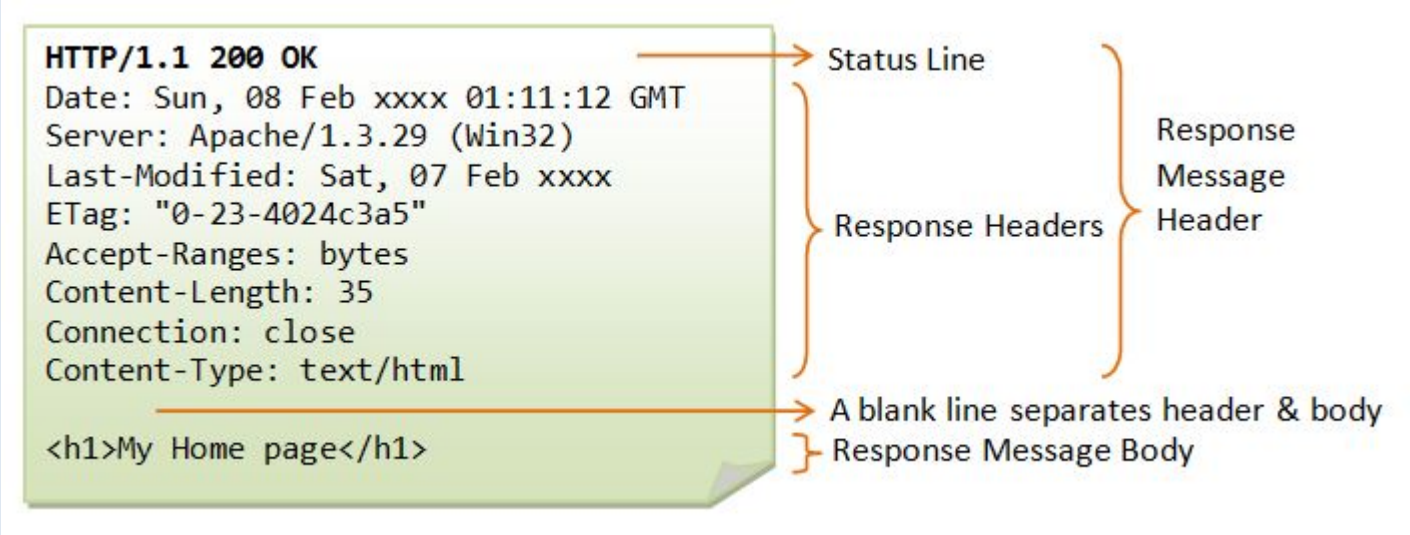


Приклад виконання HTTP-запиту

HTTP-запит



HTTP-відповідь



Приклад HTTP заголовків:

- HTTP/1.0 GET Request ⇒ HTTP/1.1 200 OK
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...

  - **GET /index.html HTTP/1.0**
  - (enter twice to create a blank line)

  - HTTP/1.1 200 OK
  - Date: Sun, 18 Oct 2009 08:56:53 GMT
  - Server: Apache/2.2.14 (Win32)
  - Last-Modified: Sat, 20 Nov 2004 07:16:26 GMT
  - ETag: "10000000565a5-2c-3e94b66c2e680"
  - Accept-Ranges: bytes
  - Content-Length: 44
  - Connection: close
  - Content-Type: text/html
  - X-Pad: avoid browser bug

  - <html><body><h1>It works!</h1></body></html>

  - Connection to host lost.

Приклад HTTP заголовків:

- HTTP/1.0 GET Request ⇒ HTTP/1.1 501 Method Not Implemented
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - get /test.html HTTP/1.0
  - (enter twice to create a blank line)

  - HTTP/1.1 501 Method Not Implemented
  - Date: Sun, 18 Oct 2009 10:32:05 GMT
  - Server: Apache/2.2.14 (Win32)
  - Allow: GET,HEAD,POST,OPTIONS,TRACE
  - Content-Length: 215
  - Connection: close
  - Content-Type: text/html; charset=iso-8859-1

  - <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
  - <html><head>
  - <title>501 Method Not Implemented</title>
  - </head><body>
  - <h1>Method Not Implemented</h1>
  - get to /index.html not supported.
  - </body></html>

Приклад HTTP заголовків:

- HTTP/1.0 GET Request ⇒ HTTP/1.1 404 Not Found
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - GET/t.html HTTP/1.0
  - (enter twice to create a blank line)

  - HTTP/1.1 404 Not Found
  - Date: Sun, 18 Oct 2009 10:36:20 GMT
  - Server: Apache/2.2.14 (Win32)
  - Content-Length: 204
  - Connection: close
  - Content-Type: text/html; charset=iso-8859-1

  - <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
  - <html><head>
  - <title> 404 Not Found </title>
  - </head><body>
  - <h1> Not Found </h1>
  - The requested URL /t.html was not found on this server.
  - </body></html>

Приклад HTTP заголовків:

- HTTP/1.0 GET Request ⇒ **HTTP/1.1 400 Bad Request**
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - **GET /index.html HTTTTTP/1.0**
  - (enter twice to create a blank line)

  - **HTTP/1.1 400 Bad Request**
  - **Date: Sun, 08 Feb 2004 01:29:40 GMT**
  - **Server: Apache/2.2.14 (Win32)**
  - **Connection: close**
  - **Content-Type: text/html; charset=iso-8859-1**

  - **<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">**
  - **<html><head>**
  - **<title> 400 Bad Request </title>**
  - **</head><body>**
  - **<h1> Bad Request </h1>**
  - ** Your browser sent a request that this server could not understand. The request line contained invalid characters following the protocol string.
 **
  - **</body></html>**

Приклад HTTP заголовків:

- HTTP/1.0 GET Request ⇒ HTTP/1.1 403 Forbidden
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - GET /forbidden/index.html HTTP/1.0
  - (enter twice to create a blank line)

  - HTTP/1.1 403 Forbidden
  - Date: Sun, 18 Oct 2009 11:58:41 GMT
  - Server: Apache/2.2.14 (Win32)
  - Content-Length: 222
  - Keep-Alive: timeout=5, max=100
  - Connection: Keep-Alive
  - Content-Type: text/html; charset=iso-8859-1

  - <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
  - <html><head>
  - <title>403 Forbidden</title>
  - </head><body>
  - <h1>Forbidden</h1>
  - You don't have permission to access /forbidden/index.html on this server.
  - </body></html>

Приклад HTTP заголовків:

- HTTP/1.0 GET Request ⇒ HTTP/1.1 301 Moved Permanently
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - GET /testdir HTTP/1.1
  - Host: 127.0.0.1
  - (enter twice to create a blank line)

  - HTTP/1.1 301 Moved Permanently
  - Date: Sun, 18 Oct 2009 13:19:15 GMT
  - Server: Apache/2.2.14 (Win32)
  - Location: http://127.0.0.1:8000/testdir/
  - Content-Length: 238
  - 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>
  - The document has moved here.
  - </body></html>

Приклад HTTP заголовків:

- **HEAD /index.html HTTP/1.0 ⇒ HTTP/1.1 200 OK**
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - **HEAD /index.html HTTP/1.0**
  - (enter twice to create a blank line)

  - **HTTP/1.1 200 OK**
  - **Date: Sun, 18 Oct 2009 14:09:16 GMT**
  - **Server: Apache/2.2.14 (Win32)**
  - **Last-Modified: Sat, 20 Nov 2004 07:16:26 GMT**
  - **ETag: "10000000565a5-2c-3e94b66c2e680"**
  - **Accept-Ranges: bytes**
  - **Content-Length: 44**
  - **Connection: close**
  - **Content-Type: text/html**
  - **X-Pad: avoid browser bug**

Приклад HTTP заголовків:

- **OPTIONS http://www.amazon.com/ HTTP/1.1 ⇒ HTTP/1.1 200 OK**
  - telnet> open 127.0.0.1 8000
  - Connecting To 127.0.0.1...
  - **OPTIONS http://www.amazon.com/ HTTP/1.1**
  - **Host: www.amazon.com**
  - **Connection: Close**
  - (enter twice to create a blank line)

  - **HTTP/1.1 200 OK**
  - **Date: Fri, 27 Feb 2004 09:42:46 GMT**
  - **Content-Length: 0**
  - **Connection: close**
  - **Server: Stronghold/2.4.2 Apache/1.3.6 C2NetEU/2412 (Unix)**
  - **Allow: GET, HEAD, POST, OPTIONS, TRACE**
  - **Connection: close**
  - **Via: 1.1 xproxy (NetCache NetApp/5.3.1R4D5)**
  - (blank line)