

Організація та принципи роботи веб-сервісу. Протокол HTTP.

Важливість WWW — World Wide Web

URL

Єдиний вказівник ресурсів URL — Uniform Resource Locator — єдинообразний локатор (визначник місцезнаходження) ресурсу. Раніше називався **Universal Resource Locator** — універсальний локатор ресурсу. **URL — це стандартизований спосіб запису адреси ресурсу в мережі Інтернет.**

Історія

URL було винайдено в 1990 році в стінах Європейської ради з питань ядерних досліджень (фр. Conseil Européen pour la Recherche Nucléaire, CERN) в Женеві, Швейцарія.

URL став фундаментальною інновацією в Інеті.

Початково URL було призначено для позначення місць розміщення ресурсів (найчастіше файлів) у Всесвітній павутині.

Зараз URL застосовується для позначення адрес майже всіх ресурсів Інет.

Стандарт URL закріплено в документі RFC 1738. Зараз URL позиціонується як частина більш загальної системи ідентифікації ресурсів URI, сам термін URL поступово поступається більш широкому терміну **URI**.

URI - Uniform Resource Identifier

Структура URL

Початково локатор URL розроблявся як система для максимально природнього вказування на місцезнаходження ресурсів в мережі.

Локатор повинен був легко розширюватися і використовувати лише обмежений набір ASCII символів (напр., пробіл ніколи не застосовується в URL). В зв'язку з цим, виникла наступна традиційна форма запису URL:

<схема>://<логін>:<пароль>@<хост>:<порт>/<URL шлях>

схема

схема звертання до ресурсу, в більшості випадків мається на увазі мережевий протокол.

логін

ім'я користувача, що використовується для доступу до ресурсу

пароль

пароль, який асоційовано з вказаним іменем користувача

хост

повністю прописане доменне ім'я хоста в системі DNS або IP-адреса хоста в формі чотирьох десяткових чисел, разделённых точками.

порт

порт хоста для підключення

URL шлях

уточнююча інформація про місце знаходження ресурсу (залежить від протоколу)

Схеми (протоколи) URL

Загальноприйняті схеми (протоколи) URL включають:

- ftp — Протокол передачі файлів FTP
- http — Протокол передачі гіпертексту HTTP
- https — Спеціальна реалізація протоколу HTTP, що використовує шифрування (як правило, SSL або TLS)
- mailto — Адреса електронної пошти
- file — ім'я локального файлу

Протокол HTTP

RFC 2616 Стандарт HTTP/1.1

HTTP - протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від **H**yper **T**ext **T**ransfer **P**rotocol, протокол передачі гіпер-текстових документів

Основним призначенням протоколу HTTP є передача веб-сторінок (текстових файлів з розміткою HTML), хоча за допомогою його з успіхом передаються і інші файли, як пов'язані з веб-сторінками (зображення і додатки), так і не пов'язані з ними (у цьому HTTP конкурує з складнішим FTP).

HTTP припускає, що клієнтська програма — веб-браузер — здатна відобразити гіпертекстові веб-сторінки і файли інших типів в зручній для користувача формі. Для правильного відображення HTTP дозволяє клієнтові дізнатися мову і кодування веб-сторінки і/або запитати версію сторінки в потрібних мові/кодуванні, використовуючи позначення із стандарту MIME.

Структура протоколу

HTTP — протокол прикладного рівня, аналогічними йому являються FTP і SMTP. Обмін повідомленнями йде по звичайній схемі «запит-відповідь». **Для ідентифікації ресурсів HTTP використовує глобальні URI.**

На відміну від багатьох інших протоколів, **HTTP не зберігає свого стану**. Це означає відсутність збереження проміжного стану між парами «запит-відповідь».

Компоненти, використовуючі HTTP, можуть самостійно здійснювати збереження інформації про стан, пов'язаної з останніми запитами і відповідями. Браузер, що посилає запити, може відстежувати затримки відповідей. Сервер може зберігати IP-адреси і заголовки запитів останніх клієнтів. Проте сам протокол не обізнаний про попередні запити і відповіді, в нім не передбачена внутрішня підтримка стану, до нього не пред'являються такі вимоги.

Кожен запит/відповідь складається з трьох частин:

1. стартовий рядок;
2. заголовки;
3. тіло повідомлення, що містить дані запиту, запитаний ресурс або опис проблеми, якщо запит не виконано.

Запит

Стартові рядки розрізняються для запиту і відповіді. Рядок запиту виглядає так:

«Метод» «URI» HTTP/«Версія»

де «Метод» можливо:

OPTIONS

Повертає методи HTTP, які підтримуються сервером. Цей метод може служити для визначення можливостей веб-сервера-сервера.

GET

Запрошує вміст вказаного ресурсу. Запитаний ресурс може приймати параметри (наприклад, пошукова система може приймати як параметр шуканий рядок). Вони передаються в рядку URI (наприклад: <http://www.example.net/resource?param1=value1¶m2=value2>). Згідно стандарту HTTP, запити типу GET вважаються ідемпотентними — багатократне повторення одного і того ж запиту GET повинне приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами). Це дозволяє кешувати відповіді на запити GET.

HEAD

Аналогічний методу GET, за винятком того, що відповідає сервера відсутнє тіло. Це корисно для витягання мета-інформації, заданої в заголовках відповіді, без пересилання всього вмісту. Зокрема, клієнт чи проксі, звіривши час, таким чином може переконатися, що сторінка на сервері не змінилася від часу попереднього запиту.

POST

Передає призначені для користувача дані (наприклад, з HTML-форми) 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 передавані дані (у прикладі з блогами — текст коментаря) включаються в тіло запиту. На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багатократне повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожної відправки коментаря з'являтиметься одна копія цього коментаря).

PUT

Завантажує вказаний ресурс на сервер.

DELETE

Видаляє вказаний ресурс.

TRACE

Повертає отриманий запит так, що клієнт може побачити, що проміжні сервера додають або змінюють в запиті.

CONNECT

Для використання разом з прокси-серверами, які можуть динамічно перемикатися в тунельний режим SSL.

В основному використовуються методи GET і POST.

Відповідь сервера

Перший рядок відповіді виглядає так:

HTTP/«Версія» «Код статусу» «Опис статусу»

Найтипівіші статуси:

- 200 OK — запит виконаний успішно;
- 403 Forbidden — доступ до запитаного ресурсу заборонений;
- 404 Not Found — запитаний ресурс не знайдений.

Заголовки

Заголовки HTTP — це рядки, кожна з яких складається з імені параметра, за яким слідує двокрапка і його значення. Вони несуть інформацію для браузера або для серверних програм (таких, як CGI-додатки). Між заголовками і тілом обов'язково повинен бути порожній рядок.

Приклад HTTP діалогу

Запит

```
GET /lect HTTP/1.1
Host: student.mr.lviv.ua
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.7) Gecko/20070914
Firefox/2.0.0.7
Connection: close
```

Відповідь

```
HTTP/1.0 200 OK
Server: Apache
Content-Language: uk
Content-Type: text/html; charset=utf-8
Content-Length: 1234
```

(далі йде текст запитаної сторінки)