Hosting WWW
Bezpieczeństwo hostingu WWW

Dr Michał Tanaś (http://mtanas.web.amu.edu.pl/)

Protokoły WWW

- Protokoły transportowe
  - HTTP – HyperText Transfer Protocol
  - HTTPS – HTTP Secured
- Format adresów WWW
  - URI – Uniform Resource Identifier
- Protokoły dynamicznego generowania zawartości
  - CGI – Common Gateway Interface

Protokoły WWW

- Protokoły dostępu do danych
  - SOAP – Simple Object Access Protocol
  - JSON – Java Script Object Notation
- Protokół tworzenia asynchronicznych stron WWW
  - AJAX – Asynchronous Java Script and XML
- Zbiór zasad tworzenia protokołów i usług WWW
  - REST – Representational State Transfer
  - Loose Coupling

HTTP

- Podstawowy protokół przesyłania treści WWW
- Działa na poziomie warstwy 7 (warstwa aplikacji) modelu ISO/OSI
- Wykorzystuje protokoły niższych warstw (TCP/IP)
- Domyślne porty
  - 80/TCP – port HTTP
  - 443/TCP – port HTTPS
  - 8080/TCP – port HTTP proxy

HTTP

- W wersji HTTP/1.1 jest protokołem
  - Bezpołączeniowym (aczkolwiek wymaga nawiązanego połączenia TCP)
  - Synchronicznym
  - Stroną inicjującą jest zawsze klient (przeglądarka)

HTTP

- Historia HTTP
  - HTTP v0.9 (1991) – pierwsza wersja
  - HTTP v1.0 (1996)
    - rozszerzony format nagłówka
    - dodanie nowych metod
    - możliwość użycia wersji szyfrowanej (HTTPS)
    - możliwość podtrzymywania połączeń (keep-alive)
    - możliwość wysyłania kolejnych żądań przy pomocy istniejących połączeń
  - HTTP/1.1 (1997) – jeszcze używany, stopniowo wypierany przez HTTP/2 i HTTP/3
    - domyślne podtrzymywanie połączeń

HTTP

- Historia HTTP
  - HTTP/2 (2015) – aktualny standard
    - Bazuje na nieoficjalnym rozszerzeniu SPDY wymyślonym przez Google
    - Równoległe wykonywanie żądań
    - Kompresja „w locie” nagłówków HTTP
    - Możliwość komunikacji asynchronicznej inicjowanej przez serwer (tzw. server push)

HTTP

- Historia HTTP
  - HTTP/3 (draft 2022) – formalnie niezatwierdzony, ale używany
    - Zmiana protokołu transportowego z TCP na QUIC („Quick UDP Internet Connection”)
      - własny protokół Google, używany zamiast TCP
      - wykorzystuje UDP, więc nie wymaga zmian w stosie TCP/IP w systemach operacyjnych
      - brak narzutu na 3-way handshaking
      - **połączenie jest utrzymywane gdy zmienia się adres IP jednej ze stron (niewykonalne przy TCP)**
    - Wymaga używania TLS (dostępny tylko HTTPS, brak HTTP)

HTTP

Odsetek serwerów WWW używających:

HTTP 1.1 – 35%

HTTP 2.0 – 35%

HTTP 3.0 – 30%

Czerwiec 2024, źródło: W³Techs Web Technology Surveys

HTTP/1.1

- Żądanie HTTP/1.1
  - Składa się z nagłówka i opcjonalnie z treści
  - Nagłówek ma postać tekstową
  - Treść ma postać
    - Tekstową
    - Binarną zakodowaną – kodowanie jest uzgadniane pomiędzy przeglądarką a serwerem

HTTP/1.1

- Ogólna postać żądania HTTP/1.1

metoda URI wersja

opcje

opcje

…

HTTP/1.1

- Ogólna postać żądania HTTP/1.1 - przykład

GET http://mtanas.web.amu.edu.pl/ HTTP/1.1

Accept-Charset: utf-8;q=0.8, iso8859-2;q=0.5

Accept-Encoding: gzip

HTTP/1.1

- Najważniejsze metody
  - GET – żądanie przesłania określonej treści. Najczęściej używana metoda HTTP.
  - POST – metoda używana do przesyłania treści formularzy
  - PUT – przesłanie określonej treści (najczęściej pliku) do serwera. Używane przy uploadzie plików.
  - DELETE – usunięcie określonej treści z serwera. Używane przy uploadzie plików.

HTTP/1.1

- Najważniejsze metody
  - HEAD – podobne do GET, ale przesyłany jest wyłącznie nagłówek odpowiedzi, bez jej treści. Używane np. do sprawdzenia w jakim kodowaniu jest dana strona WWW.
  - OPTIONS – odpytanie serwera jakie metody mogą byś użyte dla danego URI. Zamiast URI można dać * , wtedy serwer informuje jakich w ogóle metod można na nim używać.

URI

- URI – Uniform Resource Identifier
- Jest adresem, który identyfikuje daną treść w Internecie
- Jest unikalny – żadne dwie treści w całym Internecie nie mogą mieć tego samego URI (wyjątek – prywatne zakresy adresowe IP)

URI

- Teoretycznie URI składa się z dwóch części
  - URL – Uniform Resource Locator
  - URN – Uniform Resource Name
- Ponieważ jednak URN jest bardzo rzadko używane
- … więc w praktyce URI i URL są synonimami
- Uwaga, URN nie należy mylić z nazwą pliku!

URI

- Ogólna postać URI

protokół://login:hasło@serwer:port/ścieżka/plik?polecenie

- Konieczne są protokół i serwer, pozostałe części są opcjonalne

- Przykład URI

https://usosweb.amu.edu.pl/kontroler.php?_action=actionx:news/default()

HTTP/1.1

- Opcje (pola nagłówka)
- Określają dodatkowe cechy żądania lub odpowiedzi HTTP
- Mają postać tekstową – jedna linia jedna opcja
- Standard HTTP nie ogranicza ani liczby pól w nagłówku ani długości pojedynczego pola
- … ale w praktyce serwery WWW narzucają własne ograniczenia
- … np. Apache ogranicza
  - liczbę pól do 100
  - długość pojedynczego pola do 8KB

HTTP/1.1

- Najważniejsze opcje (pola nagłówka) mogące występować i w żądaniu i w odpowiedzi HTTP
  - Content-length: - długość treści żądania/odpowiedzi (bez nagłówka) w bajtach
  - Content-MD5: - suma kontrolna treści żądania/odpowiedzi (bez nagłówka)
  - Content-Type: - typ MIME przesyłanego pliku
  - Date: - czas wysłania żądania/odpowiedzi
  - Upgrade: - prośba o zmianę protokołu

HTTP/1.1

- Najważniejsze opcje (pola nagłówka) żądania HTTP
  - Accept: – typy MIME, które mogą być wysłane w odpowiedzi
  - Accept-Charset: – dopuszczalne kodowanie znaków w odpowiedzi
  - Accept-Encoding: - dopuszczalne formaty kompresji
  - Accept-Language: - dopuszczalne wersje językowe strony
  - Authorization: - hasło do strony
  - Connection: keep-alive – żądanie podtrzymania połączenia w HTTP v1.0. Ignorowane w HTTP/1.1

HTTP/1.1

- Najważniejsze opcje (pola nagłówka) żądania HTTP
  - Cookie: - treść ciasteczka przesyłana przez przeglądarkę do serwera
  - DNT: - „do not track”, może mieć wartość 0 albo 1
  - User-Agent: - informacje o przeglądarce, serwer może je wykorzystać do formatowania strony
  - Warning: - ostrzeżenie (dowolny tekst wyświetlany przez przeglądarkę)

HTTP/1.1

- Najważniejsze opcje (pola nagłówka) odpowiedzi HTTP
  - Allow: - dopuszczalne metody dla danego obiektu
  - Content-Disposition: - domyślna nazwa pod którą przeglądarka zapisze plik
  - Content-Encoding: - typ kompresji użytej dla treści odpowiedzi
  - Content-Language: - wersja językowa przesyłanej strony
  - Refresh: - URI który przeglądarka ma otworzyć po 5 sekundach
  - Server: - informacje o serwerze, przeglądarka zazwyczaj je ignoruje

HTTP/1.1

- Najważniejsze opcje (pola nagłówka) odpowiedzi HTTP
  - Set-Cookie: - ciasteczko przesyłane przez serwer do przeglądarki
  - WWW-Authenticate: - typ wymaganego logowania się do strony
  - X-Powered-By: - aplikacja która wygenerowała stronę tworzoną dynamicznie (np. PHP)
  - X-UA-Compatible: - zalecany silnik przeglądarki, używane do wymuszenia użycia innego silnika niż domyślny (np. tryby kompatybilności, Chrome Frame)

MIME

- MIME (Multipurpose Internet Mail Extension) powstał jako system oznaczania typu załącznika w e-mailach
- Potem zaczęto go wykorzystywać do oznaczania typu danych przesyłanych w żądaniach/odpowiedziach HTTP
- … gdzie wykorzystywane jest głównie pole Content-Type:
- … mające ogólną postać:

Content-Type: typ/podtyp

MIME

- Możliwe typy pola Content-Type: to
  - text
  - image
  - audio
  - video
  - application

MIME

- Podtyp określa konkretny format danych
- np.
  - text/plain
  - text/html
  - text/xml
  - text/css
  - image/jpg
  - image/vnd.djvu
  - audio/x-aac
  - video/mp4

MIME

- Typ „application” jest w praktyce używany do wszystkich typów danych nie objętych pozostałymi typami, np.
  - application/javascript
  - application/pdf
  - application/x-7z-compressed

MIME

- Oznaczenia specjalne
  - Typ „application/octet-stream” oznacza „dowolny ciąg bajtów. W praktyce oznacza się tak dane binarne nie będące w żadnym ze standardowych formatów.
  - Typy/podtypy zaczynające się od „x-” są nieoficjalne, tzn. niezarejestrowane w IANA
  - Typy/podtypy zaczynające się od „vnd.” (skrót od „vendor”) są zarządzane przez organizacje inne niż IANA

HTTP/1.1

- Po każdym żądaniu serwer informuje przeglądarkę o jego przyjęciu lub odrzuceniu
- Informacja ta ma postać 3-cyfrowego kodu
- … w postaci xyy
- … gdzie x to 1-cyfrowy typ odpowiedzi
- … a yy to 2-cyfrowy kod odpowiedzi

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 1yy – częściowe żądanie przyjęte, można kontunuować
    - 100 „Continue” – używane przy metodzie POST, sygnalizuje że serwer jest gotowy go przyjęcia następnego pola
  - 2yy – żądanie wykonane pomyślnie
    - 200 „OK” – poprawne wykonanie żądania synchronicznego
    - 201 „Created” – jw. używane przy uploadzie plików
    - 202 „Accepted” – poprawne przyjęcie (ale niekoniecznie wykonanie) żądania asynchronicznego

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 3yy – przekierowanie
    - 301 „Moved permanently” – informacja dla przeglądarki żeby używała innego URI.
  - 4yy – błąd po stronie przeglądarki
    - 400 „Bad request” – serwer nie wie o co chodzi w żądaniu, zazwyczaj oznacza błąd składni
    - 401 „Unauthorized” – żądanie wymaga zalogowania się
    - 403 „Forbidden” – dostęp zabroniony (trwale)
    - 404 „Not found” – żądanie dotyczy nieistniejącego obiektu

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 4yy – błąd po stronie przeglądarki
    - 405 „Method not allowed” – nie można zastosować żądanej metody dla danego obiektu z przyczyn obiektywnych (np. pomylenie GET z POST)
    - 406 „Not acceptable” – nie można wykonać żądania, ponieważ nie znaleziono sposobu który byłby równocześnie akceptowalny przez serwer i przeglądarkę (np. przeglądarka akceptuje tylko UTF-8 a serwer tylko ISO8859-2)
    - 408 „Timeout” – serwer zerwał połączenie ponieważ nie doczekał się kolejnego żądania

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 4yy – błąd po stronie przeglądarki
    - 413 „Payload too large” – treść żądania jest zbyt duża, używane np. przez webmaile do sygnalizacji że załącznik jest za duży
    - 414 „URI too long” – zbyt długie URI w żądaniu, zazwyczaj zbyt dużo parametrów lub pól GET
    - 415 „Unsupported media type” – wysyłany plik ma zły typ MIME, używane np. przez wydawnictwa żeby wymusić określony format wysyłanych plików
    - 418 „I’m a teapot” – sygnalizuje że serwer jest czajnikiem i spełnia jedynie żądania „zrób kawę” albo „zrób herbatę”. Dowcip prima-aprilisowy

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 4yy – błąd po stronie przeglądarki
    - 419 „Authentication timeout” – wygaśnięcie sesji, konieczne ponowne zalogowanie się
    - 421 „Misdirected request” – błędne przekierowanie
    - 426 „Upgrade required” – przeglądarka używa jakiegoś przestarzałego protokołu, zazwyczaj przestarzałej wersji TLS
    - 429 „Too many requests” – zbyt dużo żądań od konkretnego klienta w jednostce czasu
    - 431 „Header too large” – zbyt długi nagłówek lub któreś z jego pól, zazwyczaj błąd w składni

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 5yy – błąd po stronie serwera
    - 500 „Internal server error” – serwer nie mógł wykonać żądania, nie wiadomo dlaczego. Np. błąd w skrypcie PHP
    - 501 „Not implemented” – serwer nie posiada jeszcze funkcji, która jest wymagana dla spełnienia żądania. Np. zażądano kompresji, której serwer nie obsługuje.
    - 502 „Bad gateway” – proxy otrzymało złą odpowiedź od serwera docelowego
    - 503 „Service unavailable” – serwer czasowo niedostępny, zazwyczaj przeciążony

HTTP/1.1

- Najważniejsze kody odpowiedzi serwera
  - 5yy – błąd po stronie serwera
    - 504 „Gateway timeout” – proxy nie otrzymało odpowiedzi od serwera docelowego
    - 505 „HTTP version not supported” – serwer nie obsługuje danej wersji HTTP

HTTP/1.1

- Odpowiedzi HTTP w kulturze popularnej: