

pawel.rajba@gmail.com, <http://itcourses.eu/>

HTTP

Agenda

- Adresy zasobów
- Rodzaje zawartości
- Negocjacja treści
- Komunikacja
- Buforowanie
- HTTP Request/Response
- Nagłówki
- Bezstanowość
- Cookies
- Narzędzia

Adresy zasobów

- URL, <http://www.ietf.org/rfc/rfc3986.txt>
 - Schemat adresu
<scheme>://<authority><path>?<query>#fragment
- Najczęstsze zastosowanie
 - Adresy stron w Sieci
<http://localhost/index.php>
- Fragmenty
 - Nie są wysyłane do serwera
 - Przykład:
<http://localhost/index.php?z1=w1&z2=w2>
- Lokalizacja fizyczna vs. dynamiczna
- Pełny przykład
 - <http://localhost:8080/path?q=text#wyniki>

Adresy zasobów

- Przykłady innych schematów
 - <ftp://user:pass@serwer.pl:21/dokument.txt>
 - <mailto:pawel@ii.uni.wroc.pl>
 - <news://pl.comp.os.linux/>
 - <telnet://156.17.4.4/>

Rodzaje zawartości

- Standard MIME
 - Określony przez type/subtype
 - application/json
 - image/png
 - image/gif
 - text/xml
 - text/html
 - text/plain
 - Rozszerzenie jest ostatnim miejsce, po którym rozpoznawany jest rodzaj zawartości
 - Chociaż mapowania są częścią konfiguracji serwera WWW

Negocjacja treści

- Treść może być dostępna
 - W różnych formatach
 - np. możemy wybrać, czy chcemy XML czy JSON
 - W różnych językach
 - Przykład: opcje IE, zmieniamy język i pobieramy:
<http://go.microsoft.com/fwlink/?LinkId=69157>

Komunikacja

- Jedna transakcja
 - HTTP Request
 - HTTP Response
 - Określone formaty, obecnie HTTP 1.1
 - <http://www.ietf.org/rfc/rfc2616.txt>
- Jak ręcznie wymieniać komunikaty?
 - telnet
 - telnet host 80 (kwestia set localecho)
 - Fiddler

Komunikacja

- Pobieranie strony
 - Obecne strony mają dużo zasobów – przyspieszenie pobieranie poprzez równoczesne połączenia
 - Specyfikacja: max 2 jednoczesnych połączeniach do hosta
 - Stan obecny: przeglądarki otwierają więcej połączeń

Version	HTTP 1.0 server (broadband connection)	HTTP 1.1 server (broadband connection)	HTTP 1.0 server (dial-up connection)	HTTP 1.1 server (dial-up connection)
Internet Explorer 7 and earlier	4	2	4	2
Internet Explorer 8	6	6	4	2

Komunikacja

- Trwałe połączenie
 - Mechanizm pozwala na wiele żądań w ramach jednego połączenia
 - Serwer może decydować, czy akceptuje takie podejście poprzez nagłówek
 - Connection: close

Komunikacja

- Forward Proxy (czyli po prostu proxy)
 - Serwer pośredniczący pomiędzy klientem a docelowym serwerem
 - Jest po stronie klienta
 - Zastosowanie
 - Przyspieszenie transmisji
 - Filtrowanie treści, usuwanie tajnych informacji
 - Blokowanie dostępu do facebooka, twittera, allegro, ...

Komunikacja

- Reverse proxy
 - Jest po stronie serwera
 - Zastosowanie
 - Loadbalancing
 - Kierowanie żądań o statyczne zasoby do dedykowane serwera (tzw. asset server)
 - Dodatkowa warstwa zabezpieczająca
 - Nakładanie dodatkowych elementów jak kompresja, SSL
 - Buforowanie często żądanych zasobów

Buforowanie (cache)

- Metody HTTP a buforowanie
 - GET – buforowane, POST, PUT, DELETE – nie
- Public cache
 - Zasoby są buforowane na serwerach proxy
- Private cache
 - Zasoby są buforowane w przeglądarce
- Co wpływa na buforowanie
 - Cache-Control: public, private, no-cache, no-store
 - Dodatkowo można dodać
 - max-age: liczba-sekund
 - must-revalidate
 - Expires: data (np. w przeszłości)
 - Pragma: no-cache
 - Etag: „hash” zasobu – jeśli się zmieni, zasób musi zostać pobrany
- Sztuczka: dodanie ?wersja=1.2 do CSS, JS, itd.

HTTP Request

- Struktura żądania
 - [metoda] [zasób] [wersja]
[nagłówki]
[treść]
 - Przykład
 - GET / HTTP/1.1
Host: kursy24.eu

HTTP Request

- Metody
 - GET – pobranie zasobu
 - POST – aktualizacja zasobu
 - PUT – wgranie zasobu
 - DELETE – usunięcie zasobu
 - HEAD – pobranie nagłówek dla zasobu
- Poza POST użycie metod analogicznie do GET

HTTP Request

- Użycie metody POST
 - Dane są kodowane zwykle na dwa sposoby
 - x-www-form-urlencoded
 - multipart/formdata (format MIME, RFC1867)
 - Przykład, wysyłanie prostych danych
 - POST /htdocs/processdata.php HTTP/1.1
Host: localhost
Content-type: application/x-www-form-urlencoded
Content-length: 48

imie=Jan&nazwisko=Kowalski&miasto=Warszawa

HTTP Request

- Kilka popularnych nagłóweków
 - Referer
 - User-Agent
 - Accept
 - Accept-Language
 - Cookie
 - If-Modified-Since
 - Date

HTTP Response

- Struktura odpowiedzi
 - [wersja] [status] [opis]
[nagłówki]
[body]
 - Nagłówki X-cos
 - Niestandardowe dla poszczególnych serwerów

HTTP Response

- Kody odpowiedzi
 - 100-199 : Informacja
 - 200-299 : Żądanie zostało przetworzone poprawnie
 - 300-399 : Żądanie zostało przeadresowane
 - 400-499 : Błąd po stronie klienta (np. błędny URL)
 - 500-599 : Błąd po stronie serwera

HTTP Response

- Często spotykane statusy
 - 200 OK
 - 206 Partial Content
 - 301 Moved Permanently
 - 302 Moved Temporarily
 - 304 Not Modified
 - 400 Bad Request
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
 - 405 Method Not Allowed
 - 500 Internal Server Error
 - 503 Service Unavailable

Nagłówki

- Nagłówki HTTP dzielimy na 4 grupy
 - Pola ogólnego przeznaczenia
 - Pola nagłówka żądania (klient)
 - Pola nagłówka odpowiedzi (serwer)
 - Pola nagłówka zawartości
- Wielkość znaków w nazwach pól nie ma znaczenia
- Kolejność pól również nie ma znaczenia
- Dalej przejrzymy kilka nagłówków

Nagłówki ogólnego przeznaczenia

- Cache-Control – określa sposób buforowania
 - Przykładowe wartości żądania
 - no-cache, no-store, max-age
 - Przykładowe wartości odpowiedzi
 - public, private, no-cache, no-store, must-revalidate, max-age
- Connection – rodzaj połączenia
 - Connection: { keep-alive | close }
- Date – moment wysłania żądania/odpowiedzi
 - Date: format-daty
 - Czas jest zawsze względem GMT
 - Format powinien być zgodny z RFC 1123;

Nagłówki żądania

- Host
 - Accept
 - Accept-Language
 - If-Modified-Since
 - User-Agent
 - Accept-Charset
 - Authorization
 - Range
-
- Priorytety określane przez q-range, np.
 - Accept: typ/podtyp [q=ranga] [,...] (0<=ranga<=1)
 - Accept: text/*, image/gif
 - Accept-Charset – określa preferowane alfabety

Nagłówki odpowiedzi

- Accept-Ranges
- Retry-After
- Set-Cookie
- Etag
- Server
- WWW-Authenticate
- Location

Nagłówki pola zawartości

- Allow
- Content-Encoding
- Content-Language
- Content-Length
- Content-Range – określa pobrany fragment
 - Content-Range: początek-koniec/rozmiar
- Content-Type
- Expires
- Last-Modified

Bezstanowość

- Każda para request-response stanowi jedną transakcję
 - Czyli nie ma żadnego związku pomiędzy kolejnymi żądaniem
- Większość aplikacji jest „stanowa”
 - Pamiętanie stanu w sesji
 - Sesja może być pamiętana po stronie
 - Klienta (np. ViewState w ASP.NET)
 - Serwera (Session)
 - Pojęcie „sticky session”

Bezstanowość

- Skąd serwer wie, której sesji użyć?
 - Poprzez identyfikator przekazywany od klienta
 - Jak przekazać identyfikator?
 - Parametr GET
 - Cookies

Cookies

- Co to są „ciastka”? RFC6265
 - Rozmiar ciastka: do 4KB
- Po stronie klienta
 - Cookie: SESSIONID=aasoizirj
- Po stronie serwera
 - Set-Cookie: SESSIONID=aasoizirj;
domain=kursy24.eu;
path=/

Cookies

- Znaczenie parametrów
 - domain
 - path
- Ciastka sesyjne i trwałe
 - Expires
- Dodatkowe parametry:
 - Secure
 - HttpOnly

Narzędzia

- Fiddler
- Postman
- Narzędzia developerskie Google (network)