>>> PROTOKÓŁ HTTP
>>> od 1.0 do 3.0

Name: Jakub Sydor
Date: 16/01/2019

>>> Spis treści

>>> Wstęp

Protokół HTTP został stworzony przez Tim Berners-Lee w roku 1989 jako standard komunikacji dla sieci World Wide Web.

Znajduje się na szczycie modelu ISO/OSI w warstwie Sesji.

Protokół HTTP definiuje, w jaki sposób zainicjować połączenie, jak wymieniać informację oraz zasoby i jak połączenie zakończyć.

>>> Budowa zapytania

Obraz: Przykładowe zapytanie do strony ms.polsl.pl/aktualnosci.php

```
GET /aktualnosci.php?wid=6 HTTP/1.1
Host: ms.polsl.pl
Cache-Control: no-cache
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,pl;q=0.8
Cookie: _ga=GA1.1.15783512486.1496572458;
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/79.0.3945.117 Safari/537.36

```

+

Ciało zapytania (o ile istnieje)

>>> Budowa odpowiedzi

Obraz: Przykładowa odpowiedź ze strony ms.polsl.pl/aktualnosci.php

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Type: text/html; charset=ISO-8859-2
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Microsoft-IIS/10.0
X-Powered-By: PHP/5.6.31
Set-Cookie: PHPSESSID=os739h48ac9ih11c1kfsv0fjq4; path=/
X-Powered-By: ASP.NET
Date: Mon, 13 Jan 2020 08:46:09 GMT
Content-Length: 22120

```

+

Ciało odpowiedzi (o ile istnieje)

Komunikacja w HTTP/1.1

>>> Przebieg komunikacji w HTTP/1.1

1. Wysłanie żądanie do serwera DNS
2. Odpowiedź serwera DNS
3. Rozpoczęcie połączenia TCP z serwerem (wymiana pakietów SYN/ACK)
4. Wysłanie żądania HTTP do serwera
5. Odebrania danych od serwera

>>> Przebieg komunikacji w HTTP/1.1

1. Wysłanie żądanie do serwera DNS
2. Odpowiedź serwera DNS
3. Rozpoczęcie połączenia TCP z serwerem (wymiana pakietów SYN/ACK)
4. Wysłanie żądania HTTP do serwera
5. Odebrania danych od serwera

Sposób przesyłu:

* Przesyłamy żądania jako PLAINTEXT
* Nagłówki oraz ciało wysyłane są razem
* Wiele połączeń lub użycie HTTP pipelining
* Kolejkowanie zapytań

Komunikacja w HTTP/2

>>> Przebieg komunikacji w HTTP/2

1. Wysłanie żądanie do serwera DNS
2. Odpowiedź serwera DNS
3. Rozpoczęcie połączenia TCP z serwerem (wymiana pakietów SYN/ACK)
4. Ustanowienie bezpiecznego połączenia (TLS handshake)
5. Wysłanie żądania HTTP do serwera
6. Odebrania danych od serwera

>>> Przebieg komunikacji w HTTP/2

1. Wysłanie żądanie do serwera DNS
2. Odpowiedź serwera DNS
3. Rozpoczęcie połączenia TCP z serwerem (wymiana pakietów SYN/ACK)
4. Ustanowienie bezpiecznego połączenia (TLS handshake)
5. Wysłanie żądania HTTP do serwera
6. Odebrania danych od serwera

Sposób przesyłu:

* Przesyłamy żądania w formie binarnej za pomocą strumieni (NOWOŚĆ)
* Nagłówki oraz ciało wysyłane są w osobnych strumieniach (NOWOŚĆ)
* Jedno połączenie oraz priorytety (NOWOŚĆ)

Komunikacja w HTTP/3

>>> Przebieg komunikacji w HTTP/3

1. Wysłanie żądanie do serwera DNS
2. Odpowiedź serwera DNS
3. Rozpoczęcie połączenia za pomocą protokołu QUIC
4. Ustanowienie bezpiecznego połączenia (Zaimplementowane w QUIC)
5. Wysłanie żądania HTTP do serwera
6. Odebrania danych od serwera

>>> Protokół QUIC

QUIC(Quick UDP Internet Connection) jest to protokół warstwy transportowej, wymagający szyfrowania. Został stworzony w Google, aby uczynić HTTP wydajniejszym oraz bezpieczniejszym. Działa pod kontrolą UDP.

Zalety:

* Szybsze nawiązanie połączenia
* Lepsze obsługa utraty pakietów i/lub zmiany sieci
* Większe możliwości implementacji

>>> Protokół QUIC

QUIC(Quick UDP Internet Connection) jest to protokół warstwy transportowej, wymagający szyfrowania. Został stworzony w Google, aby uczynić HTTP wydajniejszym oraz bezpieczniejszym. Działa pod kontrolą UDP.

Zalety:

* Szybsze nawiązanie połączenia
* Lepsze obsługa utraty pakietów i/lub zmiany sieci
* Większe możliwości implementacji

Wady:

* Połączenie bardziej obciąża komputer
* Trudniejsze debugowanie