

# Protokół HTTP

infoShare Academy



# HELLO

## Marcin Słowikowski

DevOps Engineer  
Azure, CI/CD, Automation, Monitoring, IaC





# Agenda

- Co to jest HTTP i jakie ma zastosowania?
- HTTP w modelu sieciowym
- Żądanie, odpowiedź
- Z jakich elementów zbudowane są zapytania HTTP?
- Kody odpowiedzi
- Ciasteczka
- Sposoby przekazywania informacji do aplikacji
- Jak wziąć pod lupę HTTP?



# 01. Co to jest HTTP?

I jakie ma zastosowanie?



# HTTP – Hypertext Transfer Protocol

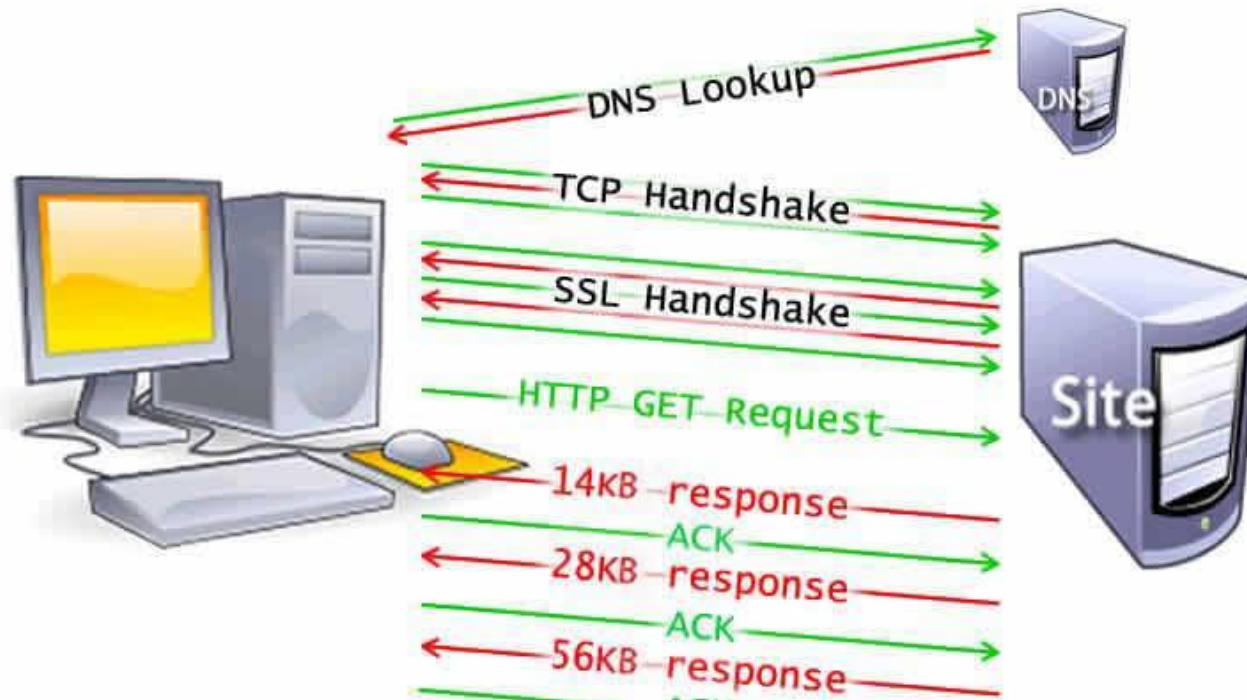
- Protokół komunikacyjny
- Architektura klient-serwer
- Opisuje formatowanie wiadomości, jak są przesyłane, oraz jakie działania powinny podejmować przeglądarki i serwery w odpowiedzi na różne polecenia
- Do najpopularniejszych zastosowań należą: strony WWW, API
- Domyślny port:
  - TCP/80 – HTTP
  - TCP/443 – HTTPS – szyfrowana wersja HTTP



HTTP działa w 7. warstwie modelu odniesienia. Wykorzystywane protokoły w niższych warstwach muszą działać poprawnie by móc korzystać z HTTP

	MODEL OSI	INTERNET	PROTOKOŁY
Dane	Aplikacji	Aplikacji	HTTP, DNS, SMTP, POP3, FTP, SSH, DNS ...
	Prezentacji		
	Sesji		
Segment	Transportowa	Transportowa	TCP, UDP, SPX
Datagram	Sieci	Sieci	IP, IPX, AppleTalk
Ramka	Łacza danych	Łacza danych	Ethernet 802.11, token ring, PPP
Bity	Fizyczna	Fizyczna	

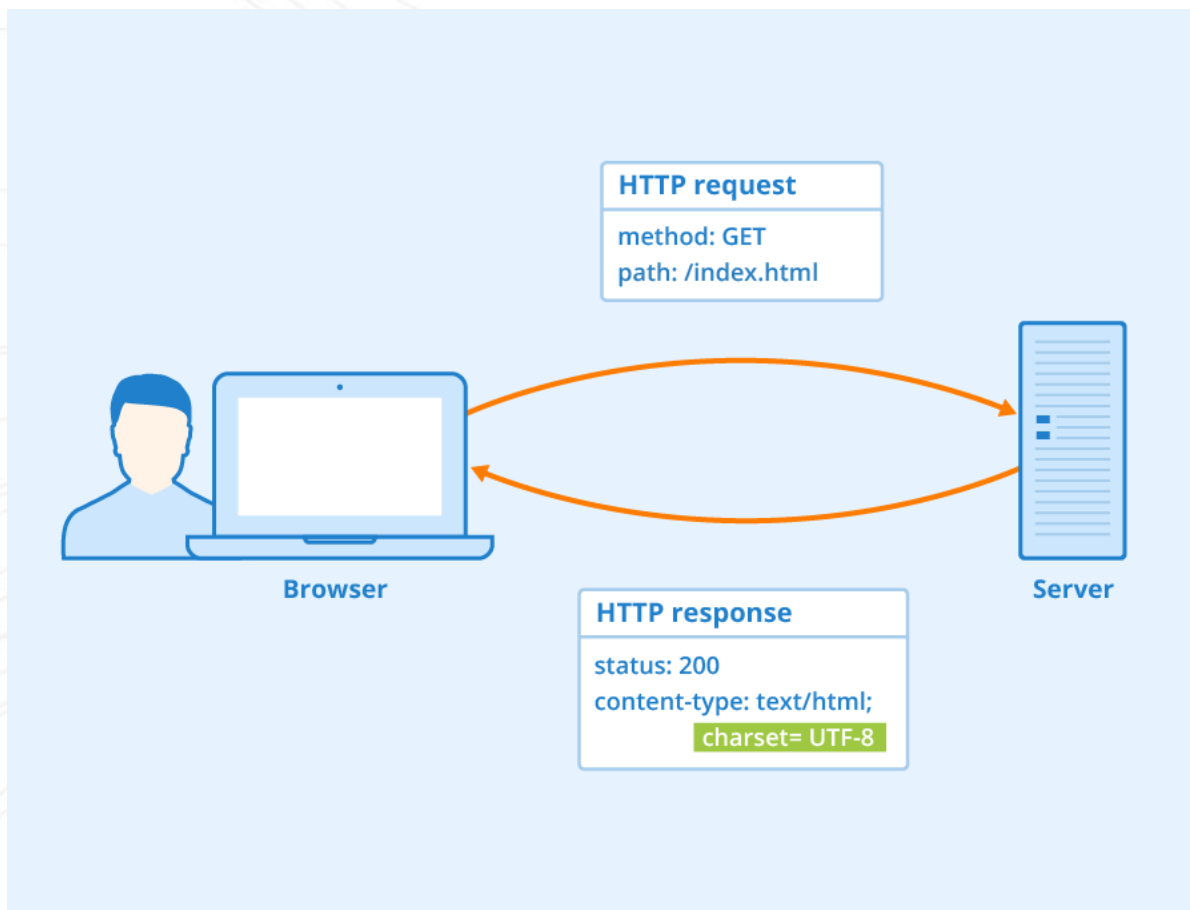
Przed rozpoczęciem komunikacji poprzez HTTP zazwyczaj wykonywane jest rozwiązanie nazwy domenowej na adres IP. Następnie wykonywane jest nawiązanie sesji TCP oraz TLS handshake



[https://developer.mozilla.org/en-US/docs/Web/Performance/How\\_browsers\\_work](https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work)

# Żądanie i odpowiedź

Przed rozpoczęciem komunikacji poprzez HTTP zazwyczaj wykonywane jest rozwiązanie nazwy domenowej na adres IP. Następnie wykonywane jest nawiązanie sesji TCP oraz TLS handshake





- **URL** – ścieżka/identyfikator zasobu
- **Metoda** – czynność wykonywana w kontekście zasobu
- **Nagłówki** (headers) – metadane definiujące przebieg komunikacji
- **Ciało** (body) – dane wysyłane oraz odbierane

```
1  POST /start/request.php HTTP/1.1
2  Host: training.local
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: pl,en-US;q=0.7,en;q=0.3
6  Accept-Encoding: gzip, deflate
7  Connection: close
8  Upgrade-Insecure-Requests: 1
9  Content-Type: application/x-www-form-urlencoded
10 Content-Length: 37
11
12  parametr1=wartosc1&parametr2=wartosc2
```



# Podstawowe metody żądań HTTP

- GET – żądanie danych
- HEAD – żądanie bez ciała odpowiedzi
- POST – przesłanie danych do serwera
- PUT – tworzy lub podmienia zasób
- OPTIONS – zwraca obsługiwane przez serwer metody
- DELETE – usuwa zasób



# Metoda GET - żądanie

- Linia żądania: metoda, URL, wersja protokołu
- Nagłówki
- Ciało żądania

```
1 GET / HTTP/1.1
2 Host: training.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: pl,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10
```



# Metoda GET - odpowiedź

- Linia statusu odpowiedzi: wersja protokołu, status
- Nagłówki
- Ciało odpowiedzi

```
1 HTTP/1.1 200 OK
2 Date: Fri, 13 Jan 2023 08:13:18 GMT
3 Server: Apache
4 Content-Length: 39
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7
8 <body>
9 <h1>trening</h1>
10 </body>
```



# Metoda POST – żądanie

- Linia żądania: metoda, URL, wersja protokołu
- Nagłówki
- Ciało żądania

```
1 POST /start/request.php HTTP/1.1
2 Host: training.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: pl,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 37
11
12 parametr1=wartosc1&parametr2=wartosc2
```





# Metoda POST – odpowiedź

- Linia statusu odpowiedzi: wersja protokołu, status
- Nagłówki
- Ciało odpowiedzi

```
1 HTTP/1.1 200 OK
2 Date: Fri, 13 Jan 2023 08:13:18 GMT
3 Server: Apache
4 Content-Length: 39
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7
8 parametr1: wartosc1
9 parametr2: wartosc2
```



# Kody odpowiedzi

- 1XX – informacja
  - 100 – prośba o dalsze wysyłanie zapytania
  - 101 – zmiana protokołu
- 2XX – sukces
  - 200 – zwrócono zawartość dokumentu
  - 201 – utworzono
  - 202 – zaakceptowano
  - 204 – brak zawartości
- 3XX – przekierowania
  - 301 – stałe przeniesienie, pozwala na zmianę metody z POST na GET
  - 308 – stałe przekierowanie
- 4XX – błąd klienta
  - 401 – nieautoryzowany dostęp
  - 404 – nie znaleziono
- 5XX – problem serwera
  - 500 – wewnętrzny błąd serwera
  - 505 – nieobsługiwana wersja HTTP

- Host – określa domenę której dotyczy żądanie
- Content-Type – typ przesyłanych danych
- Accept – typ danych jakie akceptuje klient
- Content-Length – długość przesyłanych danych
- Location – wskazuje adres przekierowania
- Referer – przekazuje adres strony z której nastąpiło przekierowanie
- Authorization – przesyła dane dotyczące autoryzacji, np. token użytkownika

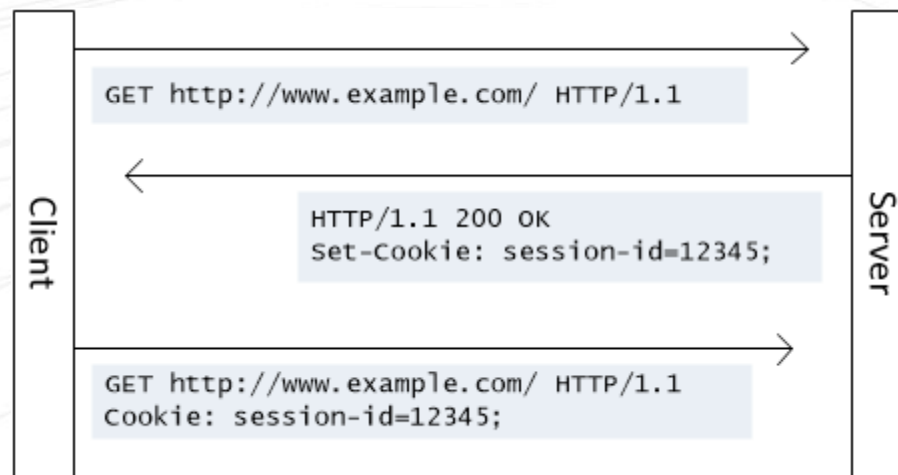
Ciasteczka to zbiór wartości klucz=wartość. Przesyłane są poprzez nagłówki

- Set-Cookie – żądanie ustawienia przez przeglądarkę ciasteczek (cookies), przesyłane przez serwer w odpowiedzi, np.:

*Set-Cookie: PHPSESSID=298zf09hf012fh2; Expires=Thu, 21 Oct 2021 07:28:00 GMT;*

- Cookie – ciasteczka przesyłane do serwera

*Cookie: PHPSESSID=298zf09hf012fh2; secondcookievalue=secondcookievalue*



<https://learn.microsoft.com/pl-pl/aspnet/web-api/overview/advanced/http-cookies>



# Przekazywanie zmiennych

Poza nagłówkami oraz ciasteczkami dane do aplikacji webowych mogą być przekazywane również poprzez:

- Parametry metody GET (query string), np.:

```
1 GET /questions?sort=Newest&filters=NoAnswers&edited=true HTTP/1.1
2 Host: stackoverflow.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101 Firefox/108.0
4
5
```

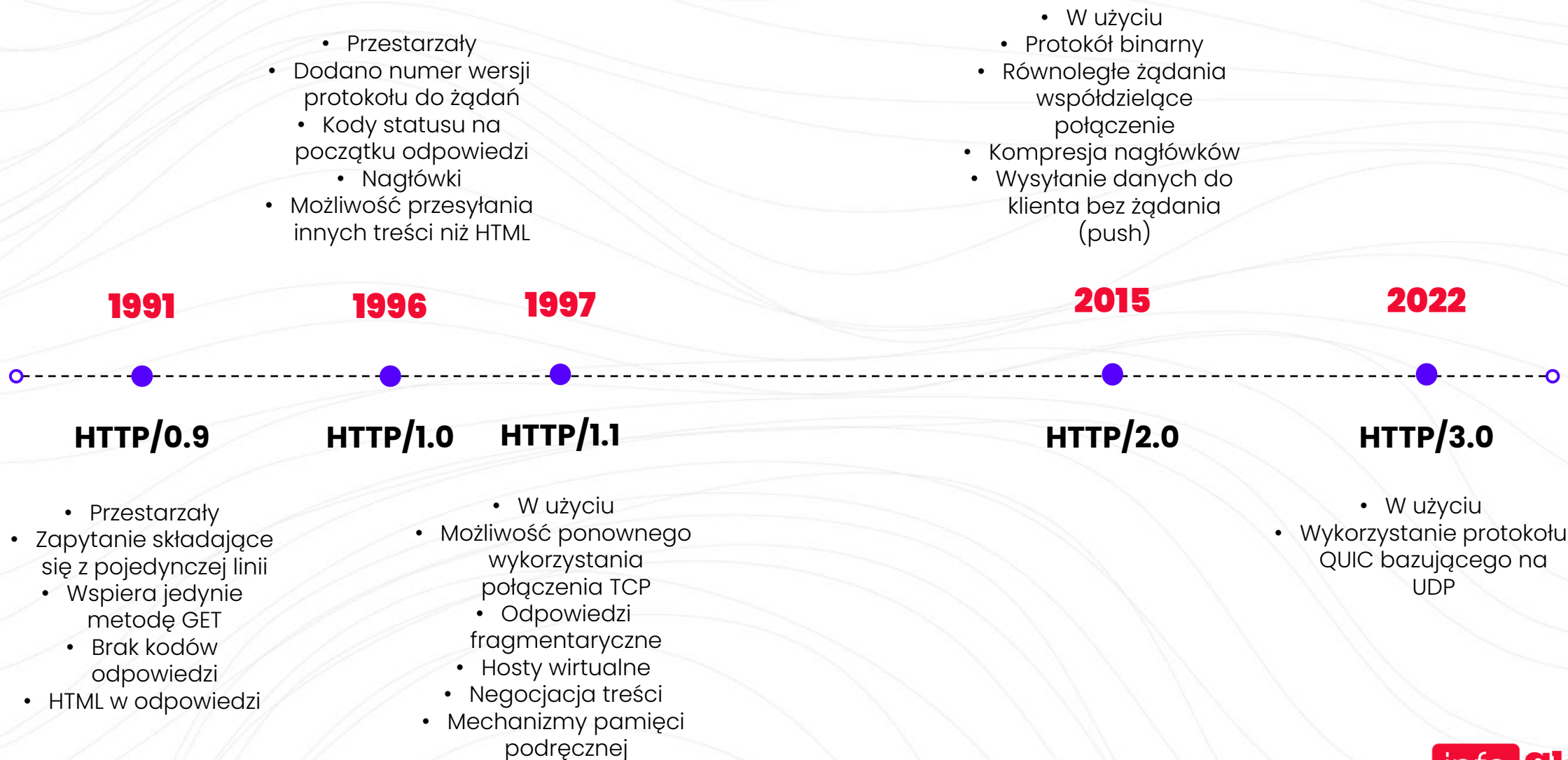
- Ciało żądania POST, np.:

```
1 POST /questions HTTP/1.1
2 Host: stackoverflow.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101 Firefox/108.0
4 Content-Type: application/x-www-form-urlencoded
5 Content-Length: 41
6
7 sort=Newest&filters=NoAnswers&edited=true
```





# Historia wersji





# HTTP pod lupą – **troubleshooting**

W celu diagnozowania oraz testowania połączeń HTTP wykorzystywane są następujące narzędzia:

- Sniffery – pełna analiza ruchu sieciowego (DNS, TLS, HTTP), np. Wireshark, tcpdump
- Narzędzia z funkcją proxy – analiza i modyfikacja ruchu HTTP, np. Burp Suite, OWASP ZAP
- DevTools w przeglądarkach – głównie analiza ruchu oraz działania aplikacji po stronie klienta





# DZIĘKI

You can find me on LinkedIn

[infoShareAcademy.com](https://infoShareAcademy.com)