PWr
Programowanie systemów webowych
Laboratorium – **Lista 1**
Oprac. D. Konieczny

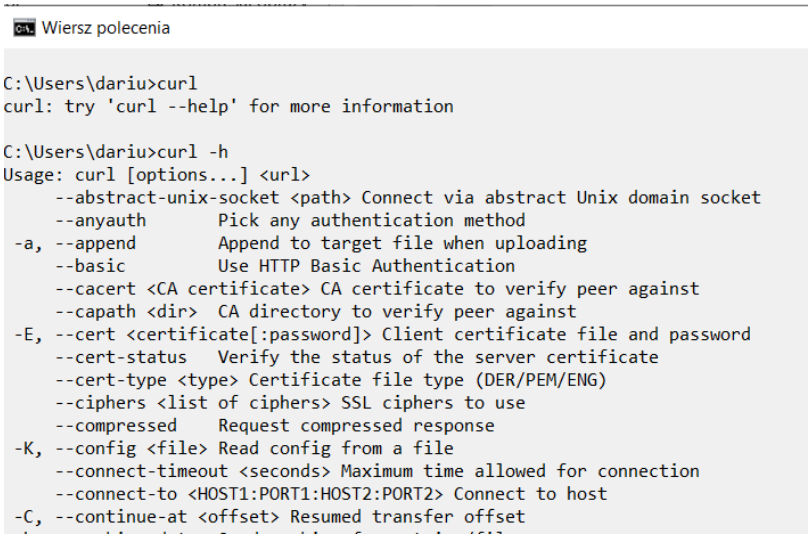
Wstęp.

Poznanie narzędzi do obserwacji działania protokołu HTTP.

Komenda `curl`

Jedną z możliwości obserwacji danych przesyłanych w protokole HTTP jest komenda `curl`. Pozwala na wysyłanie dowolnych żądań (domyślnie rodzaju GET, ale parametrami można to zmienić). Celem poniższych zadań jest poznanie tego konsolowego narzędzia.

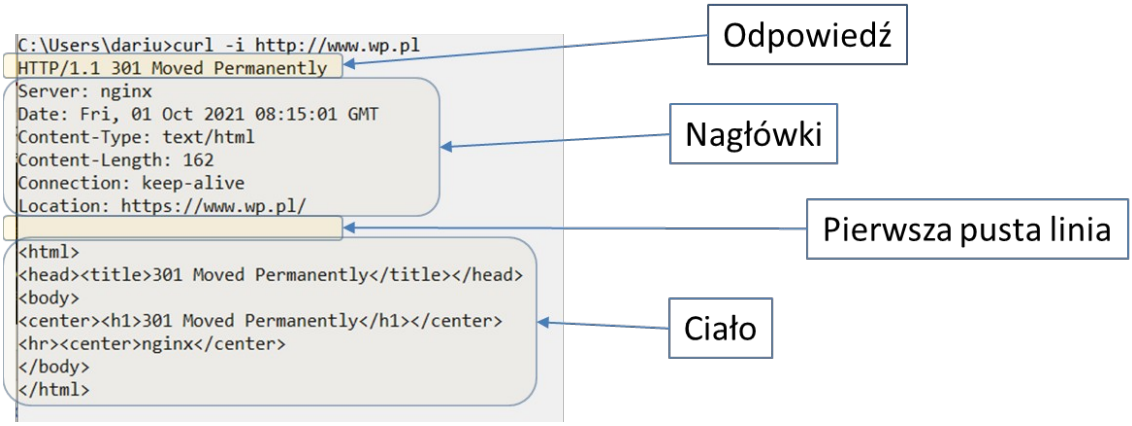
Przykład uruchamiania celem sprawdzenia opcji:



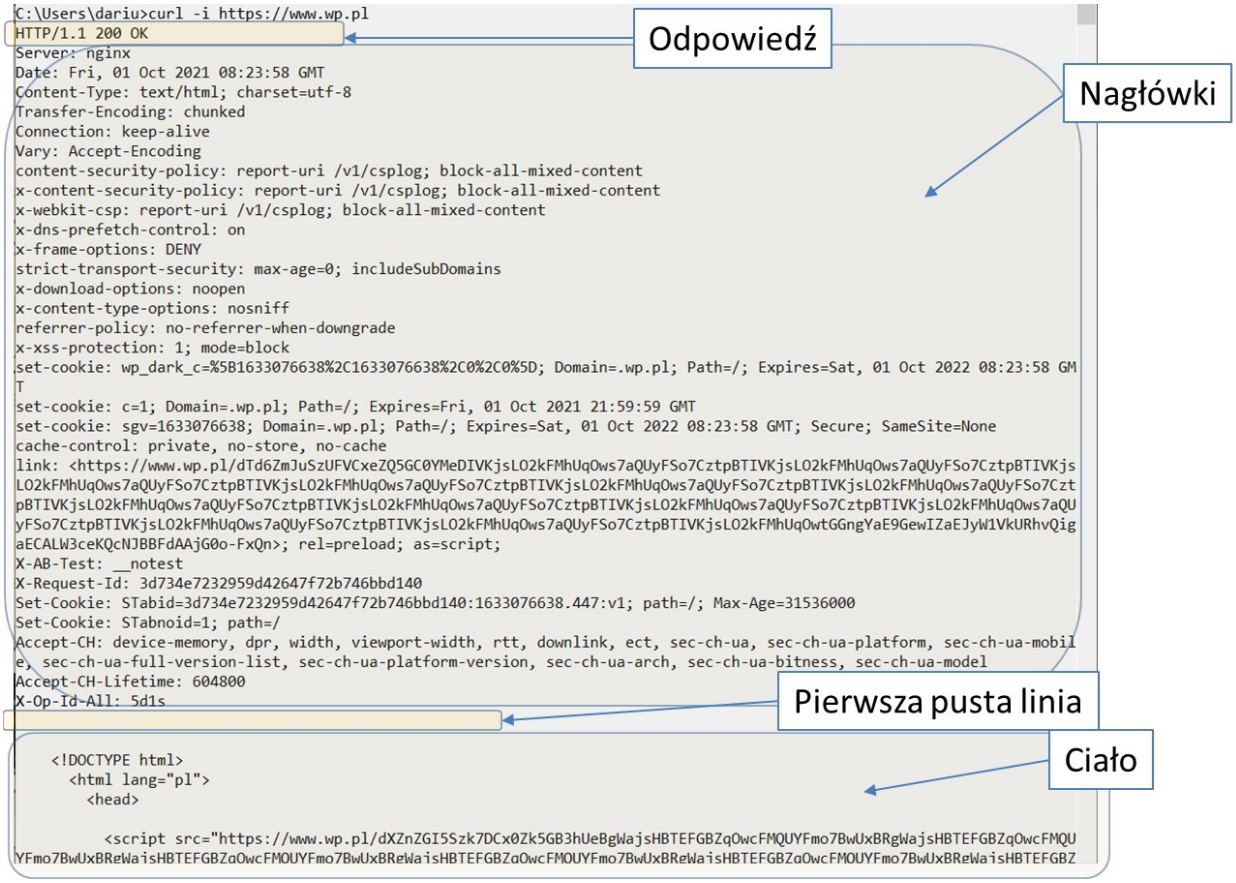
Uruchomienie z wybranym adresem URL (`http://www.wp.pl`) i otrzymana odpowiedź:

```
C:\Users\dariu>curl http://www.wp.pl
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

W wielu przypadkach warto użyć opcji `-i` (Include protocol response headers in the output) pokazującej nagłówek odpowiedzi, który może zawierać istotne informacje. Przykład użycia:

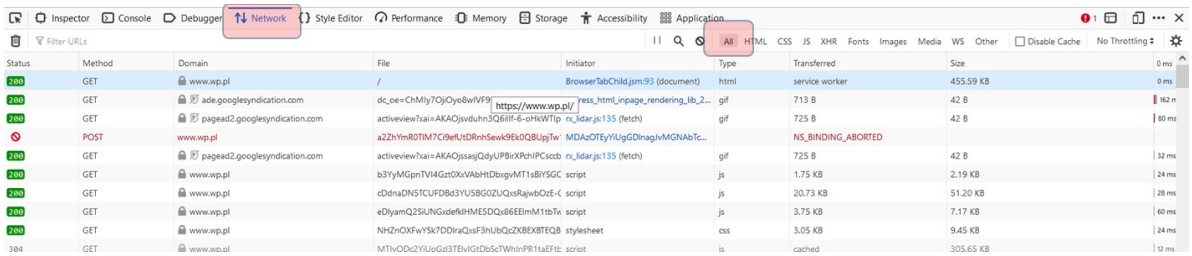


Można z niego wyczytać, że strona została na stałe przeniesiona po inny adres (`https://www.wp.pl`). Zatem po użyciu nowego adresu otrzymujemy dostęp do strumienia dla właściwej strony:

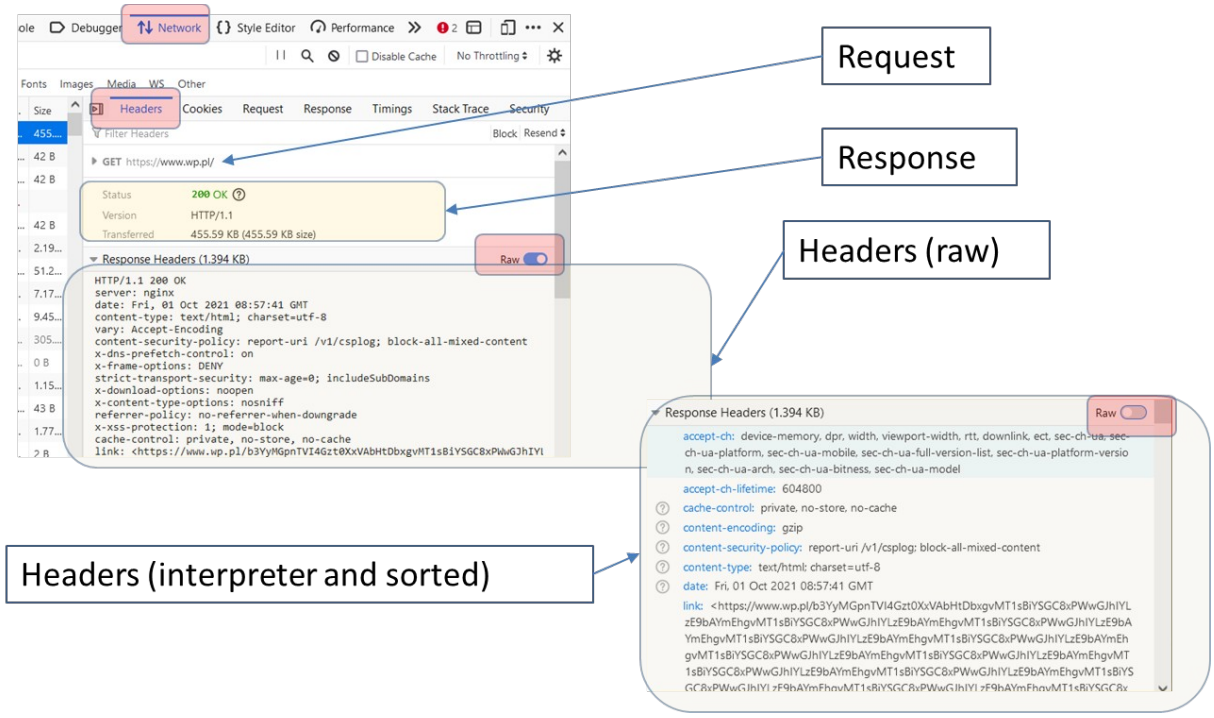


Narzędzia programistyczne przeglądarki

Dużo wygodniejszym sposobem obserwacji działania protokołu http są gotowe narzędzia programistyczne obecne w wiodących przeglądarkach internetowych. Po ich uruchomieniu (najczęściej poprzez F12 lub Ctrl+Shift+I) pojawia się okno z wieloma zakładkami zawierającymi przetworzone informacje z wysłanych żądań i otrzymanych odpowiedzi. Przykładowo strony używanej w komendzie `curl`:



Wybierając zakładkę Network (sieć) otrzymujemy tabelarycznie przedstawioną komunikację przeglądarki z serwerem, również tą spowodowaną przekierowaniem strony, ściąganiem obrazków występujących na stronie, skryptów języka JavaScript itd. Gdy wybierzemy podgląd jednego z takich żądań, np. `https://www.wp.pl` otrzymamy widok przedstawiający dane, które mogliśmy odczytać w narzędziu konsolowym, przedstawione w bardziej przejrzystej formie (trochę zależnej od przeglądarki):



Linia odpowiedzi jest rozbita na składowe, nagłówki można widzieć w postaci przetworzonej, co ułatwia ich analizę (lub nieprzetworzonej, gdy np. szukamy specyficznego błędu). Pewne części nagłówka i samej komunikacji są osobnych zakładkach, co np. ułatwia sprawdzanie tzw. ciasteczek. Można również analizować czasy komunikacji i wiele innych elementów protokołu `http`(s).

Dzięki takiej formie prezentacji łatwiej jest obserwować przekazywanie danych do i z serwera WWW.

W odpowiednich zakładkach, dla zapytania typu GET, można znaleźć parametry przekazane poprzez tzw. **query string**, czyli ciąg danych w odpowiednim formacie, który może być obecny w adresie URL po znaku zapytania ‘?’.

Analogicznie można znaleźć dane w ciele zapytania typu POST. W tym przypadku w zależności od wybranego formatu może być to zarówno format jak dla query string jak i format JSON, XML itd.

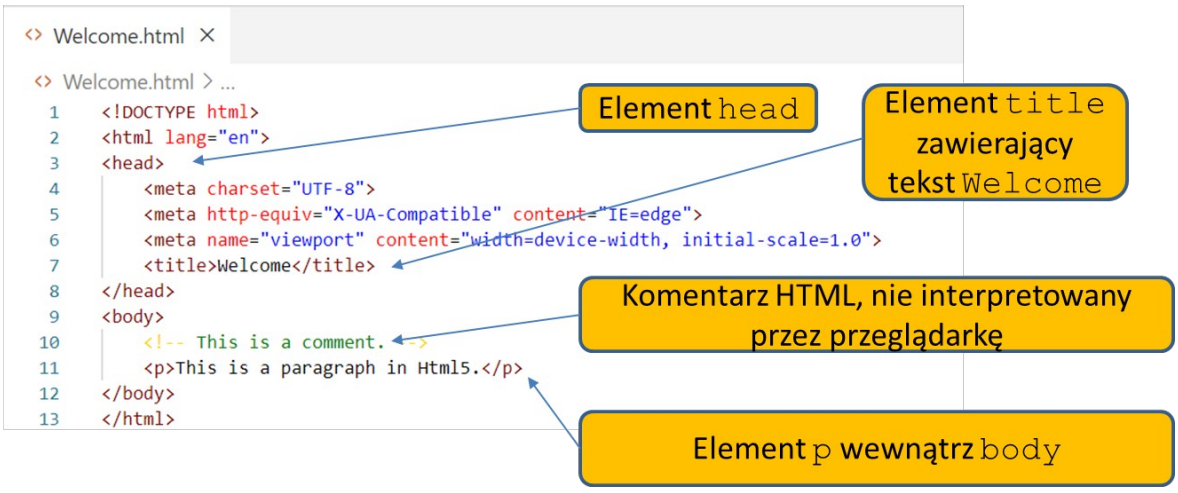
W zakładce odpowiedzi (ang. response) znajdujemy ciało odpowiedzi na zapytanie. Najczęściej jest to strona w formacie HTML, ale mogą tam być również dane w postaci JSON czy XML.

Pierwsze elementy formatu HTML

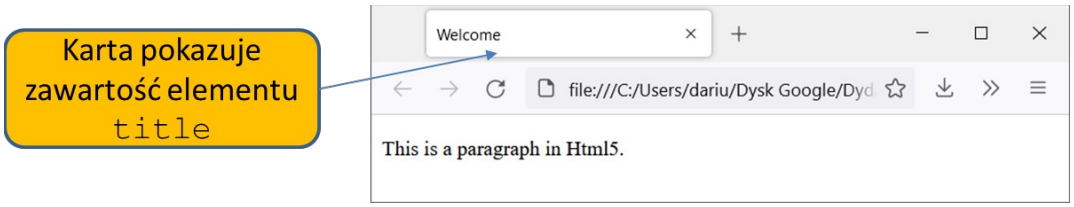
Na wykładzie przedstawione były podstawowe składowe elementy dokumenty HTML, jak:

- `<html>` - główny element dokumentu HTML
- `<head>` - nagłówek dokumentu i jego podstawowe składowe:
  - `<meta>` - element opisu dokumentu, nie prezentowane przez przeglądarkę
  - `<title>` - tytuł dokumentu
- `<body>` - ciało dokumentu, które zawiera całą pozostałą informację.

W ciele dokumentu jest bardzo duża różnorodność elementów, które albo występują jedne za drugimi albo jedne w drugich. Podstawowym jest paragraf <p> zawierający tekst pokazywany w przeglądarce. Można też wstawiać komentarze za pomocą <!--->.




```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Welcome</title>
8   </head>
9   <body>
10      <!-- This is a comment. -->
11      <p>This is a paragraph in Html5.</p>
12  </body>
13  </html>
```



Do podstawowych elementów można też zaliczyć nagłówki o różnych poziomach od <h1> do <h6> zawierające tekst nagłówka. Domyślnie posiadają one ustawione pewne wielkości czcionki od większej do coraz mniejszej.
Możliwości Internetu pokazuje dopiero możliwość przenoszenia się z jednych dokumentów do innych, co osiąga się za pomocą tzw. linków. Tak naprawdę w

dokumencie istnieją elementy kotwiczące (ang. anchor) <a>, które za pomocą atrybutu `href` pozwalają podać do jakiego dokumentu ma przejść przeglądarka po kliknięciu tekstu, który jest w środku tego elementu.
W dokumencie można również wstawiać obrazki za pomocą elementu <img> uzupełniając odpowiednim adresem atrybut `src`.

Walidacja dokumentu HTML

Oprócz poprawności elementów samych w sobie, czyli np. istnienia dla każdego znacznika otwierającego (np. <body>) znacznika zamykającego (np. </body>), zawierania się znaczników w całości w innych znacznikach, istnieją dodatkowe reguły poprawności. Jeśli je nie zastosujemy, przeglądarka będzie starała się naprawić nasz dokument podczas prezentacji na ekranie. Może się jednak okazać, że inna przeglądarka będzie prezentować to inaczej. Warto zatem pisać poprawne składniowo dokumenty. W tym celu można albo wykorzystać wtyczki do programistycznych środowisk (IDE), albo sprawdzić za pomocą walidatorów na stronach WWW, np. `validator.w3.org/#validate-by-upload`.

List zadań

1. Dla wybranej strony WWW pokaż działanie komendy `curl`. Pokaz zawartość nagłówków i ciała odpowiedzi.
2. Dla wybranej strony WWW pokaż działanie narzędzi programistycznych wybranej przeglądarki. Pokaż jakie żądania wysłała przeglądarka i jakie otrzymała odpowiedzi. Zaprezentuj rodzaj żądania, jego nagłówek i ciało. Podobnie dla odpowiedzi.
3. Znajdź stronę używającą query string w adresie URL i pokaż działanie, wpisując nowy query string z klawiatury.
4. Znajdź stronę z formularzem, która przesyła dane w ciele żądania rodzaju POST. Zaprezentuj, że rzeczywiście tak się dzieje.
5. Stwórz własną poprawną stronę (lub więcej stron) WWW, która będzie zawierać:
    a. tytuł,
    b. znaczniki dla różnych rodzajów nagłówków,
    c. tekst w paragrafach,
    d. komentarze,
    e. Podlinkowane:
        i. inne strony WWW,
        ii. obrazki,
        iii. jakiś spakowany plik,
        iv. adres email
6. Pokazać wybranym walidatorem, że strona jest poprawna.

Termin oddania: Spotkanie 2 (0 punktów)