

ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej

PWr

Spotkanie 1

*Aplikacje webowe na platformie .NET*Laboratorium – **Lista 1**


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:

 Wiersz polecenia

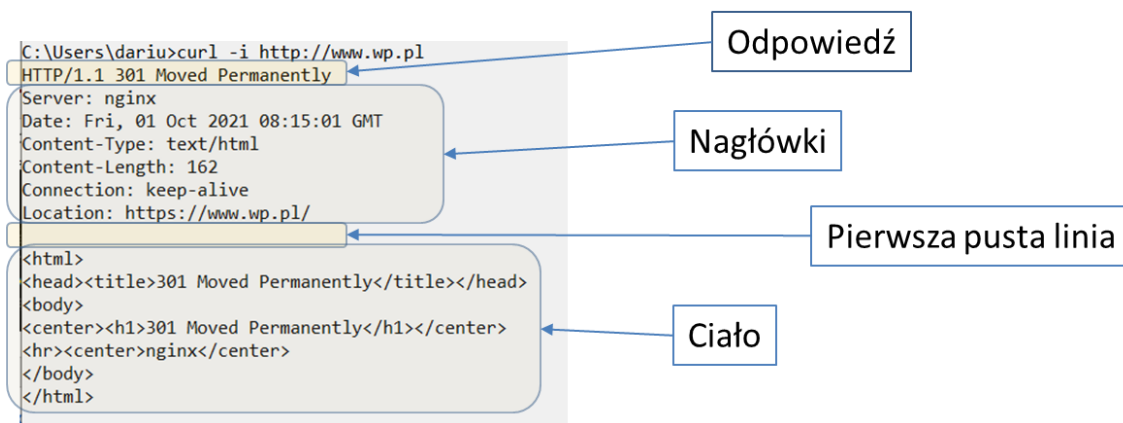
```
C:\Users\dariu>curl
curl: try 'curl --help' for more information

C:\Users\dariu>curl -h
Usage: curl [options...] <url>
  --abstract-unix-socket <path> Connect via abstract Unix domain socket
  --anyauth             Pick any authentication method
  -a, --append          Append to target file when uploading
  --basic               Use HTTP Basic Authentication
  --cacert <CA certificate> CA certificate to verify peer against
  --capath <dir>        CA directory to verify peer against
  -E, --cert <certificate[:password]> Client certificate file and password
  --cert-status         Verify the status of the server certificate
  --cert-type <type>    Certificate file type (DER/PEM/ENG)
  --ciphers <list of ciphers> SSL ciphers to use
  --compressed         Request compressed response
  -K, --config <file>   Read config from a file
  --connect-timeout <seconds> Maximum time allowed for connection
  --connect-to <HOST1:PORT1:HOST2:PORT2> Connect to host
  -C, --continue-at <offset> Resumed transfer offset
```

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 nagłówki 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:

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 jedno za drugim albo jedno w drugim. Podstawowym jest paragraf `<p>` zawierający tekst pokazywany w przeglądarce. Można też wstawiać komentarze za pomocą `<!-->`.

The diagram illustrates the structure of an HTML document. It features a code editor window titled 'Welcome.html' showing the following code:

```
<? Welcome.html X
<? Welcome.html > ...
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>
```

Annotations with arrows point to specific parts of the code:

- Element head** points to the `<head>` tag (line 3).
- Element title zawierający tekst Welcome** points to the `<title>Welcome</title>` tag (line 7).
- Komentarz HTML, nie interpretowany przez przeglądarkę** points to the `<!-- This is a comment. -->` comment (line 10).
- Element p wewnątrz body** points to the `<p>This is a paragraph in Html5.</p>` tag (line 11).

Below the code editor, a browser window titled 'Welcome' is shown. An annotation **Karta pokazuje zawartość elementu title** points to the browser's title bar, which displays 'Welcome'. The browser's address bar shows the file path `file:///C:/Users/dariu/Dysk Google/Dyd`. The main content area of the browser displays the text 'This is a paragraph in Html5.'

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 `` 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

Proszę na tej liście NIE UŻYWAĆ stylów CSS i JS, będą dopiero od trzeciej listy. Strony WWW w pewnym sensie **mają** wyglądać surowo.

1. Dla wybranej strony WWW pokaż działanie komendy `curl`. Pokaż zawartość nagłówek 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.

Data I: Spotkanie 2 (0 punktów)

Data II: Spotkanie 3 (0 punktów)

Data III: Spotkanie 4 (0 punktów)