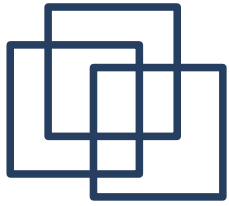




Protokół HTTP

1. Protokół HTTP,

- usługi www,
- model request-response (żądanie-odpowiedź),
- przekazywanie argumentów,
- AJAX.



Usługi WWW

WWW (*World Wide Web*) jest najpopularniejszym sposobem udostępniania informacji w sieci. Umożliwia umieszczenie w jednym dokumencie między innymi tekstu, grafiki, animacji oraz dźwięku. Standard WWW został opracowany w 1991 roku. Dwa lata później powstała pierwsza graficzna przeglądarka stron WWW – Mosaic. Strony są zapisane w języku HTML (*HyperText Markup Language*), a do ich przesyłania służy protokół HTTP (*HyperText Transfer Protocol*).

WWW opiera się na modelu klient – serwer. Strony są identyfikowane za pomocą adresów URL. Domyślny port dla usługi HTTP to 80.



Protokół HTTP

Protokół HTTP [RFC 2616]

przykładowe połączenie:

```
[klient]    HEAD / HTTP/1.1 <CRLF>
            Host: www.if.uj.edu.pl <CRLF>
            <CRLF>

[serwer]    HTTP/1.1 200 OK
            Date: Sun, 10 Oct 2004 19:54:37 GMT
            Server: Apache/1.3.27 (Unix) mod_jk/1.2.0 PHP/4.3.2
            Last-Modified: Thu, 23 Sep 2004 07:02:33 GMT
            ETag: "1357a3-294c-41527509"
            Accept-Ranges: bytes
            Content-Length: 10572
            Content-Type: text/html
```

[serwer]zamknięcie połączenia.

Komenda **HEAD** zwraca ogólne informacje na temat wskazanego dokumentu.



Protokół HTTP

W celu uzyskania treści wskazanego dokumentu należy użyć komendy **GET** lub **POST**. Za pomocą tych komend można również przekazać argumenty do serwera WWW.

Schemat wywołania:

```
[klient] GET /dokument?arg1=wart1&arg2=wart2 HTTP/1.1<CRLF>  
Host: www.if.uj.edu.pl<CRLF><CRLF>
```

lub

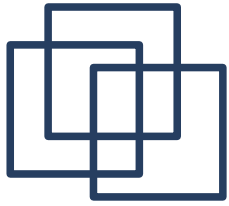
```
[klient] POST /dokument HTTP/1.1<CRLF>  
Host: www.if.uj.edu.pl<CRLF>  
Content-Type: application/x-www-form-urlencoded<CRLF>  
Content-Length: 21<CRLF><CRLF>  
arg1=wart1&arg2=wart2<CRLF>
```



Protokół HTTP

Pozostałe komendy:

- **OPTIONS** - wykaz komend dostępnych dla wskazanego dokumentu (serwera),
- **PUT** - wstawia dokument pod określony adres URL,
- **DELETE** - kasuje dokument pod określonym adresem,
- **TRACE** - zwraca informacje o poprawności zapytania. Nie zwraca, zawartości dokumentu,
- **CONNECT** - zarezerwowana na potrzeby transmisji kodowanej (SSL).



Popularne serwery WWW

- **Apache** - <http://httpd.apache.org> - darmowy serwer z udostępnionym kodem źródłowym (licencja GPL). Posiada liczne moduły umożliwiające współpracę z innymi systemami (języki programowania, bazy danych). Może działać pod kontrolą różnych systemów operacyjnych,
- **IIS** (*Internet Information Services*) - <http://www.microsoft.com/iis> - komercyjny, wspiera technologię ASP (*Active Server Pages*). Współpracuje głównie z językami programowania firmy Microsoft. Działa w środowisku MS Windows,
- **Tomcat** - <http://jakarta.apache.org/tomcat> - darmowy serwer referencyjny dla technologii JSP (*Java Server Pages*), kontener servletów. Napisany w Javie.



Konfiguracja Apache'a

Konfiguracja startowa programu serwera: `/etc/conf.d/apache2:`

```
APACHE2_OPTS="-D DEFAULT_VHOST -D PHP5"
```

Konfiguracja pracy serwera: `/etc/apache2/httpd.conf:`

```
<VirtualHost *:80>
```

```
    ServerAdmin webmaster@adres.com
```

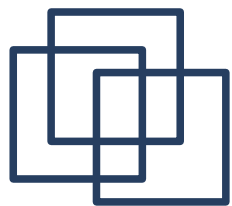
```
    DocumentRoot /var/www/docs/katalog.dla.adres.com
```

```
    ServerName adres.com
```

```
    ErrorLog /var/log/apache2/adres.com-error_log
```

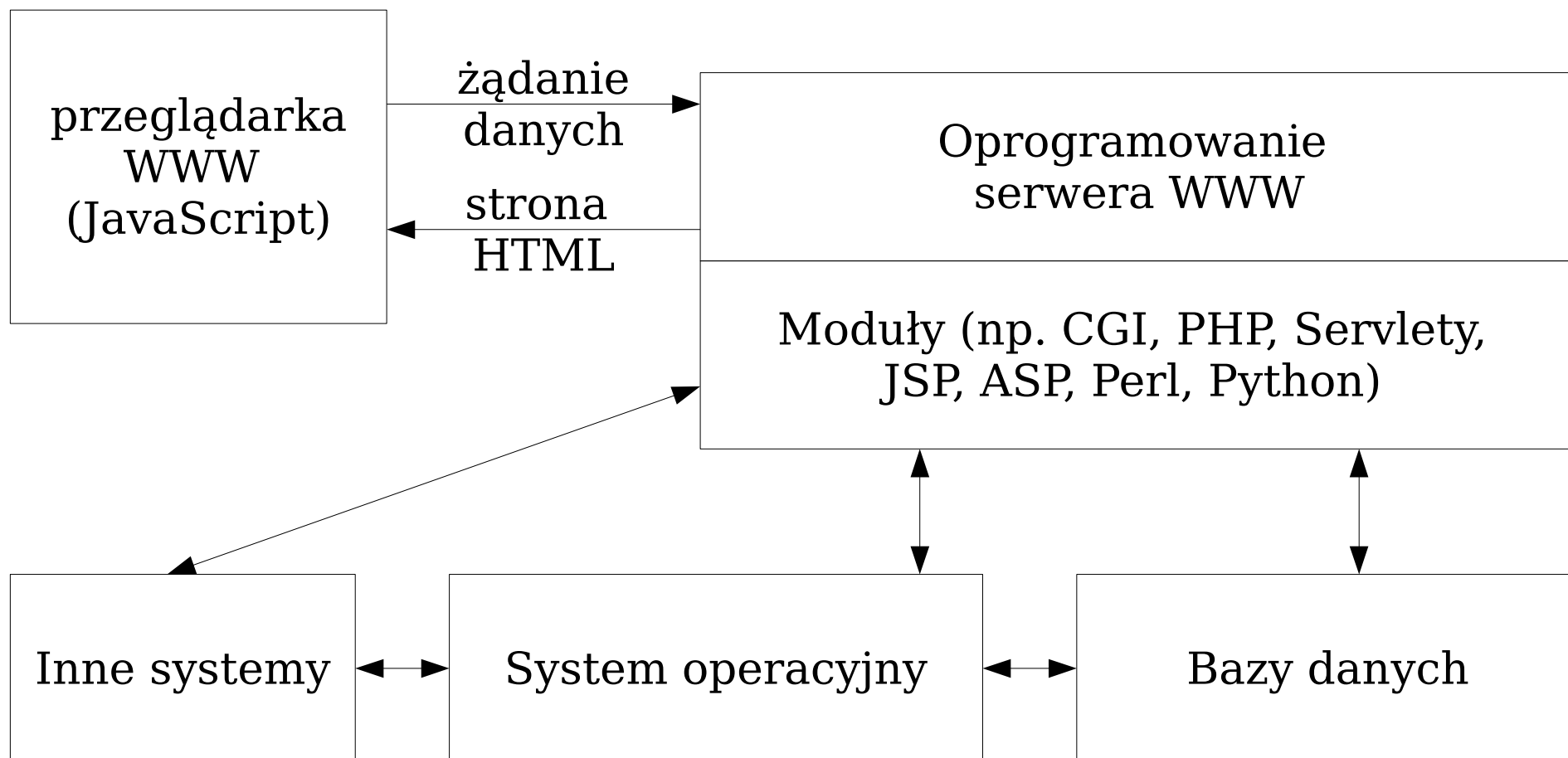
```
    CustomLog /var/log/apache2/adres.com-access_log common
```

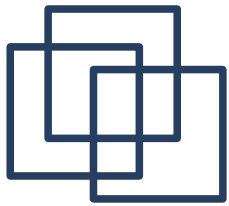
```
</VirtualHost>
```



Serwer WWW

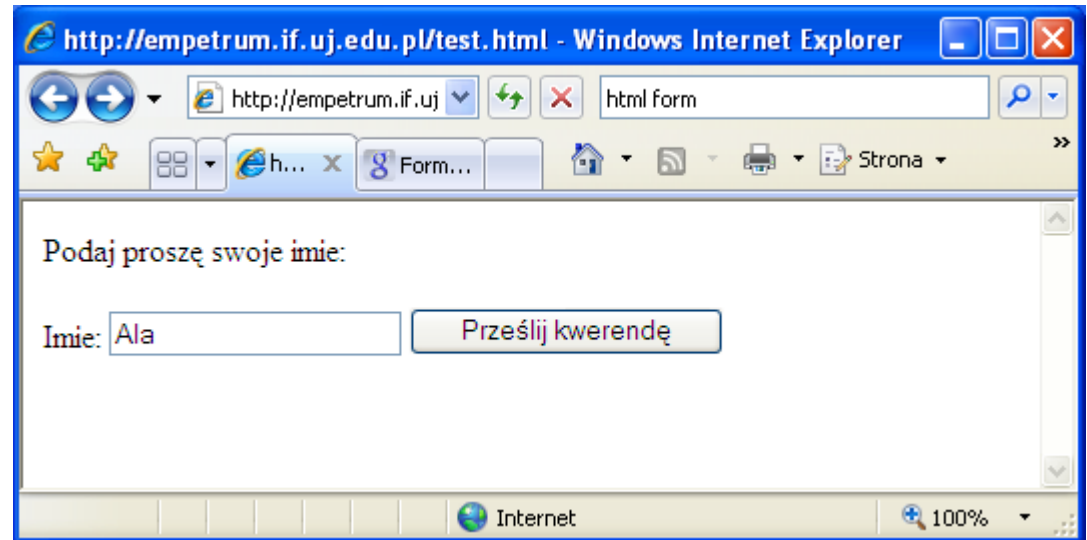
model request-response



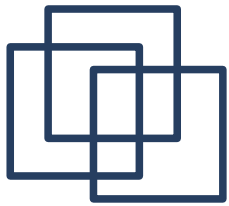


Przekazywanie argumentów

```
<html>
  <body>
    Podaj proszę swoje imie:
    <form name="formularz" method="get" action="skrypt.php">
      Imie: <input type="text" name="imie" />
      <input type="submit" />
    </form>
  </body>
</html>
```



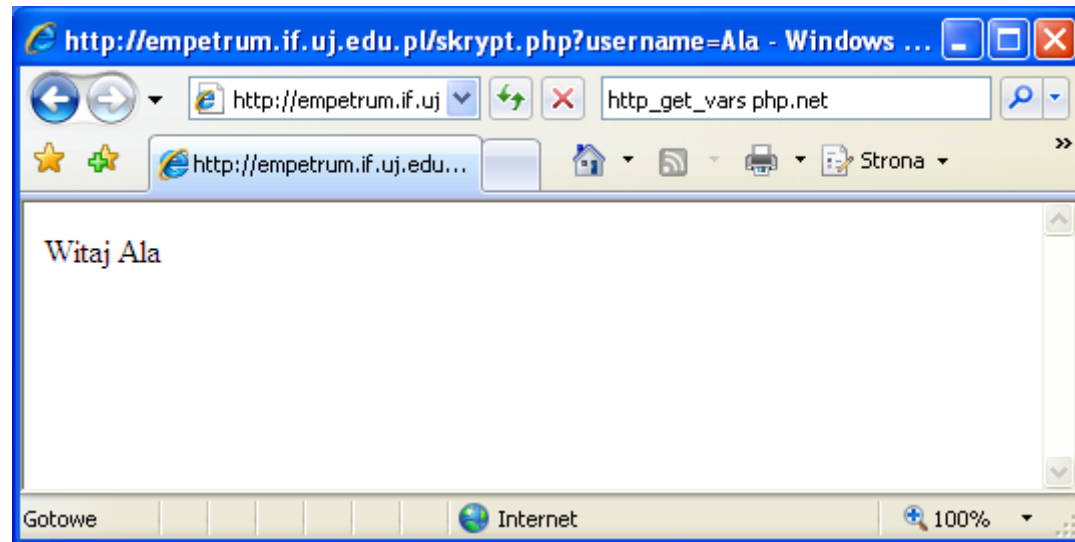
<http://empetrum.if.uj.edu.pl/skrypt.php?imie=Ala>

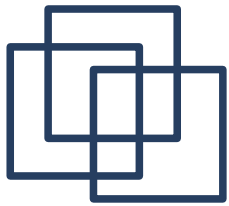


Przekazywanie argumentów

Skrypt php po stronie serwera:

```
<?php
    $s = $_GET['imie']
?>
<html>
    <body>
        Witaj <?php echo $s ?>
    </body>
</html>
```

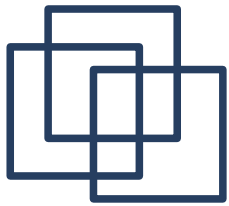




Przekazywanie argumentów

Przeglądarka i serwer mogą wymieniać się danymi:

- **GET** - argumenty wywołania skryptu/strony. Dane z przeglądarki do serwera. Przykład: `http://phpbb3.pl/viewtopic.php?f=5&t=1018`
- **POST** - Analogiczne, z tym że argumenty nie są przesyłane poprzez adres URL.
- **COOKIE** - argumenty przesłane od serwera do przeglądarki w celu późniejszej identyfikacji klienta.
- **SESSION** - argumenty związane z klientem trzymane po stronie serwera oznaczone identyfikatorem sesji przekazywanym pomiędzy serwerem i klientem.



Cookies (ciasteczka)

Serwer → przeglądarka:

```
HTTP/1.1 302 Found
```

```
Date: Sat, 25 Oct 2008 07:48:15 GMT
```

```
Server: Apache
```

```
...
```

```
Set-Cookie: imie=Michal; expires=Sat, 01 Nov 2008 07:48:15  
GMT; path=/; domain=.uj.pl; HttpOnly
```

```
Set-Cookie: nazwisko=Ciesla; path=/; domain=.uj.pl; HttpOnly
```

```
Location: http://www.uj.pl/index.php?st=0
```

```
...
```

Przeglądarka → serwer:

```
GET /index.php?st=0& HTTP/1.1
```

```
...
```

```
Cookie: imie=Michal; nazwisko=Ciesla;
```

```
...
```



Sesje

Przekazywanie sesji (identyfikatora sesji) jest możliwe na dwa sposoby:

1. Poprzez Cookie:

```
HTTP/1.1 302 Found
```

```
Date: Sat, 25 Oct 2008 07:48:15 GMT
```

```
Server: Apache
```

```
...
```

```
Set-Cookie: s=94a576da1f8e5036a0c44fce6b1e1044; expires=Sat,  
01 Nov 2008 07:48:15 GMT; path=/; domain=.uj.pl; HttpOnly
```

```
...
```

2. Poprzez dodatkowy argument GET - dodanie do każdego możliwego linka wysyłanego przez przeglądarkę członu

```
http://...&sid=94a576da1f8e5036a0c44fce6b1e1044.
```



Sesje i cookies w PHP

Ustawianie cookies:

```
bool setcookie (string $cookieName [, string $value  
                [, int $expire [, string $path  
                [, string $domain [, bool $secure  
                [, bool $httponly ]]]]]) )
```

Odczyt cookies:

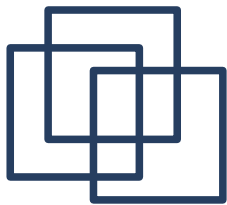
```
$_COOKIE['cookieName'] lub $_REQUEST['cookieName'].
```

Sesje:

```
bool session_start()
```

Przykład:

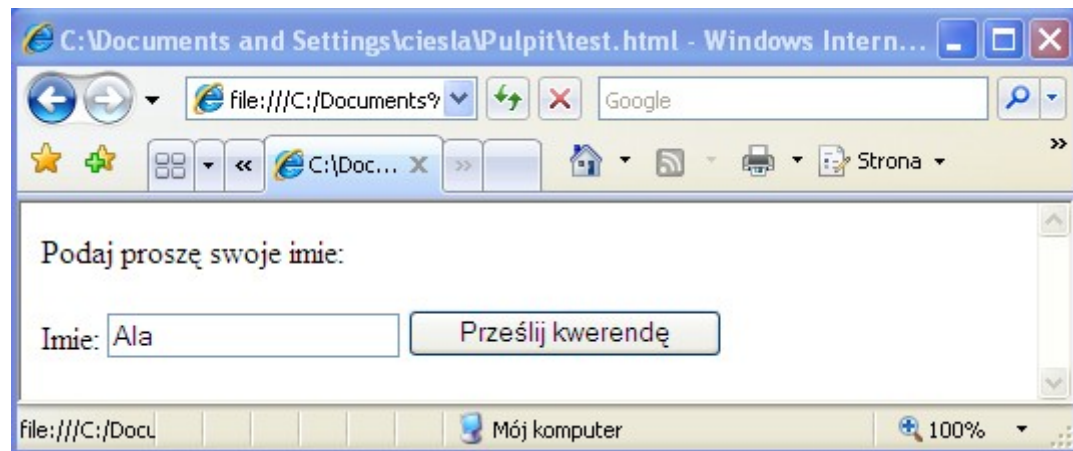
```
session_start();  
echo 'Witaj na stronie $_SESSION['strona']<br />';  
$_SESSION['strona']; = 'inna strona';
```



JavaScript

```
<SCRIPT TYPE="text/javascript">
<!--
function Test() {
    var s = document.formularz.imie.value;
    alert("Witaj " + s + "!");
}
// -->
</SCRIPT>

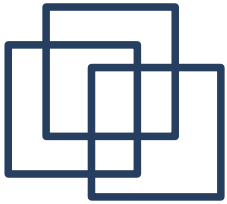
<html>
<body>
    Podaj proszę swoje imie:
    <form name="formularz" onSubmit="Test()">
        Imie: <input type="text" name="imie" />
        <input type="submit"/>
    </form>
</body>
</html>
```





AJAX

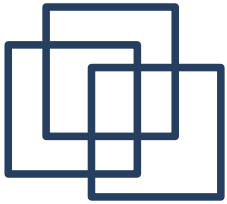
AJAX (*Asynchronous JavaScript and XML*) jest techniką pozwalającą zapewnić ciągłą (asynchroniczną) komunikację pomiędzy przeglądarką internetową a serwerem www. AJAX nie stanowi odrębnej technologii a jedynie wykorzystuje już istniejące: HTTP, JavaScript, CSS, XML



AJAX - przykład

```
<html>
<body>

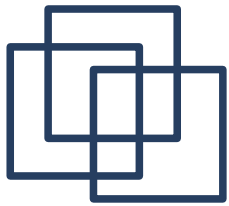
<script type="text/javascript">
function ajaxFunction(){
    var xmlhttp;
    try{ // Firefox, Opera 8.0+, Safari
        xmlhttp=new XMLHttpRequest();
    }catch (e){ // Internet Explorer
        try{
            xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
        }catch (e){
            try{
                xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
            }catch (e){
                alert("W Twojej przeglądarce nie można używać AJAXa!");
                return false;
            }
        }
    }
}
```



AJAX - przykład

```
xmlHttpRequest.onreadystatechange=function () {  
    if (xmlHttpRequest.readyState==4) {  
        document.formularz.time.value=xmlHttpRequest.responseText;  
    }  
}  
xmlHttpRequest.open ("GET", "time.php", true) ;  
xmlHttpRequest.send (null) ;  
}  
</script>
```

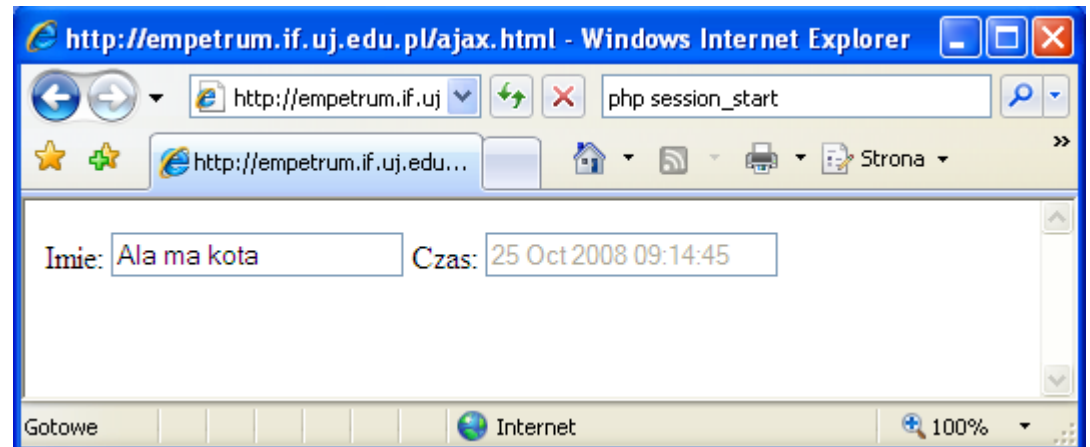
```
<form name="formularz">  
Imie: <input type="text" onkeyup="ajaxFunction();" name="imie" />  
Czas: <input type="text" name="time" disabled />  
</form></body>  
</html>
```



AJAX - skrypt time.php

```
<?php
    header("Expires: Sat, 1 Jan 2000 00:00:00 GMT");
    header("Last-Modified: ".gmdate("D, d M Y H:i:s")."GMT");
    header("Cache-Control: no-cache, must-revalidate");
    header("Pragma: no-cache");
    echo date("d M Y H:i:s");
?>
```

Czas będzie aktualizowany przy każdym wciśnięciu klawisza w polu „Imie”.



Strona nie będzie przeładowywana.

Przydatne linki:

<http://www.w3schools.com/Ajax/default.asp>

<http://www.php.net>

<http://httpd.apache.org>



Podsumowanie

1. Protokół FTP jest wykorzystywany do przesyłania plików w sieci internet. Usługi www są udostępniane poprzez protokół HTTP działający w modelu request-response. Ze względu na postępujący rozwój i dużą ilość zastosowań opracowano metody asynchronicznej wymiany danych w ramach protokołu HTTP, pomiędzy przeglądarką internetową a serwerem www.