

Wysyłanie danych metodą POST

W przypadku metody GET przesyłanie danych odbywało się przez umieszczenie ich w adresie URL, natomiast w przypadku metody POST dane muszą być wysyłane bezpośrednio do serwera. Należy w tym celu użyć argumentu metody send obiektu XMLHttpRequest. Dodatkowo niezbędne jest również wysyłanie nagłówka http informującego o sposobie kodowania danych. Najlepiej użyć tego samego rodzaju kodowania jak w przypadku metody GET, a zatem należy wysłać nagłówek Content-Type o zawartości application/x-www-form-urlencoded. Tę czynność wykonuje się za pomocą metody setRequestHeader. Dzięki takiemu samemu kodowaniu, jak w przypadku metody GET, parametry też będą przekazywane w postaci:
parametr1=wartość1¶metr2=wartość2&...=¶metrN=wartość

Przykład1

Napisz kod wysyłający pewien parametr za pomocą metody POST do umieszczonego na serwerze skryptu p1.php i odbierający odpowiedź ze skryptu. Odpowiedzią skryptu powinna być otrzymana przez niego wartość.

```
<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function przetwarzajDane(dane)
{
    if (XMLHttpRequestObject)
    {
        var div=document.getElementById("div1");

        XMLHttpRequestObject.open("POST","p1.php");
        XMLHttpRequestObject.setRequestHeader('Content-Type','application/x-www-form-urlencoded');

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {
                var str="dane otrzymane z serwera ";
                str+=XMLHttpRequestObject.responseText;
                div.innerHTML=str;
            }
        }

        XMLHttpRequestObject.send("tekst="+dane);
    }
}
```

określenie metody kodowania –
wysyłanie metody
setRequestHeader

przekazanie parametru tekst o
wartości jaką posiada
zmienna dane

```

}

</script>

</head>

<body>

<div>
<input type="button" value="kliknij" onclick="przetwarzajDane('ajax');" />
</div>

<div id="div1">
ten test zostanie zmieniony
</div>

</body>
</html>

```

p1.php

```

<?

if(isset($_POST['tekst']))
{
echo $_POST['tekst'];
}
else
{
echo "brak parametru";
}

?>

```

Przykład 2

Umieść na stronie WWW pole tekstowe oraz przycisk. Po kliknięciu przycisku wartość wprowadzona do pola tekstowego ma zostać przesłana do skryptu PHP p2.php. W odpowiedzi skrypt powinien zwrócić informację o tym, jaki jest wynik pomnożenia otrzymanej liczby przez 2.

```

<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

```

```

function przetwarzajDane(src)
{
    if (XMLHttpRequestObject)
    {
        var div=document.getElementById("div1");

        XMLHttpRequestObject.open("POST","p2.php");
        XMLHttpRequestObject.setRequestHeader('Content-Type','application/x-www-form-urlencoded');

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {
                var str=XMLHttpRequestObject.responseText;
                div.innerHTML=str;
            }

            var dane=document.getElementById(src).value;
            dane=escape(dane);
            XMLHttpRequestObject.send("liczba="+dane);
        }
    }

</script>

</head>

<body>

<div>
<input type=text id=tekst >
<input type="button" value="kliknij" onclick="przetwarzajDane('tekst');" />
</div>

<div id="div1">
ten test zostanie zmieniony
</div>

</body>
</html>

```

p2.php

```

<?
if(isset($_POST['liczba']))
{
    $liczba=$_POST['liczba'];
    if ($liczba == "")
        echo "nic nie wpisales w pole";
}

```

```

else
    if (is_numeric($liczba))
    {
        $wynik = $liczba*2;
        echo "$liczba*2=$wynik";
    }
    else
        echo "podana wartosc to nie liczba!";
}
else
{
echo "brak parametru";
}

```

Przykład 3 - samodzielnie

Napisz skrypty wyposażające witryny w hasło dostępu, tak aby weryfikacja danych nie powodowała konieczności przeładowania witryny.

Przykład 4.

Napisz skrypt wyposażający witrynę w system logowania wymagający podania nazwy i hasła użytkownika. Weryfikacja danych ma być przeprowadzana w tle, bez konieczności przeładowania strony.

```

<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function wylaczKontrolki()
{
    document.getElementById('tekst').disabled=true;
    document.getElementById('haslo').disabled=true;
    document.getElementById('przycisk').disabled=true;
}

function przetwarzajDane(srcl,srch)
{
    if (XMLHttpRequestObject)
    {
        var div=document.getElementById("div1");
        XMLHttpRequestObject.open("POST","loguj.php");
        XMLHttpRequestObject.setRequestHeader('Content-Type','application/x-www-form-urlencoded');

        XMLHttpRequestObject.onreadystatechange = function()
        {

```

wyłączone pola tekstowe i przycisk

zmienna, która przechowuje identyfikator pola tekstowego od hasła

zmienna, która przechowuje identyfikator pola tekstowego od loginu

```

200)    if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
        {
            var str=XMLHttpRequestObject.responseText;
            if (str == 'false')
                div.innerHTML="logowanie niepoprawne!";
            else
            {
                wylaczKontrolki();
                div.innerHTML=str;
            }
        }
    }

```

Odpowiedź ze skryptu loguj.php w postaci „false”, gdy logowanie nie jest poprawne lub w postaci tekstu „logowanie poprawne. Witaj na stronie”

jeżeli logowanie jest poprawne, to wywoływana jest funkcja wylaczKontrolki()

```

    var danel=document.getElementById(srcl).value;
    var daneh=document.getElementById(srch).value;
    danel=escape(danel);
    daneh=escape(daneh);
    var parametry="login="+danel;
    parametry+="&haslo="+daneh;
    XMLHttpRequestObject.send(parametry);
}

```

```

</script>

```

```

</head>

```

```

<body>

```

```

<div style="text-align:center">
Wprowadz login <input type=text id=tekst> <br>
Wprowadz haslo <input type=password id=haslo >
<br>
<input type="button" value="zaloguj" id="przycisk"
onclick="przetwarzajDane('tekst','haslo');" />
</div>

```

```

<hr>
<div id="div1" style="text-align:center">
nie jestes zalogowany
</div>

```

kliknięcie przycisku wywołuje funkcję przetwarzajDane, która przekazuje dwa parametry-identyfikatory pól tekstowych

```

</body>

```

```

</html>

```

loguj.php

```
<?

function sprawdzDane($login,$haslo)
{
    $uzytkownicy=array
    (
        'user1' => 'pass1',
        'user2' => 'pass2',
        'user3' => 'pass3'
    );
    if(array_key_exists($login,$uzytkownicy))
        if ($uzytkownicy[$login]==$haslo)
            return true;
        else
            return false;
    else
        return false;
}

if(isset($_POST['login']) && isset($_POST['haslo']))
{
    if (sprawdzDane($_POST['login'],$_POST['haslo']))
        echo "logowanie poprawne witaj na stronie";
    else
        echo "false";
}
else
{
    echo "false";
}
```

Tablica \$uzytkownicy, nazwy użytkowników to indeksy, natomiast hasła to wielkość indeksów.

[pass1,pass2,pass3]

\$uzytkownicy[user1]=pass1

Przykład 5

Napisz zestaw skryptów generujących prostą galerię obrazów, tak aby zmiana obrazu nie powodowała konieczności przeładowania strony, a dodawanie i usuwanie obrazów nie wymagało modyfikacji kodu witryny.

```
<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}
```

Identyfikator obrazu, który ma zostać wyświetlony

```
function getImgDiv(imgId,dest)
{
    if (XMLHttpRequestObject)
    {
        var div=document.getElementById(dest);
        var url="galeria.php?imgid="+imgId;
        XMLHttpRequestObject.open("GET",url);

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {
                div.innerHTML=XMLHttpRequestObject.responseText;
            }
        }

        XMLHttpRequestObject.send(null);
    }
}
```

Identyfikator warstwy, na której ma się obraz pojawić

</script>

</head>

identyfikator warstwy

<body>

<div id="divImg">
</div>

<script type="text/javascript">
 getImgDiv(0,'divImg');
</script>

</body>
</html>

Wywołanie funkcji getImgDiv, której zadaniem jest wygenerowanie treści warstwy divImg

galeria.php

```
<?
$imgDir="./obrazy";

if (isset($_GET['imgid']))
{
    $imgId=$_GET['imgid'];
}
else
{
    $imgId=0;
}

$dir=scandir($imgDir);
array_shift($dir);
array_shift($dir);

$count=count($dir);

if ($imgId<0 || $imgId>=$count || !is_numeric($imgId))
{
    $imgId=0;
}

$imgName=$dir["$imgId"];
$first=0;
$last=$count-1;

if ($imgId<$count-1)
{
    $next=$imgId+1;
}
else
{
    $next=$count-1;
}

if ($imgId>0)
{
    $prev=$imgId-1;
}
else
{
    $prev=0;
}

?>
```

Zmienna \$imgDir wskazuje na katalog, w którym są obrazy

Zmiennej \$dir przypisana jest zawartość katalogu wskazanego przez \$imgDir. Wynik działania funkcji scandir, to tablica, która zawiera wszystkie elementy katalogu posortowane alfabetycznie. Pierwsze dwa to (.)-katalog bieżący i (..)-katalog nadrzędny, chyba że wskazujemy katalog główny. Array_shift usuwa elementy z tablicy, wywołujemy dwa razy, żeby usunąć (.) i (..)

count zlicza elementy tablicy i wynik zapisuje pod zmienną \$count

Pobranie z tablicy nazwy pliku graficznego

\$first, \$last, \$prev, \$next to identyfikatory obrazu. Ponieważ numerujemy od zera (\$first=0), a nie od 1, dlatego \$last=\$count-1