

AJAX

Ajax (ang. *Asynchronous JavaScript and XML* — *Asynchroniczny JavaScript i XML*)

Technika tworzenia aplikacji internetowych, w której interakcja użytkownika z serwerem odbywa się bez przeładowywania całego dokumentu. Strona wykorzystująca Ajax jest zwykłym dokumentem HTML/CSS zawierającym skrypty JavaScript. Ajax nie wprowadza żadnych nowych języków. Interakcje użytkownika (np. kliknięcie ikony, wskazanie elementu kursorem myszki) jest realizowane poprzez zdarzenia zdefiniowane w specyfikacji HTML (m.in. **onclick**, **onmouseover**, **onmouseout**).

Tworzenie połączenia

Praktyczne poznawanie Ajax-a rozpoczniemy od tworzenia obiektu do komunikacji asynchronicznej. W przeglądarkach **Firefox**, **Opera** oraz **Internet Explorer 7** obiekt taki tworzymy następująco:

```
function getXMLHttpRequest()
{
    var request = false;

    try {
        request = new XMLHttpRequest();
    } catch(err1) {
        try {
            request = new ActiveXObject('Msxml2.XMLHTTP');
        } catch(err2) {
            try {
                request = new ActiveXObject('Microsoft.XMLHTTP');
            } catch(err3) {
                request = false;
            }
        }
    }
    return request;
}
```

Kod ten umieszczamy na stronie

```
<head>

    <title>...</title>

    <script type="text/javascript">

        function getXMLHttpRequest()

        {

            ...

        }

    </script>
```

```
var r;

r = getXMLHttpRequest();

</script>

</head>

<body>

...

</body>
```

Wysyłanie żądań HTTP i odbieranie danych z serwera

Witryna wykonana w technologii Ajax składa się z dwóch komponentów: dokumentu HTML oraz danych udostępnianych przez serwer WWW. Przeglądarka w odpowiedzi na interakcje użytkownika (np. kliknięcie ikony +) wysyła zapytanie HTTP do serwera. W odpowiedzi, serwer przekazuje do strony WWW (dokładniej: do skryptu JavaScript zawartego w dokumencie HTML) dane.

Do wysyłania żądań HTTP obiekt **XMLHttpRequest** ma metody **open()** oraz **send()**. Metoda **open()** przygotowuje zapytanie HTTP, a **send()** rozpoczyna transmisję.

Funkcja **open()** ma trzy parametry: pierwszym jest nazwa metody protokołu HTTP, drugim — adres URL danych, zaś trzecim — flaga logiczna ustalająca, czy żądanie ma być realizowane asynchronicznie (tj. w tle, bez czekania na zakończenie).

```
r.open('GET', 'dane.xml', true);
r.open('GET', 'skrypt.php', true);
r.open('GET', 'strona.asp', true);
r.open('GET', 'witryna.jsp', true);
r.open('GET', 'plik.jpg', true);
```

Można też pobrać konkretną stronę

```
r.open('GET', 'http://www.example.net/d/get.php?id=123', true);
```

Żądanie przygotowane metodą **open()** wysyłamy wywołując metodę **send()**. Metoda **send()** ma jeden parametr: dane dołączane do zapytania. Parametr ten należy wykorzystać w przypadku metody **POST**. Jeśli stosowaną metodą jest **GET**, wówczas metodę **send()** wywołujemy podając parametr **null**.

```
r.send(null);
```

W celu odebrania wyników zwracanych przez obiekt **XMLHttpRequest** należy przygotować funkcję, która zostanie wywołana po zakończeniu transmisji. Funkcja ta może mieć dowolną nazwę, np. **processResponse()**. Należy ją przypisać do obsługi zdarzenia **onreadystatechange** obiektu **XMLHttpRequest**:

```
r.onreadystatechange = processResponse;
```

W treści funkcji sprawdzamy czy nadeszła odpowiedź na wysłane żądanie oraz czy żądanie HTTP zostało poprawnie przetworzone przez serwer. Dane można odebrać w formacie tekstowym.

```
<head>
...
<script type="text/javascript">
function getXMLHttpRequest()
{
    ...
}

var r;
r = getXMLHttpRequest();

function processResponse()
{
    if (r.readyState == 4) {
        if (r.status == 200) {
            alert('Tekst z serwera: ' + r.responseText);
        };
    };
}

r.open('GET', 'dane.txt', true);
r.onreadystatechange = processResponse;
r.send(null);
</script>
</head>
```

Odbieranie danych w formacie XML

XML jest podobny do HTML, z tą różnicą, że znaczniki mogą mieć dowolną nazwę np:

```
<?xml version="1.0" encoding="utf-8"?>

<piosenka>

    <tytul>Krasnoludki</tytul>

    <tekst>

        My jesteśmy krasnoludki,

        Hopsa sa, hopsa sa!

        ...

    </tekst>

</piosenka>
```

Dane XML odebrane z serwera należy przed wyświetleniem przetworzyć. Dostęp do danych w formacie XML zapewnia właściwość `responseXML` obiektu `XMLHttpRequest`. Do przetworzenia kodu XML służą m.in.: metoda `getElementsByTagName()` i właściwości `childNodes` oraz `nodeValue`. Wewnątrz funkcji `processResponse()` najpierw odbieramy XML zwrócony przez serwer :

```
function processResponse()
{
    if (r.readyState == 4) {
        if (r.status == 200) {
            alert('XML z serwera: ' +
r.responseXML.getElementsByTagName('tekst')[0].childNodes[0].nodeValue);

        };
    }
}
```

Znaleziony zostanie pierwszy węzeł i jego wartość pierwszego dziecka.

Zadania do wykonania:

1. Z wykorzystaniem serwera XAMPP w folderze htdocs umieść 5 plików tekstowych.
2. Przygotuj stronę HTML z umieszczonym kodem JS+Ajax, która będzie posiadała pięć przycisków z nazwami plików. W momencie najechania na któryś z nich do elementu DIV na stronie ma się wczytać zawartość wybranego pliku. Do wyszukania DIVa warto użyć `getElementById` lub `jQuery`.
3. Przetestuj działanie wczytywania danych w formacie XML wg podanego wyżej przykładu. Niech i ten kod zostanie uruchomiony poprzez przycisk.
4. Przygotuj plik XML z danymi dotyczącymi 5 książek: tytuł, autor, liczba stron, opis. Za pomocą listy UL-LI lub podobnego mechanizmu stwórz coś w rodzaju zakładek tak, aby po naciśnięciu wczytać dane do kolejnego DIVa na stronie.
5. Jako, że możemy wysyłać dane metodą GET na stronę, stwórz plik PHP, który przyjmie żądanie z pliku HTML. Z HTMLa wysyłamy numer z formularza. Niech PHP zwróci po prostu tekst „otrzymałem liczbę X”. Niech Ajax wyświetli ten napis.
6. Tym razem przygotuj plik PHP generujący dane w formacie XML dla panelu newsów podobnego wizualnie jak na <http://pclab.pl/> w sekcji Aktualności na stronie głównej. W panelu mają się wyświetlać newsy dla 1 z 5 kategorii. A zatem niech istnieje panel zakładek dla newsów o danej kategorii. W panelu tym mają się pojawiać miniatury zdjęć (z adresu www) oraz klikalny tytuł do newsa. Sam link nie musi nigdzie przenosić. W każdej zakładce ma się wyświetlać 5 najnowszych newsów danej kategorii. Przypominam, że w ajaxie można ładować pliki php w sposób `plik.php?cat_id=1`

Całość bez zadań pochodzi ze strony: <http://gaidaw.pl/ajax/wprowadzenie/p4.html>

