

Laboratorium 2: Obiekt XMLHttpRequest

Czas realizacji: 4 godziny

Pliki: data.xml

Wszystkie nowoczesne przeglądarki mają wbudowany na stałe obiekt XMLHttpRequest(), który służy do wysyłania i odbierania danych z serwera.

Dodatkowo mają też również wbudowany XML parser do manipulowania formatem XML. Dzięki obiektowi XMLHttpRequest() można:

- Uaktualniać stronę bez przeładowania
- Wysłać zapytania na serwer już po załadowaniu strony
- Pobierać dane z serwera po załadowaniu strony
- Wysłać dane do serwera w tle, nie blokując działania strony.

Więcej na jego temat przeczytasz na wykładzie TZ_04_XML_xmlhttprequest.pdf

Ćwiczenie 1. Przeanalizuj poniższy przykład

Dokument XML:

Plik *data.xml*

```
<CATALOG>
  <CD>
    <TITLE>Sylvias Mother</TITLE>
    <ARTIST>Dr.Hook</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS</COMPANY>
    <PRICE>8.10</PRICE>
    <YEAR>1973</YEAR>
    <TAGS>
      <TAG>old</TAG>
      <TAG>classic</TAG>
    </TAGS>
  </CD>
  <CD>
    <TITLE>Maggie May</TITLE>
    <ARTIST>Rod Stewart</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Pickwick</COMPANY>
    <PRICE>8.50</PRICE>
    <YEAR>1990</YEAR>
    <TAGS>
      <TAG>old</TAG>
      <TAG>classic</TAG>
    </TAGS>
  </CD>
</CATALOG>
```

Metoda tworząca obiekt XMLHttpRequest() i wczytująca plik data.xml na stronę HTML.

Plik *index.html*, przed domknięciem `</body>` w `<script>...</script>`

```
function loadXMLDoc() {
    var xmlhttp = new XMLHttpRequest();           //definiujemy obiekt
    xmlhttp.onreadystatechange = function() {    //sprawdzamy status
        if (this.readyState == 4 && this.status == 200) {
            //jeśli wszystko jest ok, odpalamy naszą metodę do danych
            exampleFunction(this);
        }
    };
    xmlhttp.open("GET", "data.xml", true);      //otwieramy połączenie
    xmlhttp.send();                             //wysyłamy żądanie
}
```

Po utworzeniu metody należy dodać metodę *exampleFunction(this)*, która obsłuży dane pobrane z pliku data.xml. Może ona wyglądać tak:

```
function exampleFunction(xml) {
    var i, j;
    var xmlDoc = xml.responseXML;              //odpowiedź z danymi
    var data="";
    var x = xmlDoc.getElementsByTagName("CD");  //elementy <CD>
    var y;
    for (i = 0; i <x.length; i++) {
        data += "<h2>Artist Name: ";
        data += x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue;
        data += "</h2><h3>Album title: "
        data += x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue;
        data += "</h3>";
        y = x[i].getElementsByTagName("TAGS")[0].getElementsByTagName("TAG");
        data += "<p><strong>Music category:</strong></p><ul>";
        for(j = 0; j <y.length; j++) {
            data += "<li>" + y[j].childNodes[0].nodeValue+ "</li>";
        }
        data += "</ul>";
    }
    document.getElementById("xmlContent").innerHTML = data;
}
```

A na końcu wywołać metodę *loadXMLDoc()*:

```
loadXMLDoc();
```

Dokument index.html będzie miał więc postać:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Dokument bez tytułu</title>
    <script>
      ... tutaj obie metody do wczytywania pliku data.xml oraz wywołanie
      loadXMLDoc();
    </script>
  </head>
  <body>
    <section id = "xmlContent">Tutaj będzie XML</section>
  </body>
</html>
```

Ćwiczenie 2. Wczytaj dane ze swojego pliku XML na stronę HTML

1. W katalogu z plikiem XML z poprzednich zajęć utwórz plik index.html.
2. Sformatuj go poprawnie: dodaj HEAD i BODY.
3. Wykorzystując obiekt XMLHttpRequest() wczytaj **WSZYSTKIE** dane xml do pliku html i wyświetl stosując odpowiednie formatowanie.
4. Korzystając z przykładu:
https://www.w3schools.com/xml/met_element_getattribute.asp wczytaj na stronę również atrybuty.
5. Na koniec sformatuj plik index.html wykorzystując style css - strona powinna zawierać **DWIE KOLUMNY** – jedną z danymi z pliku XML, druga powinna pozostać pusta.
6. Uporządkuj kod: przenieś funkcje JS i style css do osobnych plików i dołącz je do strony index.html.