

Przetwarzanie dokumentów XML i zaawansowane techniki WWW

“Wykorzystanie zewnętrznych źródeł danych”

(Zajęcia 11 - 30.05.2016 r.)

1. Wstęp: AJAX i jQuery

Poznaliśmy różnice pomiędzy synchronicznym i asynchronicznym modelem przetwarzania danych poprzez protokół HTTP. Dzięki asynchronicznej technologii możliwe jest np. uzupełnienie danego dokumentu HTML od dane pobrane z serwera bez konieczności przeładowania danego dokumentu HTML. Podobną funkcjonalność oferuje biblioteka jQuery.

Możemy asynchronicznie pobrać dodatkowe dane z serwera a następnie korzystając z metod dzięki którym możliwe jest modyfikowanie struktury DOM wyświetlić te dane użytkownikowi.

W jQuery metody odpowiedzialne za komunikację asynchroniczną to:

`.load()` - Pozwala na ustalenie zawartości elementu treścią pobraną z adresu URL podanego jako argument metody.

`.ajax()` - Pozwala na pobranie danych z adresu określonego jako parametr wywołania metody.

Metoda wykorzystanie metody `.load()` ma postać:

```
$( selektor ).load( URL, parametry-URL, function() );
```

Przykład użycia metody `load`:

```
$(function() {  
    $('#tresc').load('tresc.html', 'bcs=34&sort=asc', function(html) {  
        $('#span#info').text('Tresc została załadowana!');  
    });  
});
```

```
<body>  
    <div id="tresc"></div>  
    <span id="info"></span>  
</body>
```

Postać metody `.ajax()` jest następująca:

```
$.ajax( {  
    url: URL,  
    timeout: ms,  
    cache: true/false,  
    success: function(html) {},  
    beforeSend: function() {},  
    error: function() {}  
});
```

Metodę `.ajax` można wykorzystać do wywołania zewnętrznych usług dostarczających dane. Najczęściej używa się jej do usług własnego backendu. Jednak można użyć jej również, w ograniczonym zakresie do agregacji danych od zewnętrznych dostawców udostępniających je w postaci usługi API. Ograniczenia pojawiają się w sytuacji kiedy usługę działającą w innej domenie, niż ta z której hostowana jest nasza aplikacja. Jest to tzw. problem “cross-domenowy” (*ang. cross-domain*), objawiający się tym że ze względów bezpieczeństwa przeglądarki blokują

zapytania Ajax do innych domen niż te z której hostowana jest dana aplikacja. Jednym z możliwości rozwiązania tego problemu jest zezwolenie po stronie serwera i po stronie przeglądarki na tego typu zapytania, poprzez ustawienie tzw. *Cross-Origin Resource Sharing* (CORS). Inną możliwością jest wykorzystanie zapytania z użyciem techniki JSONP (JSON with Padding), w przypadku usług które zwracają dane w postaci obiektu JSON. Trzecią możliwością jest użycie zewnętrznego "proxy" które będzie pośredniczyć pomiędzy usługą zewnętrzną, a aplikacją. Jednymi z popularnych rozwiązań tego typu jest:

<https://crossorigin.me/>, lub <http://cors.io/>, które pozwala w łatwy sposób zasymulować CORS.

W tym wypadku należy w żądaniu ajaxowym zamiast docelowego adresu podać adres serwera CORS wraz z ścieżką do usługi zewnętrznej. Przykładowo jeśli mamy usługę zlokalizowaną pod adresem:

<http://www.nbp.pl/kursy/xml/a101z160527.xml>

która ma ograniczenia "cross-domenowe", to należy w żądaniu ajax użyć adresu proxy w następującej postaci:

<https://crossorigin.me/http://www.nbp.pl/kursy/xml/a101z160527.xml>
(adres_crossorigin.me) / (adres_usługi)

Lub

<http://cors.io/?u=http://www.nbp.pl/kursy/xml/a102z160530.xml>
(adres_cross.io) /?u= (adres_usługi)

2. Zapytanie AJAXowe za pomocą serwisu proxy CORS

Spróbujmy napisać z użyciem serwisu proxy zapytanie ajax do pobrania danych z tabeli dziennych kursów walut ze strony Narodowego Banku Polskiego. Różne tabele znajdują się na stronie <http://www.nbp.pl/home.aspx?f=/statystyka/kursy.html>. Wykorzystajmy "Tabele".

```
var root = 'http://cors.io/?u=http://www.nbp.pl';

$.ajax({
  url: root + '/kursy/xml/a102z160530.xml',
  dataType: 'xml',
  method: 'GET'
}).success(function(data) {
  console.log(data);
});
```

W wyniku wykonania zapytania AJAX dotaniemy z serwera odpowiedź w postaci pliku XML z którego możemy wyłuskać dane i osadzić je w odpowiednich elementach strony do tego przygotowanych.

Zadanie

Proszę przygotować szablon strony zawierający tabelę (użyć znaczników <table> <tr> <td>), do której będą trafiać dane pobrane żądaniem ajaxowym ze strony kursów walut NBP. Proszę użyć do tego celu proxy CORS z powyższego przykładu. Tabela powinna zawierać wszystkie dane odebrane z pliku XML.

3. Wykorzystanie techniki JSON-P

Wykorzystanie techniki JSON-P, jest bardzo od strony składniowej jest bardzo podobne do skorzystania z serwera proxy CORS. Różnica polega na określeniu innego typu danych odbieranych z serwera w dyrektywie:

```
dataType: 'jsonp'
```

Tym samym informujemy metodę ajax o wykonaniu zapytania JSONP.

Aby przetestować tą metodę skorzystamy z serwisu testowego API dostępnego pod adresem:

<http://jsonplaceholder.typicode.com/>

Zadanie

Proszę przygotować szablon wraz z metodą testową z w/w strony z przykładem JSONP:

```
var root = 'http://jsonplaceholder.typicode.com';

$.ajax({
  url: root + '/posts/1',
  method: 'GET'
}).then(function(data) {
  console.log(data);
});
```

Następnie proszę przygotować miejsce do którego będą trafiać pobrane dane, może to być np. Element blokowy <div>.