**Aplikacje WWW Zadanie 19**

**AJAX w JQUERY+JSON**

Do obsługi połączeń AJAXowy w jQuery można użyć metod .get, .post lub po prostu .ajax.

$.ajax({

    type     : "POST",

    url      : "zapisz\_do\_bazy.php",

    data     : {

            imie : 'Marcin',

            kraj : 'Polska'

    },

    success : function(msg) {

        //ten fragment wykona się po POMYŚLNYM zakończeniu połączenia

        //msg zawiera dane zwrócone z serwera

    },

    complete : function(r) {

        //ten fragment wykona się po ZAKONCZENIU połączenia

        //"r" to przykładowa nazwa zmiennej, która zawiera dane zwrócone

    },

    error:    function(error) {

        //ten fragment wykona się w przypadku BŁĘDU

    }

});

Powyższa metoda wysyła w JQuery+AJAX dane metodą POST. Dane po stronie serwera w PHP odbieramy zatem jako $\_POST[‘imie] oraz $\_POST[‘nazwisko’]

Parametry połączenia:

* type - typ połączenia: POST lub GET. Domyślnym jest GET
* data - przekazywane dane. Przykłady na stronie ukazują różne sposoby definiowania danych
* dataType - typ danych jaki oczekujemy. Omówimy go za moment. Jeżeli nie zostanie ustawiony, jQuery spróbuje określić typ danych na podstawie tego co dostanie w odpowiedzi.
* zdarzenia - zdarzenia związane z połączeniem

Obsługa odpowiedzi:

1. **success** - wywoływane po pomyślnym wysłaniu danych do skryptu
2. **complete** - wywoływane po zakończeniu połączenia
3. **error** - wywoływane gdy nastąpi błąd w połączeniu

Wysyłanie danych GETem jest lekko innej, różni się polem data

$.ajax({

    type     : "GET",

    url      : "zapisz\_do\_bazy.php",

    data     : "imie=Marcin&kraj=Polska", //inny sposób przekazywania danych

    success : function(zwrot) {

        $('#wynik').html(zwrot);//zapisze dane w oknie o id wynik jeśli takowe istnieje

    },

    complete : function(r) {

        $('#loading').hide();

    },

    error: function(err) {

        console.log(err)

    }

});

Dane można odbierać tekstowo, jako XML lub poprzez format JSON. Jest to sposób zapisu danych mający nam dawać uproszczony dostęp do przesyłanych informacji. Tutaj jsona parsuje sukces, a nie complete.

$("input#start").click(function() {

    var zmienna1 = $('.jakis\_input\_1').val();

    var zmienna2 = $('.jakis\_input\_2').val();

    $.ajax({

        type: "POST",

        url: "test.php",

        dataType : 'json',

        data: {

            dana1 : zmienna1,

            dana2 : zmienna2,

        },

        success : function(json) {

            alert( json['wynik\_1'] + "\n" + json['wynik2']);

        }

    });

});

Tutaj dane zostaną przesłane do pliku PHP i będą tam dostępne poprzez $\_POST. Dane po powrocie wrócą już jako JSON. Plik PHP do obróbki wygląda następująco:

<?php

    $a = $\_POST['dana1'];

    $b = $\_POST['dana2'];

    $tablica = array();

    $tablica['wynik\_1'] = 'Po obróbce ' + $a + $a; //przykładowe działanie

    $tablica['wynik\_2'] = 'Po obróbce ' + $b + $b;

    echo json\_encode($tablica); //przekształcamy tablicę na zapis typu json

?>

Zwrócone dane to **{"wynik\_1":0,"wynik\_2":0}**

Zadania do wykonania:

* 1. Spróbuj Ajaxem wysłać imię i nazwisko metodą GET oraz metodą POST. Pamiętaj, że tutaj stosujemy jQuery, więc podepnij tą bibliotekę. Niech na serwerze istnieje plik PHP, który odbierze te dane i odeśle tekst: Witaj Jan Kowalski, gdzie Jan oraz Kowalski to przesłane dane.
  2. Spróbuj to samo wykonać przy pomocy JSON.
  3. Zbuduj na stronie prosty formularz na imię i nazwisko. Niech dane będą wysyłane teraz do pliku PHP, który wrzuci je do bazy danych. Po zapisie niech skrypt PHP wyśle informacje o wszystkich imionach i nazwiskach z bazy. Do pobrania długości danych służy json.length

Więcej informacji na stronie: <http://kursjs.pl/kurs/jquery/jquery_ajax.html>