

Korzystanie z jQuery i Ajax

Spis treści

- [Korzystanie jQuery and Ajax](#)
 - [O jQuery](#)
 - [jQuery \\$](#)
 - [Funkcje w jQuery](#)
 - [Selektory](#)
 - [Metody i wywołania zwrótne](#)
 - [Dostępne wtyczki w PrestaShop](#)
 - [Kilka innych narzędzi](#)
 - [Wykonywanie poczty Ajax z jQuery](#)
 - [O Ajax](#)
 - [Metody jQuery's Ajax](#)

Korzystanie jQuery and Ajax

O jQuery

jQuery jest solidną biblioteką JavaScript. Wśród wielu zalet:

- Działa zgodnie z oczekiwaniami wielu przeglądarek internetowych.
- Jasna, zwiza i intuicyjna architektura
- Wiele dostępnych wtyczek.


Możesz się dowiedzieć więcej o jQuery na oficjalnej stronie: <http://jquery.com/>

jQuery \$

jQuery posiada funkcję `$()`, nazwę, która zawiera wszystkie inne funkcje.

Funkcja `$()` może być użyta na wiele sposobów:

- Przy zastosowaniu funkcji jako parametr, `$()` będzie wykonywać funkcje, gdy strony w DOM są w pełni załadowane.
Na przykład: `$(function(){ /* do something */ });`
- Po przejściu ciągu zawierającego selektor CSS `$()` zwróci wszystkie wyzwy HTML, które odpowiadają selektorowi.
Na przykład: `$('ul#nav');`
- Ten zestaw wywołań może być przypisany do każdej z metod jQuery. Na przykład. Jeśli chcesz ukryć elementy nawigacji zwracane przez powyższy selektor.
Na przykład: `$('ul#nav').slideUp('fast');`
- Przy użyciu czystego kodu HTML jako jego parametru `$()` stworzy wyzwy (element DOM). Ten wyzwy może znowu być stosowany z dowolną metodą jQuery
Na przykład: `$("Sign Off").appendTo("ul#nav");`

 `$()` jest jedynie skrótem specyficznej biblioteki funkcji `jQuery()`. Istnieje w celu przyspieszenia kodowania. Wiele innych bibliotek JavaScript używa `$()` ich skrótów.

W związku z tym te dwie linie są równoważne:

```
$('#ul#nav').slideUp('fast');  
jQuery('#ul#nav').slideUp('fast');
```

Jeli mieszasz biblioteki JavaScript w swoim temacie, może będzie lepiej powiedzieć `jQuery` aby uwolni `$()`. Prosty zwrot `$.noConflict()`, i od tej pory każde wywołanie `jQuery()` będzie wymaga korzystania z `$()`.

Możesz dowiedzieć się więcej na temat `jQuery()` i `$()` na oficjalnej stronie: <http://api.jquery.com/jquery/>

Funkcje w jQuery

jQuery's namespace `$` zawiera wszystkie funkcje, które nie są dołączone do konkretnego wza.

Wśród nich znajduje się kilka funkcji Ajax i funkcji użytecznych. Na przykład:

```
$.post('/handler.php', {'action': 'purchase', 'product': 434}, function(data){/* do something */});
```

Można dołączyć zdarzenia do zbioru skróconych wzywań zwróconych przez `$()`. Jedną z zalet korzystania z jQuery, po raz kolejny jest to, że podczas obsługi zdarzenia, pomaga on poprzez harmonizację, tak, że nie trzeba uzgodnia specyfiki każdej przeglądarki.

Aby przypisać funkcję click event, jest prosty dodatek `.click(function)`.

W celu wygenerowania click event, można również użyć `.click()` z każdym parametrem. Na przykład: `$('#button').click(function(){/* do something */});`

Możesz się dowiedzieć więcej na temat jQuery's Ajax i funkcji użytecznych na oficjalnej stronie:

- <http://api.jquery.com/category/ajax/> (na przykład, <http://api.jquery.com/jquery.post/>)
- <http://api.jquery.com/category/utilities/>

Selektory

jQuery oferuje dwa sposoby, aby wybrać elementy strony:

- Pocz selektory CSS i XPath w ciągu znaków, które są używane z funkcją `$()`. Na przykład: `$("div > ul a")`
- Użyj jednej z kilku metod, które są już dostępne w jQuery namespace.

Oba te sposoby mogą być używane

```

var my_jQuery_object = $("#my_image");
var my_jQuery_object = $("#menu a");
var my_jQuery_object = $("#id > .classe, #id td:last-child");

/* returns the 'td' elements within the odd 'tr'. */
var my_jQuery_object = $('tr:odd td');

/* returns the fourth paragraph. */
var my_jQuery_object = $("p:eq(4)");

/* returns the 7 first paragraphs. */
var my_jQuery_object = $("p:lt(8)");

```

Możesz dowiedzieć się więcej na temat selektorów jQuery's na oficjalnej stronie: <http://api.jquery.com/category/selectors/>

Metody i wywołania zwrótne

Cy zestaw metod dostępny w standardowym API to: manipulacja DOM, manipulacja CSS, event management, efekty wizualne, etc.

Na przykład jeśli chcesz, aby zniknęły powoli wszystkie akapity na stronie, użyj tego:

```

$("p").fadeOut();

```

Niektóre metody (takie jak `fadeOut()`) przyjmują inne metody jako parametr. Taki sposób wykonany raz jest już zrobiony. To nazywamy wywołaniem zwrótnym

Na przykład:

```

$(".test").fadeOut("slow",function(){
    $(this).fadeIn("slow");
});

```

Wszystkie metody powrotu przedmiotu `jQuery`. To sprawia, że metody acuchają się bez ograniczeń. Można nawet napisać swój kod, tak, że jest on czytany jak blok funkcyjny.

Na przykład, ten kod działa doskonale i atwy do odczytu i aktualizacji:

```

$(".emptyContent").append("This is a test")
.css("border", "1px solid red")
.addClass("fullContent")
.removeClass("emptyContent");

```

Dostępne wtyczki w PrestaShop

To jest lista wtyczek jQuery które są dostępne w domyślnej instalacji PrestaShop:

Nazwa wtyczki	Opis wtyczki
jquery.colorpicker	Selektor stylu kolorów PrestaShop.

jquery.cookie-plugin	Odczytuje, zapisuje i usuwa pliki cookie.
jquery.dimensions	Zarządza elementami wymiarów.
jquery.easing	Zarządza prdkoci elementów.
jquery.excanvas	Zmienia obszar elementu roboczego (zaokrąglone naroniki/gradientsy/krycia /rysowanie lini, uku itp.).
jquery.fieldselection	Wykorzystuje i zastępuje tekst w wybranej strefie.
jquery.flip	Odwraca element.
jquery.flot	Tworzy wykres prezentacji danych, takich jak krzywe, bary, itp.
jquery.highlight	Dodaje skadnie koloryzacji.
jquery.hoverIntent	Dodaje efekt przewidywania na JavaScript hover event.
jquery.idTabs	Zarządza kartami.
jquery.ifxtransfer	Animuje elementem podczas przenoszenia z jednego pojemnika do drugiego.
jquery.jqminmax	Dodaje min-width, max-width, min-height i max-height na wszystkich przegldarkach.
jquery.pngFix	Zarządza przejrzystoci w IE 5.5 i IE 6.
jquery.scrollTo	Dodaje strony lub przewijanie elementu do okrelonej pozycji.
jquery.serialScroll	Wprowadza szereg elementów przewijania do okrelonej pozycji.
jquery.tablednd	Przeciga i upuszcza wiersze tabeli.
jquery.typewatch	Wykonuje funkcje kiedy uytownik wpisa jaki tekst w strefie i przesta pisa po pewnym czasie.
jquery.validate-creditcard	Weryfikuje numer karty kredytowej w zalenoci od jego typu.

Kilka innych narzdzi

jQuery's API jest bardzo kompletne, i spdzisz godziny na poszukiwaniu nowych moliwoci.

Oto kilka funkcji, które mog by niezwykle uyteczne podczas tworzenia tematu.

Po pierwsze, `each()`, co umoliwia petli przejscie przez liste elementów:

```
$("#img").each(function(){ console.log($(this).attr("src")); });
```

Po drugie, `browser`, jest obiektem, który pozwala dowiedzie si, która przegldarka pracuje z:

```
if($.browser.msie) {
  if($.browser.version == 6) {
    // Your IE6 specific code.
  } else {
    // Code for any other IE browser.
  }
}
```

Wykonywanie pocze Ajax z jQuery

O Ajax

Termin "Ajax" jest naprawd skrótem od Asynchronous JavaScript i XML. Opisuje uycie JavaScript do zaadowania nowej zawartoci do biecej strony bez koniecznoci przeadowania caj. Proces ten nazywany jest asynchronicznym, poniewa gdy adanie Ajax zostao wysane do serwera webowego, przegldarka nie musi czeka na odpowied w celu wykonania innych zada. Przesyana tre nie musi by sformatowana w XML, mona uy JSON, HTML, lub zwyky tekst.

W efekcie uycie Ajax sprawia, e mozliwe jest zbudowanie bardzo dynamicznych stron internetowych

Moesz si dowiedzie wiecej na temat techniki Ajax na nastepujcej stronie:

- Wikipedia: <https://pl.wikipedia.org/wiki/AJAX>
- Mozilla Developer Network: <https://developer.mozilla.org/en/docs/AJAX>

About JSON

JavaScript Object Notation jest najczciej stosowanym formatem podczas przesywania danych przy uyciu techniki Ajax, z dwóch gównych powodów, uwaa sie, e jest lejszy ni XML, a to moe by bardzo tve wykorzystane przez JavaScript gdy przypomina podzbiór tego jzyka. Moesz dowiedzie si wiecej o JSON na nastpujcych stronach.

- <http://json.org/>
- <https://pl.wikipedia.org/wiki/JSON>
- <https://developer.mozilla.org/en-US/docs/JSON>.

Jak uy Ajax w PrestaShop

Domyne kontrolery PrestaShop uzywaj standardowej penej strony- procesu przeadowania.

Jeli PrestaShop wykryje parametr "ajax" w GET albo POST zapytania, kontroler `ajax` obiektu jest przeczyony na true: `$this->ajax = true;`

Na przykad, Jeli kod wyzwalaa URL, takie jak : `http://...&ajax&action=updateList` ...regulator nastpnie wykona `displayAjaxUpdateList()` jeli metoda istnieje. Jeli nie istnieje, to wykona `displayAjax()` metod (domyln).

Dlatego on musi zawiera kod który wemie pod uwag wezwanie Ajax. Oto jak mona napisa prost kwerend za pomoc Ajax jQuery:

```
var query = $.ajax({
  type: 'POST',
  url: baseDir + 'modules/mymodule/ajax.php',
  data: 'method=myMethod&id_data=' + $('#id_data').val(),
  dataType: 'json',
  success: function(json) {
    // ....
  }
});
```

Oto jak skrypt `ajax.php` bdzie dziaaa:

```
// Located in /modules/mymodule/ajax.php
require_once(dirname(__FILE__).'../../../config/config.inc.php');
require_once(dirname(__FILE__).'../../../init.php');
switch (Tools::getValue('method')) {
    case 'myMethod' :
        die(Tools::jsonEncode(array('result' => 'my_value')));
        break;
    default:
        exit;
}
exit;
```

Jak widzisz w przykładzie powyższy kod, PrestaShop `Tools` zawiera obiekt `jsonEncode()`, to metoda, która sprawia, że łatwo przekształci tablicę PHP do obiektu JSON:

```
public function displayAjax() {
    $return = array(
        'hasError' => true,
        'errors' => 'Ceci est le message'
    );
    die(Tools::jsonEncode($return));
}
```

Metody jQuery's Ajax

jQuery.ajax(url [, ustawienia])

Parametry:

- url: URL do której należy wysłać zapytanie
- ustawienia: opcje i funkcje

Niektóre opcje:

`async` (boolean): domyślnie jest `true`

- `type`: domyślnym jest `'GET'`
- `cache` (logiczna): domyślnie jest `true`
- `data`: GET tablica wysyłana do serwera
- `dataType`: zarówno `xml`, `json`, `script` albo `html`

Niektóre funkcje:

- `beforeSend`: wyzwalane przed wywołaniem Ajax
- `error`: w przypadku błędów
- `success`: w przypadku powodzenia
- `timeout`: w przypadku przekroczenia czasu
- `complete`: wyzwalane po sukcesie zwołania, bez względu na wynik.

Możesz się dowiedzieć więcej o `ajax()` tutaj: <http://api.jquery.com/jQuery.ajax/>

Alternatywny: `jQuery.load(url [, data] [, complete(responseText, textStatus, XMLHttpRequest)])`

Metoda ta stosowana jest do bezpośredniego załadunku wyniku HTML do wywołania Ajax, w obrębie elementu na bieżącej stronie.

Na przykład :

```
$('#result').load('ajax/test.html', function() {  
    alert('Load was performed.');
```

Alternatywny: `jQuery.get(url [, data] [, success(data, textStatus, jqXHR)] [, dataType])`

Metoda ta pozwala na poczenie si z zapytaniem Ajax ze skryptem HTTP GET. Jest to równoznaczne z nastupjym poczeniem Ajax:

```
$.ajax({  
    url: url,  
    data: data,  
    success: success,  
    dataType: dataType  
});
```

Wartoci zwracane na wywoanie Ajax jQuery

Wszystkie te metody jQuery zwracaj obiekt `jqXHR`, który jest przedueniem obiektu `XMLHttpRequest`. Obiekt ten umoliwia wywoanie rónych funkcji w zalenoci od wyniku poczenia:

- Powodzenie: `jqXHR.done(function(data, textStatus, jqXHR) {});`
- Bd: `jqXHR.fail(function(jqXHR, textStatus, errorThrown) {});`
- Powodzenie albo Bd: `jqXHR.always(function(data|jqXHR, textStatus, jqXHR|errorThrown){ });`
- Powodzenie i Bd: `jqXHR.then(function(data, textStatus, jqXHR) {}, function(jqXHR, textStatus, errorThrown) {});`

Oto przykad z poaczenia funkcji w zalenoci od wyniku kwerendy:

```
// Assign handlers immediately after making the request,  
// and remember the jqxhr object for this request  
var jqxhr = $.get("example.php", function() {  
    alert("success");  
})  
.done(function() { alert("second success"); })  
.fail(function() { alert("error"); })  
.always(function() { alert("finished"); });  
  
// perform other work here ...  
// Set another completion function for the request above  
jqxhr.always(function(){ alert("second finished"); });
```