

Hoofdstuk 10 : Ajax en jQuery

Hoofdstuk 10 : Ajax en jQuery

1. jQuery : Inleiding
2. jQuery : Selectors
3. jQuery : Events – Animation
4. jQuery : HTML manipulation – css
5. AJAX : Inleiding
6. AJAX : XMLHttpRequest
7. AJAX : JSON
8. Ajax : jQuery
9. Ajax : ASP.NET MVC Unobtrusive Ajax
10. Hijax
11. jQuery UI : Datepicker
12. REST API's gebruiken van andere sites

jQuery : Inleiding

jQuery is a new kind of JavaScript Library.

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. **jQuery is designed to change the way that you write JavaScript.**

► Meer op <http://jQuery.com>

jQuery : Inleiding

- Gecreëerd door John Resig
- Open source: MIT and GPL license
- Cross-browser compatible: IE 6+, Firefox 2+, Opera 9+, Safari 2+
- Small footprint: 15KB
- jQuery is een lightweight "write less, do more" JavaScript library.
- The jQuery library bevat :
 - HTML element selections
 - HTML element manipulation
 - CSS manipulation
 - HTML event functions
 - JavaScript Effects and animations
 - HTML DOM traversal and modification
 - AJAX
 - Utilities

jQuery : Inleiding

► Javascript

```
var innerWidth;  
// all browsers except IE < 9  
if (typeof window.innerWidth !== "undefined") {  
    innerWidth = window.innerWidth;  
}  
else {  
    innerWidth = document.documentElement.clientWidth;  
}  
document.getElementById("myDiv").innerText = "InnerWidth of the window is " + innerWidth;
```

► jQuery

```
var innerWidth = $(window).width(), innerHeight = $(window).height();  
$("#myDiv").text("InnerWidth of the window is: " + innerWidth);
```

jQuery

- Selectors zijn veel eenvoudiger
- Bevat wrappers rond DOM objecten die jQueryfied objecten retourneren : \$(window) is wrapper voor window Object
- bevat dan de functies bvb width()
- => geen cross browser problemen meer
- => Bijna alle functies retourneren een jQuery object => method chaining

jQuery : Inleiding

► jQuery onderdelen

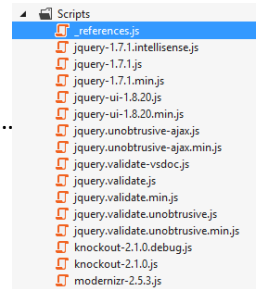
- **jQuery library** is de core functionaliteit van jQuery.
 - Bevat ondersteuning voor doorlopen van DOM, selecteren en manipuleren van DOM elementen, implementatie van AJAX en event afhandeling.
Meer op <http://jquery.com/>
- **jQuery UI** is een verzameling van user interface componenten die HTML elementen converteren in "rich objects" met gedrag, events en styling (vb. DatePicker).
Meer op <http://jqueryui.com/>
- **jQuery plugins** zijn third-party stukjes software gebouwd met de jQuery plug-in patterns en core library.
Meer op <http://plugins.jquery.com/>

jQuery : Inleiding

Om de laatste versie te bekomen van script libraries: Tools > Library Package Manager > Package Manager Console > commando :
~~Update Package~~

► Script libraries in MVC project > Scripts folder

- *intellisense.js : extensie in VS om Intellisense te verschaffen bij editeren jquery.
- -references.js : gebruikt door VS voor de basis intellisense (methodenamen, pars,...)
- *-min.js : minified script files : spaces, comments,... verwijderd
- jQuery-1.7.1* : core jQuery library
- jQuery-UI* : jQuery plugin met effects en widgets (datepicker, slides,...)
- jQuery-validate* : jQuery validation plugin voor client side validatie
- *.unobtrusive* : extensie MVC : bridge tussen MVC en jquery libraries voor validatie en ajax.
- Knockout : MVVM pattern framework + databinding
- Modernizr : open-source JavaScript library voor bouwen HTML5 en CSS3 websites



jQuery : Inleiding

► Script libraries : Toevoegen javascript/jquery aan View

- In _Layout.cshtml

```
@Scripts.Render("~/bundles/jquery")
@RenderSection("scripts", required: false)
```

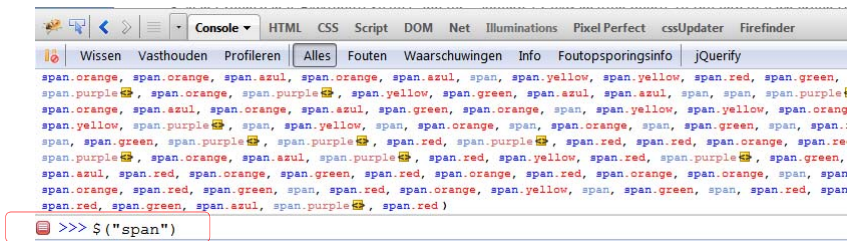
- In View

```
@section Scripts
{
}
```

- Opm : scripts voor performantie bovenaan toevoegen (uitz. Modernizr)

jQuery : Inleiding

- ▶ jQuery hulpmiddelen : ingeven van javascript
 - FireFox : Firebug > Console tab



- Chrome : Customize and Control Google Chrome button > Tools (Extra's) > Javascript console

jQuery : Inleiding

- ▶ Tips debuggen Javascript/jQuery :
 - Firefox : Firebug > Script . Javascript fouten weergeven : Console
 - Chrome : Google Chrome aanpassen en beheren > Extra > Javascript Console (Fouten : onderaan klikken op Errors)
 - IE8 : Klik F12 of selecteer Tools menu > Developer tools > 'script' tab. Meer op [http://msdn.microsoft.com/en-us/library/dd565625\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd565625(v=VS.85).aspx) (Start without Debugging

jQuery : Inleiding

► Hello jQuery

- Libraries reeds geincludeerd in de layout page

```
@Scripts.Render("~/bundles/jquery")
```

- jQuery leest de DOM structuur of past deze aan
- => Pas als document geladen is (anders mogelijks fouten)

```
$(document).ready(function() {  
    // do stuff when DOM is ready  
});
```

```
//verkorte schrijfwijze  
$(function() {  
});
```

- **!!!MVC : Naamgeving elementen in gegenereerde Html pagina**
 - `EditorFor(m=>m.Description)` => name van tag is Description
 - `EditorFor(m=>m.Gemeente.Naam)` => name van tag is Gemeente_Naam

jQuery : Inleiding

► Hello jQuery

- Pas event aan

```
@section Scripts
```

```
{  
    <script type="text/javascript">  
        $(document).ready(function () {  
            $("a").click(function () {  
                alert("Hello world!");  
            });  
        });  
    </script>  
}
```

```
<h2>HelloWorld</h2>|  
<a href="">Link</a>
```

- `$("a")` = selector (hier alle a href tags) => creëert een jQuery object
- `click()` function = methode van dat object. Het bindt een click event aan alle a tags en voert de functie uit bij een click
- Er wordt gewerkt met anonieme functies!

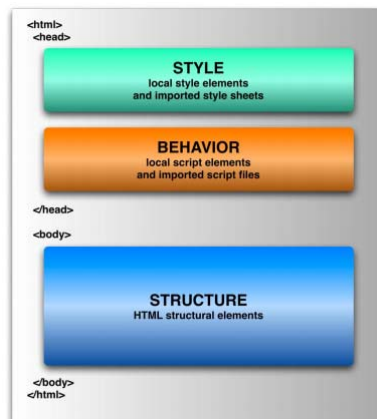
- Doet hetzelfde als onderstaande, maar voor alle a tags

```
<a href="" onclick="alert('Hello world')">Link</a>
```

jQuery : Inleiding

► Hello jQuery

- Mooie scheiding van structuur (HTML) en gedrag (JS)



jQuery : Inleiding

► Basis syntax

- Concept: "select some elements and do something with them"
- **\$(selector).action()**
 - een dollar sign \$ wijst op een jQuery instructie
 - een (selector) dient om HTML elementen te selecteren (een array van elementen). (=> Selector engine : Sizzle)
 - een jQuery action() wordt uitgevoerd op de geselecteerde elementen
 - Enkele voorbeelden
 - `$(this).hide()` : verbergt huidig element
 - `$("p").hide()` : verbergt alle paragrafen
- Method **chaining** : bijna alle methodes op selectors retourneren jQuery objecten (wrappers rond DOM elementen), dus kan je chainen: `$(sel).abc().def()`
 - `$("p").css("color", "red").fadeOut()`
 - `$(".normal > span").contains("Welcome!").text("...").css({color: "red"});`

jQuery : Selectors

► Overzicht van selectors :

<http://api.jquery.com/category/selectors/>

Basic

The following selectors are based on the Cascading Style Sheet 1 specification, as outlined by the W3C. For more information about the specifications, visit <http://www.w3.org/Style/CSS/#specs>.

All Selector ("*")

Selects all elements.

Class Selector (".class")

Selects all elements with the given class.

Element Selector ("element")

Selects all elements with the given tag name.

ID Selector ("#id")

Selects a single element with the given id attribute.

Multiple Selector ("selector1, selector2, selectorN")

Selects the combined results of all the specified selectors.

jQuery : Selectors

► jQuery Element Selectors

◦ jQuery gebruikt **CSS selectors** voor selectie HTML elementen

- \$(this) : het huidige element
- \$("p") : alle <p> tags.
- \$("p.intro") : alle <p> tags met class="intro".
- \$(".intro") : alle elementen met class="intro"
- \$("#intro") : het element met id "intro"
- \$("p#demo") : de eerste <p> tag met id="demo".
- \$("ul li") of \$("ul > li") : de li elementen van alle ul tags
- \$("ul li:first") : het eerst element van
- \$("div#intro .head") : alle elementen met class="head" binnen een <div> element met id="intro"

: id selector

. : class selector

> : parent – child relatie

Selector	Example	Selects
*	\$("*")	All elements
#id	\$("#lastname")	The element with id=lastname
.class	\$(".intro")	All elements with class="intro"
element	\$("p")	All p elements
.class.class	\$(".intro.demo")	All elements with the classes "intro" and "demo"
:first	\$("p:first")	The first p element
:last	\$("p:last")	The last p element
:even	\$("tr:even")	All even tr elements
:odd	\$("tr:odd")	All odd tr elements
:eq(index)	\$("ul li:eq(3)")	The fourth element in a list (index starts at 0)
:gt(no)	\$("ul li:gt(3)")	List elements with an index greater than 3
:lt(no)	\$("ul li:lt(3)")	List elements with an index less than 3
:not(selector)	\$("input:not(:empty)")	All input elements that are not empty
:header	\$(":header")	All header elements h1, h2 ...
:animated	\$(":animated")	All animated elements
:contains(text)	\$(":contains('W3Schools')")	All elements which contains the text
:empty	\$(":empty")	All elements with no child (elements) nodes
:hidden	\$(":p:hidden")	All hidden p elements
:visible	\$(":table:visible")	All visible tables
s1,s2,s3	\$("th,td,.intro")	All elements with matching selectors

jQuery : Selectors

- ▶ jQuery Attribute Selectors
 - jQuery gebruikt **XPath** voor selectie elementen met bepaald attribuut
 - \$("[href]") : alle elementen met href attribuut.
 - \$("[href='#']") : alle elementen met href waarde gelijk aan"#".
 - \$("[href!='#']") : alle elementen met href attribuut !="#".
 - \$("[href\$='.jpg']") : alle elementen met href attribuut *eindigend* op ".jpg".
 - \$('input[name*="man"]') : alle input elementen met name attribuut die het woord "man" *bevat*
 - \$('input[name^="news"]') : alle input elementen met name attribuut die *begint* met "nieuws"

jQuery : Selectors

► jQuery Form selectors

:input	<code>\$(":input")</code>	All input elements
:text	<code>\$(":text")</code>	All input elements with type="text"
:password	<code>\$(":password")</code>	All input elements with type="password"
:radio	<code>\$(":radio")</code>	All input elements with type="radio"
:checkbox	<code>\$(":checkbox")</code>	All input elements with type="checkbox"
:submit	<code>\$(":submit")</code>	All input elements with type="submit"
:reset	<code>\$(":reset")</code>	All input elements with type="reset"
:button	<code>\$(":button")</code>	All input elements with type="button"
:image	<code>\$(":image")</code>	All input elements with type="image"
:file	<code>\$(":file")</code>	All input elements with type="file"
:enabled	<code>\$(":enabled")</code>	All enabled input elements
:disabled	<code>\$(":disabled")</code>	All disabled input elements
:selected	<code>\$(":selected")</code>	All selected input elements
:checked	<code>\$(":checked")</code>	All checked input elements

jQuery : Selectors

► Een voorbeeldje

Ajax en jQuery

jQuery XMLHttpRequest Hixax Hixax JSON jQueryUI About

ElementSelectors

My name is Donald

I live in Duckburg

My best friend is Mickey

Who is your favourite:

- Goofy
- Mickey
- Pluto

www.hogent.be

Highlight

Back to list

Vul selector in en klik Highlight

-Alle p tags

-Eerste p tag

-p tag met class error

-ul met id choose, alle li tags

-a tag met href attribuut

jQuery : Selectors

- ElementSelectors.cshtml

```
@section Scripts
{
<script type="text/javascript">
$(document).ready(function () {
    $("button").click(function () {
        $("p, ul, li, a, :text").removeAttr("style");
        $("#selector").val().css("background-color", "yellow");
    });
});
</script>
}

<h2>ElementSelectors</h2>
<p class="field-validation-error">My name is Donald</p>
<p>I live in Duckburg</p>
<p>My best friend is Mickey</p>
Who is your favourite:
<ul id="choose">
<li>Goofy</li>
<li>Mickey</li>
<li>Pluto</li>
</ul>
<p><a href="http://www.hogent.be">www.hogent.be</a></p>
<input type="text" id="selector" />
<button>Highlight</button>
<p>@Html.ActionLink("Back to list", "Index") </p>
```

HoGent

jQuery : Events

- ▶ Event methode

- triggert een event
- of koppelt (binding) een functie aan een bepaald event voor alle matching elementen
- Meer op : <http://api.jquery.com/category/events/>

- Trigger voorbeeld :

- \$('#target').submit(); submits element met id target

- Binding voorbeeld :

- \$("button").click(function(){
 \$("img").hide()
});
- Koppelt een functie aan click event van de knoppen, nl. als er op de knop geklikt wordt, worden alle beeldjes verborgen

HoGent

jQuery : Events

► Event methode (voorbeeld)

```
@section Scripts
{
<script type="text/javascript">
$(document).ready(function () {
    $("button").click(function () {
        $("p").toggle(1000);
    });
});
</script>
}
<h2>Events</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Hide the paragraphs</button>
<br /><br />
@Html.ActionLink("Back to list", "Index")
```

Pas aan :

- slideUp(1000);
- toggle(1000); (2 * op knop klikken)
- fadeOut(1000)

jQuery : Animate functions

► Animaties

Function	Description
\$(selector).hide()	Hide selected elements
\$(selector).show()	Show selected elements
\$(selector).toggle()	Toggle (between hide and show) selected elements
\$(selector).slideDown()	Slide-down (show) selected elements
\$(selector).slideUp()	Slide-up (hide) selected elements
\$(selector).slideToggle()	Toggle slide-up and slide-down of selected elements
\$(selector).fadeIn()	Fade in selected elements
\$(selector).fadeOut()	Fade out selected elements
\$(selector).fadeTo()	Fade out selected elements to a given opacity
\$(selector).animate()	Run a custom animation on selected elements

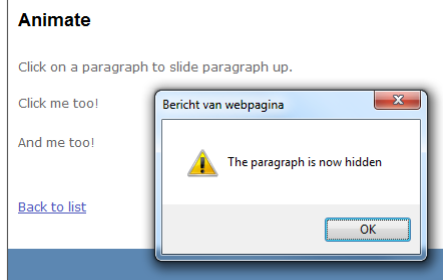
- Animatie methodes bevatten meestal 2 parameters
 - speed : de snelheid van uitvoering : "slow", "fast", "normal", of milliseconds
 - callback : naam vd functie die moet worden uitgevoerd als effect volledig uitgevoerd is

jQuery : Animate functions

- CallBack = functie die wordt aangeroepen als animatie is afgelopen

@section Scripts

```
{
<script type="text/javascript">
$(document).ready(function () {
    $("p").click(function () {
        $(this).slideUp(1000, function () {
            alert("The paragraph is now hidden");
        });
    });
});
</script>
<h2>Animate</h2>
Click on a paragraph to slide paragraph up.
<p>If you click on me, I will slide up.</p>
<p>Click me too!</p>
<p>And me too!</p>
<br /><br />
@Html.ActionLink("Back to list", "Index")
}
```



Pag. 25

jQuery: manipulatie Html

HoGer

Method	Description
addClass()	Adds one or more classes (for CSS) to selected elements
after()	Inserts content after selected elements
append()	Inserts content at the end of (but still inside) selected elements
appendTo()	Inserts content at the end of (but still inside) selected elements
attr()	Sets or returns an attribute and value of selected elements
before()	Inserts content before selected elements
clone()	Makes a copy of selected elements
detach()	Removes (but keeps a copy of) selected elements
empty()	Removes all child elements and content from selected elements
hasClass()	Checks if any of the selected elements have a specified class (for CSS)
html()	Sets or returns the content of selected elements
insertAfter()	Inserts HTML markup or elements after selected elements
insertBefore()	Inserts HTML markup or elements before selected elements
prepend()	Inserts content at the beginning of (but still inside) selected elements
prependTo()	Inserts content at the beginning of (but still inside) selected elements
remove()	Removes selected elements
removeAttr()	Removes an attribute from selected elements
removeClass()	Removes one or more classes (for CSS) from selected elements
replaceAll()	Replaces selected elements with new content
replaceWith()	Replaces selected elements with new content
text()	Sets or returns the text content of selected elements
toggleClass()	Toggles between adding/removing one or more classes (for CSS) from selected elements
unwrap()	Removes the parent element of the selected elements
val()	Sets or returns the value attribute of the selected elements (form elements)
wrap()	Wraps specified HTML element(s) around each selected element
wrapAll()	Wraps specified HTML element(s) around all selected elements
wrapInner()	Wraps specified HTML element(s) around the content of each selected element

Pag. 26

jQuery : CSS

- ▶ 1 belangrijke methode `css()` in 3 flavours
 - `css(name)` : retourneert css property waarde
 - `$(this).css("background-color");`
 - `css(name,value)` : stelt css property in op waarde
 - `$("#p").css("background-color","yellow");`
 - `css({properties})` : stelt meerdere css properties in
 - `$("#p").css({"background-color":"yellow","font-size":"200%"});`
- ▶ Height en width
 - `$("#div1").height("200px");`
 - `$("#div2").width("300px");`

jQuery : Oefening

Exercise

Seven Simple Selector Exercises

As we have seen, you select items using CSS selectors and then DO something to them. Inside the DIV with id "special-container" below are 200 SPAN tags, each with one of 7 class tags for each color of the rainbow. They are generated at random. Let's do a few simple exercises to get started.

- First - create the document ready wrapper. Everything else will go inside that. The reason: The javascript is at the TOP of the page, and if it gets called before the page is loaded, there will be no elements to select!
- Next, select all of the spans with class of "red" and make the css "font-weight" property "bold"
- Now, a simple demo that jQuery is just javascript. When you select things, you are creating an array of jQuery objects. So, you can use array functions like `length`. Create an alert of how many orange tags there are;
- Make every other yellow tag have a 1 pixel border (set "border" to "1px solid black");
- Alan Greenspan is Exotising!! So let's add (or, "append") 3 exclamation points to the end of each green span!!!
- Let's use some Spanish too. For spans of class blue, remove the blue class and add an azul class. You can do this in one line by chaining the functions together. use `addClass` and `removeClass`
- I always liked Skittles but not the purple ones. So purple has got to go. Let's give it a classy exit and fade it out over 1 second (1000 milliseconds). Warn the user when a purple span tag is gone.

purple orange green blue blue red orange blue yellow yellow red purple red orange orange purple green purple orange orange green green red blue yellow purple yellow green yellow orange red red orange orange blue orange blue yellow yellow red green green orange purple orange orange red purple purple orange purple yellow green blue blue purple red orange orange red purple purple yellow purple red blue green purple purple red purple red orange red red blue purple red blue red red green purple orange blue purple red yellow red purple green purple blue green yellow blue red blue blue red orange green red orange red orange orange orange blue blue green yellow green green blue blue orange red green red orange yellow green red orange blue red yellow red red blue orange yellow red green blue purple red

AJAX : Inleiding



HoGent

Pag. 29

AJAX: Inleiding

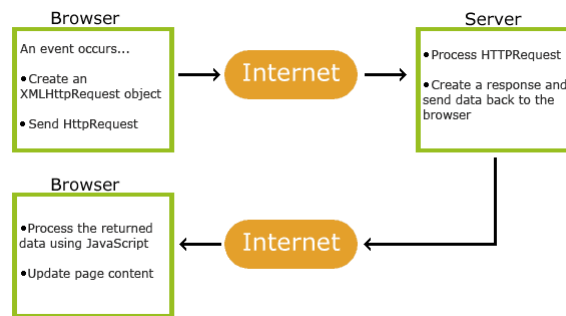
- ▶ Ajax staat voor Asynchroon Javascript en Xml.
- ▶ Ajax is geen nieuwe programmeertaal maar een combinatie van bestaande technologieën:
 - Javascript, CSS, DOM, XML
- ▶ Maakt webpagina's dynamischer, met een betere user experience, door achter de schermen informatie met de server uit te wisselen.
 - Asynchronous
 - JavaScript
 - Javascript wordt gebruikt om het request naar de server te sturen en wanneer het resultaat is binnen gekomen de huidige pagina aan te passen. Maakt eveneens gebruik van CSS en DOM.
 - XML
 - De data die je van de server ontvangt zal vaak een stukje XML zijn dat je eenvoudig kan manipuleren met Javascript. Een andere mogelijkheid is JSON.

HoGent

Pag. 30

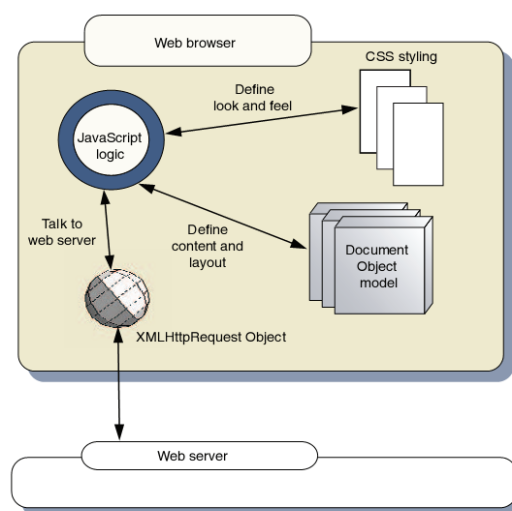
AJAX : Inleiding

► Werking



AJAX : Inleiding

- Ajax is gebaseerd op standaarden
 - JavaScript en DOM
 - CSS
 - XMLHttpRequest object voor communicatie met de server
 - XML of JSON
 - Is browser en platform onafhankelijk
 - Populair geworden In 2005 : Google Suggest



AJAX : XmlHttpRequest

► XmlHttpRequest object



AJAX : XmlHttpRequest

► XmlHttpRequest object

- Ontworpen door Microsoft als ActiveX Object in IE5 voor gebruik in Outlook Web Access Met Exchange Server 2000
- Later omgezet naar Javascript Object
- Verstuurt aanvraag naar server, wacht op antwoord en ontvangt antwoord
- Verschillende implementatie in verschillende browsers
 - In IE7+, Firefox, Safari, Google Chrome : native built-in object
 - `var req = new XmlHttpRequest();`
 - In IE5 en 6 : ActiveX;
 - `var req = new ActiveXObject("Microsoft.XMLHTTP")`
- Maar ongeveer dezelfde properties en methodes
- Standaardisatie : Release Candidate opgesteld door W3C : <http://www.w3.org/TR/XMLHttpRequest/>

AJAX : XmlHttpRequest

- Stap 1 : Aanmaken van XmlHttpRequest object

```
function getXmlHttpRequest() {  
    var xhr;  
    //check for IE implementation(s)  
    if (typeof XMLHttpRequest != 'undefined') {  
        try {  
            xhr = new XMLHttpRequest("Msxml2.XMLHTTP");  
        } catch (e) {  
            xhr = new XMLHttpRequest("Microsoft.XMLHTTP");  
        }  
    } else if (XMLHttpRequest) {  
        //this works for Firefox, Safari, Opera  
        xhr = new XMLHttpRequest();  
    } else {  
        alert("Sorry, your browser doesn't support ajax");  
    }  
    return xhr;  
}
```

AJAX : XmlHttpRequest

- Stap 2 : Versturen van een request naar server
 - **open(method, url, async)** : specificeert type van request, de url, en wijze van versturen
 - **method** : mogelijke waarden : GET, POST, PUT, DELETE of HEAD.
 - Get of Post?
 - Get is eenvoudiger en sneller (data wordt meegestuurd met querystring)
 - Gebruik Post indien
 - Data op server wordt gewijzigd
 - Veel data naar de server moet gestuurd worden (geen limiet)
 - Versturen van user input met ongekende karakters. Post is veiliger
 - **async** : boolean, true als het script asynchroon uitvoert. Dit betekent dat het script verder loopt zonder te wachten op het antwoord. Met het onreadystatechange event weet je dan wanneer het klaar is.
 - **send(string)** : stuurt het request door naar de server.
 - De string wordt enkel gebruikt voor POST requests (= tekst of xml)

AJAX : XmlHttpRequest

- Stap 2 : Versturen van een request naar server (vervolg)
 - Als de boodschap asynchroon verstuurd wordt, dien je in het **onreadystatechange** eventhandler een functie te specificeren.
 - Deze functie laat aan de client toe om de current state van een asynchrone call te monitoren. Als de status van het XmlHttpRequest call process verandert, wordt dit event getriggerd en wordt de toegekende functie uitgevoerd.

```
function getMessage() {  
    //get XmlHttpRequest object  
    var xhr = getXmlHttpRequest();  
  
    //setup request  
    xhr.open("GET", "Home/GetMessage", true)  
  
    //setup callback function  
    xhr.onreadystatechange = function () {  
        //to be specified  
    };  
  
    //fire request  
    xhr.send(null);  
}
```

```
public ActionResult GetMessage()  
{  
    return Content("Hello world");  
}
```

AJAX : XmlHttpRequest

- Stap 3 : Monitoren status request en als antwoord ontvangen is wordt die verwerkt in de pagina
 - **onreadystatechange** event van xmlHttpRequest wordt afgevuurd telkens de status (readyState) van asynchrone call wijzigt.
 - **readyState** property : de status van xmlHttpRequest Object
 - 0 = Uninitialized, 1 = Loading, 2 = Sent, 3 = Receiving, **4 = Complete (antwoord volledig ontvangen)**
 - **status** property : toont status van antwoord (beschikbaar vanaf readyState = 3).
 - 200 : operatie is gelukt
 - 404 : zou bvb betekenen dat de service niet gevonden werd.
 - 500 : internal server error

AJAX : XmlHttpRequest

- Stap 3 : Monitoren status request en als antwoord ontvangen is wordt die verwerkt in de pagina (vervolg)
- Als readyState = 4 en de status = 200 dan kan het antwoord in de pagina worden verwerkt
- Het antwoord :
 - responseText :
get response data als string
 - responseXML :
get response data als xml data

```
function getMessage() {  
    //get xmlhttprequest object  
    var xhr = getXmlHttpRequest();  
  
    //setup request  
    xhr.open("GET", "/Home/GetMessage", true);  
  
    //setup callback function  
    xhr.onreadystatechange = function () {  
        //readyState 4 : received answer from server  
        if (xhr.readyState == 4 && xhr.status == 200)  
            //add result to page  
            $('#result').text(xhr.responseText);  
    };  
  
    //fire request  
    xhr.send(null);  
}
```

AJAX : JSON

► JSON (JavaScript Object Notation)

XML

```
<planten>  
<plant>  
    <naam>Roos</naam>  
    <prijs>1</prijs>  
</plant>  
...  
</planten>
```

Meer info op : www.json.org

JSON

```
[  
    {  
        "naam": "Roos",  
        "prijs": 2  
    },  
    {  
        "naam": "Tulp",  
        "prijs": 1  
    }  
]
```

AJAX : JSON

► JSON

- Structuur bevat volgende elementen
 - Object: begint en eindigt met accolades {} .
 - Object members: naam waarde paren gescheiden door een dubbele punt. Meerdere object members worden van mekaar gescheiden door een komma.
 - Array: begint en eindigt met vierkante haakjes [], en de overeenkomstige waarden worden gescheiden door een komma. Een array is een verzameling van objecten
 - Values: mogelijkheden : string, number, object, array of null
 - Strings: staan tussen " "

```
[  
  {"naam": "Roos", "prijs": 2},  
  {"naam": "Tulp", "prijs": 1}  
]
```

AJAX : JSON

► AJAX en JSON

◦ Voordelen

- Eenvoudige verwerking met JavaScript : eval functie die text terug omzet in een Javascript object
- Sneller

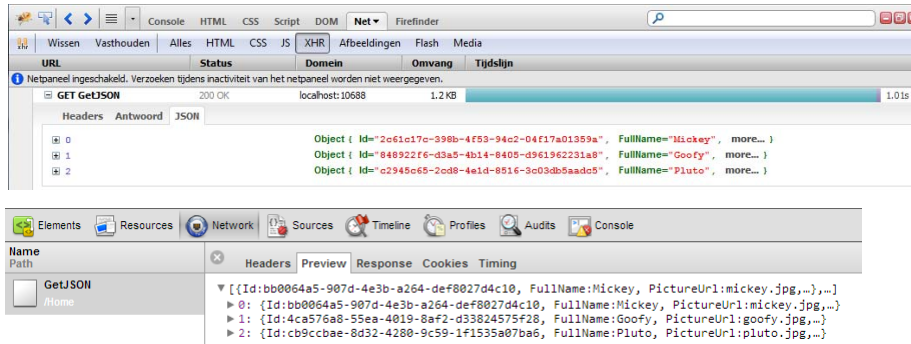
```
public ActionResult GetJSON()  
{  
    IEnumerable<Speaker> speakers = speakerRepository.FindAll();  
    return Json(speakers, JsonRequestBehavior.AllowGet);  
}
```

```
function getJSONMessage() {  
    var xhr = getXmlHttpRequest();  
    xhr.open("GET", "/Home/GetJSON", true);  
    xhr.onreadystatechange = function () {  
        if (xhr.readyState == 4 && xhr.status == 200) {  
            var speakers = eval(xhr.responseText);  
            $('#result').text(speakers[0].FullName);  
        }  
    };  
    xhr.send(null);  
}
```

AJAX : JSON

► AJAX en JSON

- In FireBug/Chrome developer tools kan je XMLHttpRequest object bekijken. En dus ook de json die als antwoord wordt teruggestuurd



HoGent

Ajax en jQuery Pag. 43

Ajax : jQuery

► .get

- Voert AJAX Http GET request uit

```
$(selector).get(url,data,success(response,status,xhr),dataType)
```

Parameter	Description
url	Required. Specifies the url to send the request to
data	Optional. Specifies data to send to the server along with the request
success(response,status,xhr)	Optional. Specifies the function to run if the request succeeds Additional parameters: <ul style="list-style-type: none">• response - contains the result data from the request• status - contains the status of the request ("success", "notmodified", "error", "timeout", or "parsererror")• xhr - contains the XMLHttpRequest object
dataType	Optional. Specifies the data type expected of the server response. By default jQuery performs an automatic guess. Possible types: <ul style="list-style-type: none">• "xml" - An XML document• "html" - HTML as plain text• "text" - A plain text string• "script" - Runs the response as JavaScript, and returns it as plain text• "json" - Runs the response as JSON, and returns a JavaScript object• "jsonp" - Loads in a JSON block using JSONP. Will add an "?" callback="?" to the URL to specify the callback

HoGer

Pag. 44

Ajax : jQuery

► .get

◦ Voorbeeld

```
<p><strong>Click the button to issue an ajax request to the server: </strong><br />
<button type="button" id="GetMessage" >Get the Message</button>
</p>
<div id="result"></div>
```

```
$(document).ready(function () {
    $("#GetMessage").click(function () {
        $.get("/Home/GetMessage", null, function (data) {
            $("#result").html(data);
        });
    });
});
```

◦ De Controller

```
public ActionResult GetMessage()
{
    return Content("Hello world");
}
```

Ajax : jQuery

► .getJSON

◦ Voert AJAX Http GET request uit, en krijgt JSON als antwoord

```
$(selector).getJSON(url,data,success(data,status,xhr))
```

Parameter	Description
url	Required. Specifies the url to send the request to
data	Optional. Specifies data to be sent to the server
success(data,status,xhr)	Optional. Specifies the function to run if the request succeeds Additional parameters: <ul style="list-style-type: none">• data - contains the data returned from the server.• status - contains a string containing request status ("success", "notmodified", "error", "timeout", or "parsererror").• xhr - contains the XMLHttpRequest object

```
$("#GetJson").click(function () {
    $.getJSON("/Home/GetJson", null, function (data) {
        $("#result").html("");
        $.each(data, function (key, val) {
            $("#result").append(val.FullName + "<br/>");
        });
    });
});
```

Ajax : jQuery

► .getJSON

- De Controller

```
public ActionResult GetJSON()
{
    IEnumerable<Speaker> speakers = speakerRepository.FindAll();
    return Json(speakers, JsonRequestBehavior.AllowGet);
}
```

- JsonRequestBehavior.AllowGet moet er staan als je json via get ophaalt anders runtime fout (owv security): *"This request has been blocked because sensitive information could be disclosed to third party web sites when this is used in a GET request. To allow GET requests, set JsonRequestBehavior to AllowGet."*

Ajax : jQuery

► .ajax

- Voert ajax request uit (zowel get als post mogelijk)
- Syntax : \$.ajax([settings])
- \$.ajax() retourneert een jQuery XMLHttpRequest (jqXHR) object (een superset van browser's native XMLHttpRequest object.). Dit bevat de methodes
 - jqXHR.done(function(data, textStatus, jqXHR) {});
 - jqXHR.fail(function(jqXHR, textStatus, errorThrown) {});
 - jqXHR.always(function(data|jqXHR, textStatus, jqXHR|errorThrown) { }); jqXHR.then(function(data, textStatus, jqXHR) {}, function(jqXHR, textStatus, errorThrown) {});

Ajax : jQuery

► Voorbeeld

```
$("#AjaxJson").click(function() {  
    $.ajax({ url: "/jQuery/GetJson" })  
    .done(function(data) {  
        $("#result").html("");  
        $.each(data, function(key, val) {  
            $("#result").append(val.FullName + "<br/>");  
        });  
    })  
    .fail(function() { alert("error"); })  
    .always(function() { alert("complete"); });  
});
```

Ajax : jQuery

- Tot nu toe staat de javascript in de html file
 - <-> unobtrusive javascript
- Oplossing :
 - Maak een aparte script file aan
 - In de Scripts folder > maak folder CustomScripts > rechtsklik > Add > new item > jScript file > jQueryAjaxScripts.js
 - Plaats script gedeelte uit Views > jQuery > Ajax in dit bestand (verwijder <script> tag)



Ajax : jQuery

- Voeg in View Ajax.cshtml referentie toe naar scriptfile (drag en drop script file)

```
@section Scripts
{
<script type="text/javascript" src="~/Scripts/CustomScripts/jqueryAjaxScripts.js"></script>
}
```

- Vervang src door

```
@section Scripts
{
<script type="text/javascript" src="@Url.Content("~/Scripts/CustomScripts/jqueryAjaxScripts.js")"></script>
}
```

- ~ : vanaf de virtuele root (je kan later de view verplaatsen, zonder dit te moeten aanpassen)
- OF Maak gebruik van de bundles!

Ajax : ASP.NET MVC Unobtrusive Ajax

► Unobtrusive Javascript

- markup scheiden van javascript. Maakt gebruik van script files
- Laat progressive enhancement toe.

http://en.wikipedia.org/wiki/Progressive_enhancement

Progressive enhancement is a strategy for [web design](#) that emphasizes accessibility, [semantic HTML markup](#), and external stylesheet and scripting technologies. Progressive enhancement uses web technologies in a layered fashion that allows everyone to access the basic content and functionality of a web page, using any browser or Internet connection, while also providing an enhanced version of the page to those with more advanced browser software or better bandwidth.

Core principles

Progressive Enhancement consists of the following core principles:

- basic content should be accessible to all web browsers
- basic functionality should be accessible to all web browsers
- sparse, semantic markup contains all content
- enhanced layout is provided by externally linked CSS
- enhanced behavior is provided by [unobtrusive](#), externally linked JavaScript
- end-user web browser preferences are respected

ASP.NET MVC Unobtrusive Ajax

► Ajax Helpers

- Voor formulieren : `Ajax.BeginForm`
 - Vergelijkbaar met `Html.BeginForm` maar nu wordt request asynchroon verstuurd EN wordt de nieuwe inhoud geïnjecteerd in de html pagina
- Voor links : `Ajax.ActionLink`
- Beide hebben een parameter `AjaxOptions`

```
public static System.Web.Mvc.Html.MvcForm BeginForm(this System.Web.Mvc.AjaxHelper ajaxHelper, string actionName,
string controllerName, object routeValues, System.Web.Mvc.Ajax.AjaxOptions ajaxOptions, object htmlAttributes)
    Member of System.Web.Mvc.Ajax.AjaxExtensions
```

Summary:
Writes an opening <form> tag to the response.

Parameters:
ajaxHelper: The AJAX helper.
actionName: The name of the action method that will handle the request.
controllerName: The name of the controller.
routeValues: An object that contains the parameters for a route. The parameters are retrieved through reflection by examining the properties of the object. This object is typically created by using object initializer syntax.
ajaxOptions: An object that provides options for the asynchronous request.
htmlAttributes: An object that contains the HTML attributes to set for the element.

Returns:
An opening <form> tag.

HoGe

55

ASP.NET MVC Unobtrusive Ajax

► Ajax Helpers

- parameter `AjaxOptions`, met de properties

Name	Description
Confirm	Gets or sets the message to display in a confirmation window before a request is submitted.
HttpMethod	Gets or sets the HTTP request method ("Get" or "Post").
InsertionMode	Gets or sets the mode that specifies how to insert the response into the target DOM element.
LoadingElementId	Gets or sets the id attribute of an HTML element that is displayed while the Ajax function is loading.
OnBegin	Gets or sets the name of the JavaScript function to call immediately before the page is updated.
OnComplete	Gets or sets the JavaScript function to call when response data has been instantiated but before the page is updated.
OnFailure	Gets or sets the JavaScript function to call if the page update fails.
OnSuccess	Gets or sets the JavaScript function to call after the page is successfully updated.
UpdateTargetId	Gets or sets the ID of the DOM element to update by using the response from the server.
Url	Gets or sets the URL to make the request to.

HoGent

Pag. 56

ASP.NET MVC Unobtrusive Ajax

► Ajax.ActionLink() : ophalen van **tekst**

```
@Ajax.ActionLink("get message Hello World from Ajax MVC", "GetErrorMessage", "GetErrorPageJavascriptDisabled",  
    new AjaxOptions {  
        Url = Url.Action("GetMessage"),  
        HttpMethod = "GET",  
        InsertionMode = InsertionMode.Replace,  
        UpdateTargetId = "result"  
    })
```

Tekst op hyperlink

Wordt aangeroepen als browser
geen javascript toelaat

- Url : actie die aangeroepen wordt door ajax
- HttpMethod : via de Get
- Het resultaat van de actiemethode wordt geplaatst in het div element met id result (UpdateTargetId). De inhoud wordt vervangen (InsertionMode)

ASP.NET MVC Unobtrusive Ajax

► Ajax.ActionLink() : ophalen van **json**

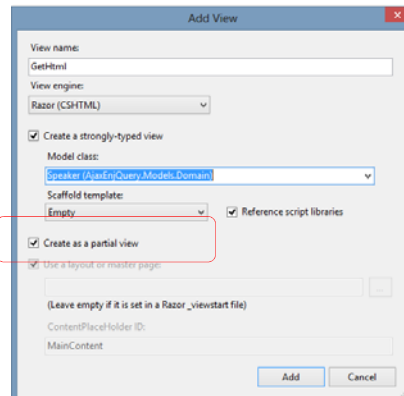
```
@Ajax.ActionLink("get json from Ajax MVC", "GetErrorMessage", "GetErrorPageJavascriptDisabled",  
    new AjaxOptions {  
        HttpMethod = "GET",  
        Url = Url.Action("GetJson"),  
        OnSuccess = "handleSuccess"  
    })
```

- Extra functie nodig om array van json objecten in div result te plaatsen (hier kan je niet werken met UpdateTargetId)

```
function handleSuccess(data) {  
    $("#result").html("");  
    $.each(data, function (key, val) {  
        $("#result").append(val.FullName + "<br/>");  
    });  
};
```

ASP.NET MVC Unobtrusive Ajax

- ▶ Ajax.ActionLink() : ophalen van **Html**
 - Werkt met een partial View
 - = een onderdeelje van een View pagina
 - Aanmaken van partial view



HoGent

Pag. 59

ASP.NET MVC Unobtrusive Ajax

- ▶ Ajax.ActionLink() : ophalen van **Html**
 - Pas de Partial View verder aan
 - Verwijder de CDU links
 - Vervang in View de image

```
@model IEnumerable<AjaxEnjQuery.Models.Domain.Speaker>

<table>
<tr>
<th>FullName</th>
<th>PictureUrl</th>
<th>Bio</th>
</tr>

@foreach (var item in Model) {
<tr>
<td>
@Html.DisplayFor(modelItem => item.FullName)
</td>
<td>

</td>
<td>
@Html.DisplayFor(modelItem => item.Bio)
</td>
</tr>
}
</table>
```

HoGent

Pag. 60

ASP.NET MVC Unobtrusive Ajax

► Ajax.ActionLink() : ophalen van **Html**

- De Controller

```
public ActionResult GetHtml()
{
    IEnumerable<Speaker> speakers = speakerRepository.FindAll();
    return PartialView(speakers);
}
```

- De View

```
@Ajax.ActionLink("get html from Ajax MVC", "GetErrorPageJavascriptDisabled",
new AjaxOptions {
    Url = Url.Action("GetHtml"),
    HttpMethod = "GET",
    InsertionMode = InsertionMode.Replace,
    UpdateTargetId = "result"
})
```

HoGent

Pag. 61

Hijax

Hijax

Hijax is the name for the application of **progressive enhancement** to **AJAX**. This allows the use of AJAX to increase **usability** and responsiveness without harming **accessibility** and can, as a side-effect, increase server side code quality. It is called Hijax because there is a standard HTML web page underneath but its normal communication with the server is *hijacked* and redirected to create the AJAX effects.

The process

Implementing Hijax involves two steps (taken from [1] [🔗](#)):

1. First, build an old-fashioned website that uses hyperlinks and forms to pass information to the server. The server returns whole new pages with each request.
2. Now, use JavaScript to intercept those links and form submissions and pass the information via XMLHttpRequest instead. You can then select which parts of the page need to be updated instead of updating the whole page.

HoGent

Pag. 62

Hijax

- ▶ Veel voorkomende patronen
 - Een pagina bevat een formulier. Als formulier is verstuurd wordt deel van de pagina aangepast
 - Voorbeelden
 - Blog Comments
 - Sessions voorbeeld
 - Klikken op een link, retourneert dezelfde pagina, maar een deel van de pagina is aangepast
 - Voorbeelden
 - product ratings
 - Zoekresultaten : past een deel van de pagina aan
 - Pagineren
 - Speakers voorbeeld

Hijax

- ▶ Zorg dat de applicatie werkt zonder Ajax
 - De pagina bevat momenteel de lijst van sessions. Bij elke update (Remove, Add) dient eigenlijk enkel deze lijst te worden aangepast => Ajax achteraf toevoegen

```
<table id="sessions">
  <tr>
    <th>Title</th>
    <th>Description</th>
    <th>Level</th>
  </tr>
  @if(Model.Count() == 0) {
    <tr> <td colspan="4" align="center">There are no sessions. Add one below!</td></tr>
  }
  @foreach (var item in Model) {
    <tr>
      <td>@Html.DisplayFor(modelItem => item.Title)</td>
      <td>@Html.DisplayFor(modelItem => item.Description)</td>
      <td>@Html.DisplayFor(modelItem => item.Level)</td>
      <td>
        @using (Html.BeginForm("Remove"))
        {
          <input type="hidden" name="id" value="@item.id" />
          <input type="submit" value="remove" />
        }
      </td>
    </tr>
  }
</table>
```


Hijax

- ▶ Door het toepassen van Ajax en partial views kunnen we de form post hijaxen en vervangen door Ajax request.
 - 1. Vervang het onderdeel die moet worden aangepast bij een update door een partial view.
 - Rechtsklik in de action methode > Add View
 - Noem deze SessionList. Maak een Strongly Typed Partial View aan voor de domeinklasse Session. Kies voor empty.

Hijax

- Voeg de code uit index.cshtml voor het tonen van de tabel toe aan deze partial view

```
@model IEnumerable<AjaxEnjQuery.Models.Domain.Session>
<?
@Html.ActionLink("Create New", "Create")
</p>
<table id="sessions">
<tr>
<th>Title</th>
<th>Description</th>
<th>Level</th>
</tr>
</tr>
@if (Model.Count() == 0) {
<tr> <td colspan="4" align="center">There are no sessions. Add one below!</td></tr>
}
@foreach (var item in Model) {
<tr>
<td>@Html.DisplayFor(modelItem => item.Title)</td>
<td>@Html.DisplayFor(modelItem => item.Description)</td>
<td>@Html.DisplayFor(modelItem => item.Level)</td>
<td>
@using (Html.BeginForm("remove", "hijax", FormMethod.Post))
{
<input type="hidden" name="id" value="@item.Id" />
<input type="submit" value="remove" />
}
}
</td>
```

Hijax

- In Index.cshtml roep je een partial view als volgt op

```
<div id="session-list">  
@Html.Partial("SessionList", Model)  
</div>
```

- Dit wordt uitgevoerd als de View gerendered wordt (dus bij de eerste request, of indien de daaropvolgende requests geen ajax requests zijn)

Hijax

- 2. De controller moet de partial view genereren ipv de volledige pagina igv Ajax request
 - De Index methode in Controller wordt

```
public ActionResult Index()  
{  
    var sessions = repository.FindAll();  
    if (Request.IsAjaxRequest())  
        return PartialView("SessionList", sessions);  
    return View(sessions);  
}
```

- Dit zal igv een AjaxRequest de HTML van de partial View retourneren

Hijax

- De Add methode in de Controller wordt

```
[HttpPost]
public ActionResult Add(Session session)
{
    repository.Add(session);
    if (Request.IsAjaxRequest())
        return Index();
    return RedirectToAction("index");
}
```

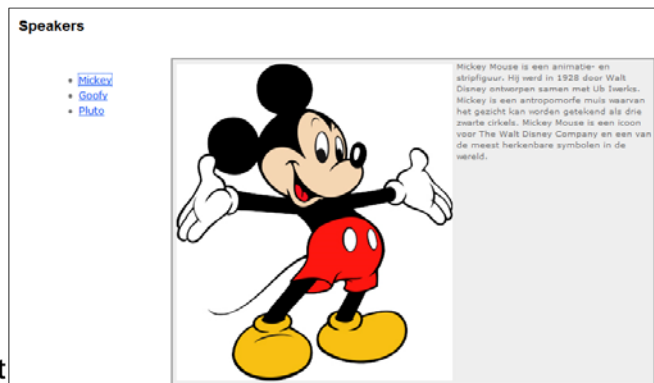
- Merk op : return Index() igv Ajax : return de HTML van partial view!
- Je zou in de Add natuurlijk ook de PartialView kunnen teruggeven ipv return Index op te roepen, dan dien je de Index methode niet aan te passen
- Testen : Maak gebruik van ModelBinder

Hijax

- ▶ Oefening : pas Controller/View verder aan zodat ook voor de Remove met Ajax gewerkt wordt

Hijax : json

- ▶ Bekijk SpeakerController
 - Ook hier voegen we Ajax toe aan pagina. Hier wordt gebruik gemaakt van JSON (kan ook met partial views). Ipv van naar een aparte detail pagina te gaan, wordt i.g.v. ajax de detail in pagina zelf weergegeven



Hijax : json

- ▶ SpeakerController
 - Controller :

```
public ActionResult Details(Guid id, string format)
{
    var speaker = repository.FindBy(id);

    if (Request.IsAjaxRequest())
        return Json(speaker, JsonRequestBehavior.AllowGet);

    return View(speaker);
}
```

Hijax : json

► SpeakerController

◦ View :

```
@Scripts.Render("~/bundles/jqueryval")

function loadSpeakerDetails(speaker) {
    var box = $("#speaker");
    box.html("");
    $('<img/>')
        .attr("src", @Url.Content("~/Images/") + speaker.PictureUrl)
        .attr("alt", "pic")
        .attr("style", "float:left;margin:5px")
        .appendTo(box);
    $('<span/>')
        .attr("style", "font-size: .8em")
        .html(speaker.Bio).appendTo(box);
    $('<br style="clear:both" />').appendTo(box);
    $("#speaker").show();
}

<li>@Ajax.ActionLink(item.FullName, "Details", new { id = item.Id },
    new AjaxOptions
    {
        Url = Url.Action("Details", new { id = item.Id }),
        OnSuccess = "loadSpeakerDetails"
    })
</li>
```

Pag. 73

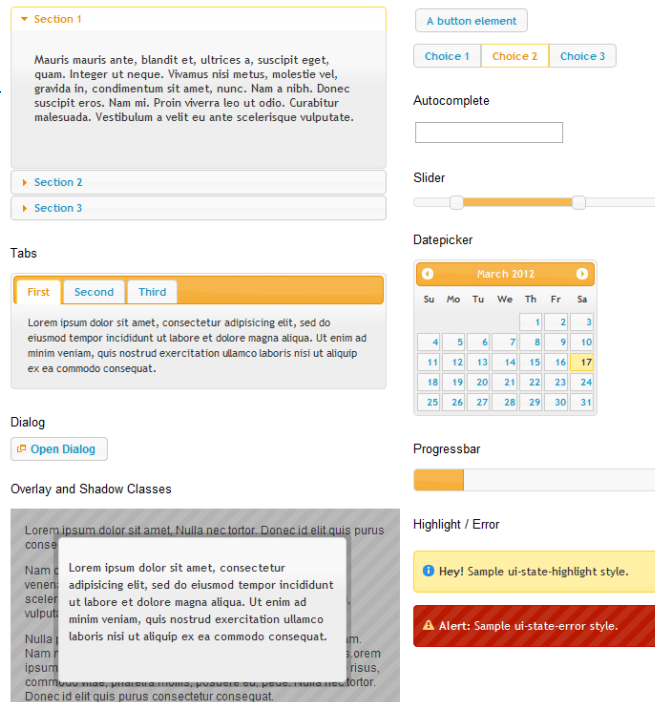
jQuery UI

► jQuery UI



- jQuery plugin met effects en widgets
- Meer op <http://jqueryui.com/>
- Reeds onderdeel van MVC project
- Zie scripts folder voor de scripts jquery-ui.(min).js
- De bijhorende css en beeldjes in Content folder > Themes/Base. Bevat een theme (een theme bestaat uit een css en beeldjes)
- Je kan nieuwe themes downloaden van <http://jqueryui.com/themeroller/>. Ga naar de Gallery
 - De zip bevat oa. css folder met css en de beeldjes. Die moet je aan MVC project toevoegen

► Widgets :



HoGent

g. 75

jQuery UI

► jQuery UI

- DatePicker (nog niet in alle browsers bij input type="date")

- Voeg verwijzing toe naar

- Script
- Stylesheet

@section Scripts

```
@Scripts.Render("~/bundles/jqueryui")
@Styles.Render("~/Content/themes/base/css")
@Scripts.Render("~/bundles/jqueryval")
```


Je kan ook enkel een referentie toevoegen naar datapicker css ipv alle ui css

- Nog een scriptje toevoegen zodat de DatePicker geactiveerd wordt, voor alle browsers die input type date niet ondersteunen

```
<script>
if (!Modernizr.inputtypes.date) {
    $('input[type=date]').datepicker({ dateFormat: "yy-mm-dd" });
}
</script>
```

HoGent

Pag. 76


Modernizr
 FRONT-END DEVELOPMENT DONE RIGHT

[DOWNLOAD](#)
[DOCUMENTATION](#)
[RESOURCES](#)
[NEWS](#)

“Modernizr is an essential part of my toolkit of files.”
 — Andy Clarke, *Stuff & Nonsense*

Download Modernizr 2.5.3

Use the commented, uncompressed Development version to develop with and learn from.

Then, dive into the Production build tool and pick just the tests you need!

[View documentation](#)

DEVELOPMENT
 Uncompressed, 42 Kb

PRODUCTION
 Configure Your Build

Why use Modernizr?

Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites.

Taking advantage of the new capabilities of HTML5 and CSS3 can mean sacrificing control over the experience in older browsers. Modernizr 2 is your starting point for making the best websites and applications that work exactly right no matter what browser or device your visitors use.

Thanks to the new Media Query tests and built-in [YepNope.js](#) micro-library as `Modernizr.load()`, you can now combine feature detection with media queries and conditional resource loading. That gives you the power and flexibility to optimize for every circumstance.

Check out the [full list of features](#) that Modernizr detects, or learn more about [conditional resource loading with Modernizr](#).

Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScript-driven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- [Documentation: Getting started](#)
- [Taking Advantage of HTML5 and CSS3 with Modernizr](#), Faruk Ateş
- [How to use Modernizr](#), Inayati de León
- [Modernizr: front-end development done right](#), Ryan Seddon
- [wiki] [The Undetectables: features that cannot be detected](#)
- [wiki] [Cross-browser Polyfills](#)

Also check out our [Resources section](#).

Tip: check our [Modernizr test suite](#) to quickly test your current browser's

jQuery UI

► Modernizr : Feature detection

Aan head gedeelte toevoegen. Creëert een nieuw javascript object Modernizr die een 40tal properties bevat om na te gaan of een bepaalde feaature geïmplementeerd is

```
@Scripts.Render("~/bundles/modernizr")
```

```
<script src="/Scripts/modernizr-2.5.3.js"></script>
```

Modernizr.

video	TRUE
audio	TRUE
canvas	TRUE
localStorage	TRUE
webworkers	TRUE
websockets	TRUE
...	...

Die properties kan je dan opvragen

```
if (Modernizr.canvas) {
    // let's draw some shapes!
} else {
    // no native canvas support available :(
}
```

jQuery UI

► jQuery UI

- Aanmaken van EditorTemplate, zodat alle properties met annotation [DataType(DataType.Date)] een jQuery UI datepicker gebruiken
- Maak een partial view aan in de folder Views > Shared > EditorTemplates. Noem het Date.cshtml

```
@model DateTime  
@Html.TextBox("", Model.ToString("dd/MM/yyyy"), new { @class = "datepicker" })
```

- Opm : wens je DisplayTemplate aanmaken, maak dan folder DisplayTemplates aan

jQuery UI

- ### ► Maak nog een script file in Scripts/CustomScripts, EditorHookup.js

```
$(function () {  
    $('.datepicker').datepicker({ dateFormat: "dd/mm/yy" });  
})
```

- ### ► In _layout.cshtml of in de view (of in bundle) voeg je toe

```
@Scripts.Render("~/bundles/jqueryui")  
@Styles.Render("~/Content/themes/base/css")  
<script src="@Url.Content("~/Scripts/CustomScripts/EditorHookup.js")"></script>
```

- ### ► In klasse Event verwijder je DisplayFormat

jQuery UI en Plugins

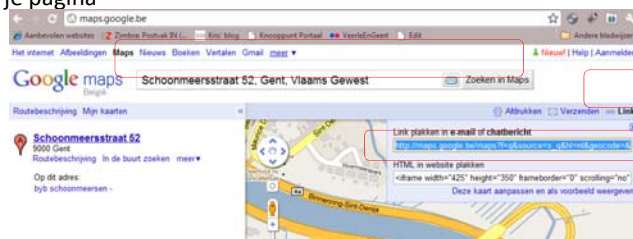
► jQuery Plugin Repository

- <http://plugins.jquery.com/>

- | | |
|------------------------------|-----------------------------|
| ▪ AJAX (85) | ▪ Media (40) |
| ▪ Animation and Effects (94) | ▪ Menues (39) |
| ▪ Browser Tweaks (36) | ▪ Metaplugin (15) |
| ▪ Data (51) | ▪ Navigation (64) |
| ▪ DOM (58) | ▪ Tables (33) |
| ▪ Drag-and-Drop (11) | ▪ User Interface (231) |
| ▪ Events (50) | ▪ Utilities (104) |
| ▪ Forms (123) | ▪ Widgets (81) |
| ▪ Integration (24) | ▪ Windows and Overlays (22) |
| ▪ JavaScript (53) | |
| ▪ jQuery Extensions (103) | |
| ▪ Layout (67) | |

Ajax en rest api's

- Heel wat websites bieden rest api's aan die je via ajax kan aanspreken
 - Google, Amazon,...
 - Meestal vind je op de site voldoende uitleg over de rest api
 - Vb google maps als onderdeel van de site's about.aspx pagina
 - Ga naar maps.google.com. Zoek naar straat. Klik Link. Kopieer de HTML in je pagina



- Eigen Rest api aanmaken : 3de jaar

Oefening

► SportsStore

- Voeg een Zoek toe aan de Store
- Gebruik Ajax (Hijax) voor toevoegen (en verwijderen) van een product aan een cart
- Voeg een DatePicker toe voor via EditorTemplate
- Voeg in Cart ook de mogelijkheid toe om de aantallen aan te passen, en een knop Update
- Voeg een NavigatieController toe. De Index methode retourneert een partial view met de categorieën. De partial View bevat een menu met de categorieën. Toon de menu op alle pagina's (_layout aanpassen, aside toevoegen). Pas de Store aan zodat enkel de producten van de gekozen categorie getoond worden. Als geen categorie gekozen, toon dan alle producten

Referenties

- www.w3schools.com : ajax en jQuery