

SAE MAGAZINE

2008 | 2

The Magazine of SAE Institute and SAE ALUMNI Association



Get your Degree from the
new SAE Oxford – It's for Life

:: Preview SAE ALUMNI CONVENTION 2008 in Berlin

:: PEOPLE & BUSINESS

- :: Interview with Engineer and Producer John Agnello
- :: Interview with Andreas Meyer, Founder of make.tv

:: EVENTS & ACTIVITIES

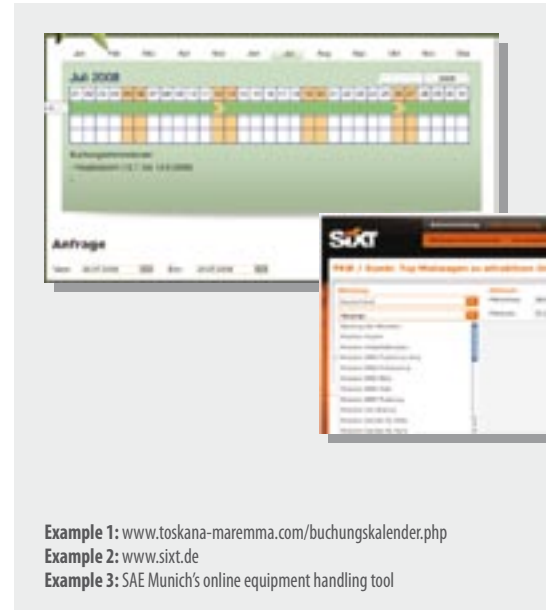
- :: Tom Misner's Rock Musical "The Madman"
- :: SAE ALUMNI Awards 2008
- :: Meet the Masters 2008

:: PRODUCTION & KNOW HOW

- :: Audio deconstructed
- :: Dynamic with Ajax
- :: Sample your Room
- :: The Need of Manipulation

Dynamic with Ajax – a key technique for the interactive Internet Web 2.0

After its implementation, some SAE students are searching for the sense behind and the application of Ajax („unsynchronized JavaScript and XML“). Those having been involved in programming websites in the past will be familiar with the old frame sets structure technique. Along with its countless disadvantages, it offered one powerful advantage. Only the frame that needed to be changed had to be reloaded. The rest simply stayed put and gave the impression of a computer program that worked absolutely independently and with no loading time. Ajax exactly takes over this task and offers far more possibilities as well.



Example 1: www.toskana-maremma.com/buchungskalender.php

Example 2: www.sixt.de

Example 3: SAE Munich's online equipment handling tool

Thanks to the “prototype” JavaScript library, which is an ideal Ajax engine, most difficulties are taken care of and the first tests can be started with just a few lines of program code. The object “Ajax” is given the desired parameters and commands regulating what is to be done with the data received. Note: Don't forget to integrate “prototype” into the page.

Code example 1

The variable name (myAjax) is freely definable and can be used as often as desired because it is used within a function. The PHP file called (getdetails.php) can just as easily be an html or XML document. The “method” refers to the sending characteristic of the document, not how the data is returned. “onComplete” defines the function, which processes the returned data. It is important that no parentheses appear after the function so that it only executes after all data has been returned.

As soon as the answer comes back from the server, the (showDetails) function executes. The XML HTTP Request object, which is filtered out of the Ajax object by the “responseText” characteristic, is automatically passed on as a parameter.

Code example 2

In this function, both types commonly used to look at an element are used. The shorthand used in “prototype” \$(,target') and the normal JavaScript version “getElementById(,target')”. Note: All existing data in the target, which can be any html element such as div, span, p, td, etc., will be overwritten.

It is normally not sufficient to reload regular statistical data. As a rule, the desired effect is a dynamically created response to a user input. To achieve that, additional parameters need to be passed on to the Ajax request using the GET or POST methods.

Code example 3

Here, individually defined parameters are passed on. These can also be filled by variables.

Code example 4

In this example, all variables are read from the form (“ausleihen”) and passed on.

Code example 1

```
function getDetails(){
    var myAjax = new Ajax.Request(
        „getdetails.php“, { method:'get', onComplete: showDetails }
    );
}
```

Code example 3

```
var myAjax = new Ajax.Request(
    „getdetails.php“,
    { method:'get', parameters:„a=1&b=2“, onComplete: showDetails }
);
```

Code example 5

```
var myAjax = new Ajax.Request(
    „getdetails.php“,
    {
        method:'get',
        parameters: Form.serializeElements( $(,ausleihen').getInputs(,text',
        ,student')),
        onComplete: showDetails
    }
);
```

Code example 5

Here only one single input field (student) is read from the form (ausleihen) and its content passed on.

Dynamik mit Ajax – eine Schlüsseltechnik für das interaktive Internet Web 2.0

Beispiel 1: www.toscana-maremma.com/buchungskalender.php
 Beispiel 2: www.sixt.de
 Beispiel 3: Verleih- und Reservierungs-Tool der SAE



Einige der SAE Studenten grübeln nach der Einführung über den Sinn und die Umsetzung von Ajax („Asynchronous JavaScript and XML“). Wer sich früher schon mit der Programmierung von Webseiten beschäftigt hat, kennt noch die alte Strukturierungs-Technik der Frame-Sets. Diese boten neben ihren unzähligen Nachteilen einen großen Vorteil. Es wurde immer nur der Frame neu geladen, der geändert werden sollte. Der Rest blieb einfach stehen und vermittelte den Eindruck wie ein Computerprogramm absolut eigenständig und ohne Ladezeiten zu arbeiten. Ajax übernimmt genau diesen Aufgabenbereich und bietet noch wesentlich mehr Möglichkeiten.

Code example 2

```
function showDetails( ajaxRequest ) {
    $(target).innerHTML = ajaxRequest.responseText;
    /*document.getElementById(target).innerHTML = ajaxRequest.responseText;*/
}
```

Code example 4

```
var myAjax = new Ajax.Request(
    „getdetails.php“,
    { method:„get“, parameters: Form.serialize(„ausleihen“),
    onComplete: showDetails }
);
```

Dank der JavaScript-Library „prototype“, die sich ideal als Ajax Engine verwenden lässt, fallen bereits die meisten Schwierigkeiten weg, es können mit wenigen Zeilen Code die ersten Versuche gestartet werden. Hierbei verwendet man das Objekt „Ajax“, übergibt ihm die gewünschten Parameter sowie die Information, was mit den erhaltenen Daten zu geschehen hat. Hinweis: nicht vergessen, „prototype“ auf der Seite einzubinden

Code example 1

Der Variablenname („myAjax“) kann frei gewählt und beliebig oft verwendet werden, da dieser innerhalb einer Funktion eingesetzt wird. Das aufgerufene File („getdetails.php“) kann neben PHP auch ein html oder XML Dokument sein. Die Angabe „method“ betrifft die Sendeeigenschaft an das Dokument, nicht wie die Daten zurückgeliefert werden. „onComplete“ übergibt die Funktion, welche die erhaltenen Daten verarbeiten soll. Wichtig ist, dass keine Klammern nach der Funktion stehen, damit diese erst ausgeführt wird, wenn alle Daten angekommen sind.

Sobald die Antwort vom Server angekommen ist, wird die angegebene Funktion („showDetails“) ausgeführt. Als Parameter wird automatisch das XML HTTP Request Objekt übergeben, welche mit der Eigenschaft „responseText“ des Ajax Objektes ausgelesen wird.

Code example 2

In dieser Funktion sind beide Arten angegeben, die gerne verwendet werden, um ein Element anzusprechen. Die Kurzschreibweise von „prototype“ \$(target)“ und die übliche JavaScript Version „getElementById(target)“. Hinweis: Im angesprochenen Ziel (target), das ein beliebiges html Element wie div, span, p, td usw. sein kann, werden alle vorhandenen Daten überschrieben.

Reguläre, statische Daten nachzuladen ist meist nicht ausreichend. Eine dynamisch erstellte, auf eine Benutzereingabe eingehende Antwort ist in der Regel der gewünschte Effekt. Hierzu ist es erforderlich, dem Ajax Request über die GET oder POST Methode weitere Parameter zu übergeben.

Code example 3

Hier werden individuell bestimmte Parameter übergeben. Diese können auch mit Variablen gefüllt werden.

Code example 4

Hier werden alle Variablen aus dem angesprochenen Formular („ausleihen“) ausgelesen und übergeben.

Code example 5

In diesem Beispiel wird nur ein einzelnes Input-Feld („student“) aus dem Formular („ausleihen“) ausgelesen und dessen Inhalt übergeben.

How are the functions executed?

Numerous websites such as Wikipedia, Selfhtml and others are involved in explaining how Ajax works or is generated. But often, explaining what makes it work is neglected. In order to make the Ajax responses appear interactively on the screen, the function calls need to be built in with much skill.

Several possibilities are available. Here are a few suggestions:

onclick call: when an element is clicked with the mouse

ondblclick call, when an element is double clicked with the mouse

onfocus call, when the cursor is placed in this element, i.e. an input field

onchange call, when the content of this element is changed, i.e. an input field

onblur call, when the currently selected element is exited, i.e. an input field

onselect call, when the element is selected, i.e. an <object> in a select list

Encoding and the German "Umlaut" problem!

When an html or PHP file is reloaded by Ajax, the data is displayed without encoding information. The problem that occurs in the German-speaking world due to the special "Umlaut" characters is a pain, but it can be solved quite easily. Simply include the following code in the first line of the file to be loaded:

```
<?php header('Content-Type: text/html; charset=utf-8');?>
```

The greatest disadvantages.

Websites generated and/or altered with Ajax can only exist „temporarily“ in working memory so they cannot be saved 100% with a bookmark. Any bookmark links to the page as it was in the beginning, before the Ajax changes were made.

The second serious disadvantage is that the browser's „one page back“ button does not react to the Ajax alterations, but ignores them all and skips back an entire website.

Why use Ajax at all?

In the Front-end:

The possibility to load Tooltips (for example www.sixt.de, input field for the city) or to validate entries and load appropriate answers or forms (for example www.toskana-maremma.com/buchungskalender.php, date entry).

In the Back-end:

Content and display changes in program-like applications. There are abundant examples, but, as expected, they are all password protected. An alternative is to have a look at SAE Munich's lending and reservation tool – just talk to the supervisors.

Gregor Müller-Elmau



HIGON[®]
CONNECTORS

Qualitätssteckverbinder für
die Audio- und Videotechnik

Generalvertretung: **SOMMER CABLE GmbH**
Phone +49 (0) 70 82 / 4 91 33-0 · Fax 4 91 33-11
info@sommercable.com · www.sommercable.com

Wie werden die Funktionen aufgerufen?

Zahlreiche Webseiten wie Wikipedia, Selfhtml und andere beschäftigen sich mit der Erklärung, wie Ajax funktioniert oder generiert wird. Doch sehr oft wird vernachlässigt, den Auslöser zu erklären. Damit die Ajax Antworten wirklich interaktiv am Bildschirm erscheinen, ist es erforderlich den Aufruf der Funktionen geschickt einzubauen.

Hierzu bieten sich eine Vielzahl von Möglichkeiten an. Hier ein paar Anregungen:

onclick call, wenn ein Element mit der Maus geklickt wird

ondblclick call, wenn ein Element mit der Maus doppelt geklickt wird

onfocus call, wenn der Cursor in dieses Element gesetzt wird, z.B. ein Input-Feld

onchange call, wenn der Inhalt dieses Elements geändert wird, z.B. ein Input-Feld

onblur call, wenn das aktuell gewählte Element verlassen wird, z.B. ein Input-Feld

onselect call, wenn das Element selektiert wird, z.B. ein <object> in einer Select-Liste

Encoding und das Umlautproblem!

Wenn zum Beispiel ein html oder PHP File per Ajax nachgeladen wird, entsteht das Problem, dass die Daten ohne Encoding Informationen angezeigt werden. Das dadurch entstehende Umlaut-Problem ist im deutschsprachigen Raum lästig, lässt sich aber leicht beheben. Einfach dem zu ladenden File in die erste Zeile folgenden Code hinzufügen:

```
<?php header('Content-Type: text/html; charset=utf-8');?>
```

Die größten Nachteile.

Da mit Ajax veränderte bzw. generierte Webseiten nur „temporär“ im Arbeitsspeicher existieren, können diese auch nicht hundertprozentig mit einem Lesezeichen gespeichert werden. Das bedeutet, dass das Lesezeichen lediglich die Seite anspricht, welche zu Beginn, also vor den Ajax Veränderungen, zu sehen war.

Der zweite gravierende Nachteil ist, dass die Zurücktaste des Browsers nicht auf die Ajax Veränderungen reagiert, sondern alle Ajax Änderungen ignoriert und eine ganze Webseite zurück schaltet.

Warum sollte Ajax überhaupt verwendet werden?

Im Frontend:

Die Möglichkeit, Tooltips nachzuladen (Beispiel: www.sixt.de, Eingabefeld für den Ort) oder Eingaben zu validieren und entsprechende Antworten oder Formulare nachzuladen (Beispiel: www.toskana-maremma.com/buchungskalender.php, Datumseingabe).

Im Backend:

Content- und Anzeigenänderungen in programmähnlichen Anwendungen. Beispiele gibt es reichlich, doch sind diese wie zu erwarten passwortgeschützt. Alternativ kann ein Blick in das Verleih- und Reservierungs-Tool der SAE München geworfen werden – spricht einfach mal die Supervisor an.

Gregor Müller-Elmau

Author – Gregor Müller-Elmau



Website: www.cuda.at

Email: gregor@cuda.at oder gregorme@online.de

SAE Webdesign Diploma February 2008

since 2008 freelancer, SAE Webdesign & Development Supervisor and webteam member of eSixt (Sixt AG)