

FACHARTIKEL (FREESTYLE)

Industry Perspectives & Issues

Module Name:	Creative Media Practice
Module Number:	SAE 2000
Block:	BA Essentials
Date Submitted:	23.01.2011
Award Name:	BA / BSc (Hons) Web Development
Year:	2010/2011
Name:	Böck, Max
City:	Vienna
Country:	Austria
Staffing:	Ergert, Klaus
Word Count:	1274

FORMULARE IM WEB DER ZUKUNFT- NEUE WEGE MIT HTML 5

Jan 2011

Autor: Max Böck



Webformulare haben Entwicklern schon immer einiges an Kopfzerbrechen bereitet. Hinter den scheinbar simplen Eingabefeldern steckt mehr Arbeit als vermutet, denn eine ordentliche Überprüfung der eingetragenen Daten ist sowohl server- wie clientseitig unerlässlich. Daran wird sich auch in Zukunft nichts ändern, doch bieten die neuen Helden HTML5 und CSS3 einige interessante Lösungen, um dem User optimal beim Ausfüllen zu assistieren und dem Entwickler Zeit und Nerven zu sparen. Clientseitige Formularvalidierung könnte schon bald einfacher sein als je zuvor.

Ansätze & Neuerungen

Seit der ersten Veröffentlichung der HTML5 Spezifikation im Januar 2008 hat das W3C zahlreiche neue Elemente vorgestellt, darunter bis dato auch 13 neue Typvarianten des `<input>` Tags. Wurden bis jetzt die meisten Formulare hauptsächlich aus `<input type='text'>` Feldern konstruiert, so ermöglicht HTML5 eine sehr viel genauere Definition der erwarteten Daten, was einige Vorteile mit sich bringt.

Ein Beispiel eines neuen Inputs ist etwa das lang erwartete email-Feld, das ohne weiteren Code eine zulässige Emailadresse erkennen kann. Auch Eingabemöglichkeiten, die bereits zum guten Ton gehören, bislang aber nur über 'UI-Plugins' möglich waren, wie zB. Slider und Date- oder Colorpicker, werden vertreten sein. Sechs der neuen Inputtypen behandeln Datum oder Zeit: 'datetime', 'date', 'month', 'week', 'time', und 'datetime-local'. Andere Varianten sind 'number', 'tel' oder 'url' – ihre Funktionsweise ist weitgehend selbsterklärend. All diese Elemente haben gewissermaßen eine

direkt eingebaute Validierung und verhindern bei einem Fehler das Absenden des Formulars.

Während für viele der oft gefragten simplen Datentypen nun eigene Inputfelder existieren, gibt es auch Lösungsansätze für komplexere Strukturen. Mit dem neu eingeführten Attribut 'pattern' lassen sich Eingaben mit jeder beliebigen Form von Zeichenkette über reguläre Ausdrücke abgleichen.

Bei numerischen Daten helfen außerdem die Attribute 'min', 'max' und 'step', welche Unter- und Obergrenze bzw. Interval eines Wertes festlegen. Folgendes Markup:

```
<input type='number' name='example'  
min='1000' max='9990' step='10'  
required />
```

würde also nur numerische Eingaben zwischen 1000 und 9990 erlauben, die in Zehnerschritten erfolgen - Zum Beispiel 1010, 1020 ... 9980 usw. Das Attribut 'required' zeichnet das Feld zusätzlich als verpflichtend aus.

Aufgreifen bestehender Praktiken

Inline Validierung von Formularen hat sich im Bezug auf Usability bereits bewährt – dem User wird gewissermaßen ‘live’ Hilfestellung geleistet. Fehler werden früher erkannt, korrigiert, und die Korrektur sofort bestätigt. Dadurch wird ein Abschicken von fehlerhaften Formulardaten verhindert, und die serverseitige Validierung (obgleich unverzichtbar) dient im Optimalfall nur als zusätzliche Versicherung.

Bislang wurden clientseitige Validierungstechniken immer über Javascript oder entsprechende Frameworks / Plugins realisiert – ein Umstand, der viele nützliche Skripte geschaffen hat, die diese Aufgabe optimal bewältigen. Doch sollte eine so elementare und weitverbreitete Funktion nicht nativ abgedeckt werden? Genau diese Frage greift das W3C in der HTML5 Spec auf. Ein Grundsatz aus ihren *Design Principles* lautet:

“When a practice is already widespread among authors, consider adopting it rather than forbidding it or inventing something new.”

-HTML Design Principles, W3C Working Draft 26

Eine Idee, die sinnvolle Weiterentwicklung von Code vorantreiben soll. Das Prinzip der clientseitigen Validierung wird laut Spezifikation des W3C (aktuell in Working Draft 13) also künftig mittels nativer Inputfelder übernommen, entsprechendes User-Feedback wird in CSS umsetzbar sein.

Um unterschiedliche Zustände der Formularfelder zu erkennen und behandeln, sind im ‘UI Module’ von CSS3 einige neue Pseudo-Klassen vorgesehen. Die wichtigsten sind ‘valid’, ‘invalid’ und ‘required’ – Sie ermöglichen entsprechendes Styling der Felder in Echtzeit und geben dem User somit Feedback über seine Eingaben. So kann etwa mit dem Selektor ‘input:focus:required:invalid’ ein Pflichtfeld angesprochen werden, dass falsch ausgefüllt wurde. Umgekehrt kann natürlich auch eine Meldung über die korrekte Eingabe angezeigt werden, sobald die Bedingungen des Feldes erfüllt wurden. Der Zusatz ‘focus’ stellt sicher, dass das jeweilige Feedback erst beim Ausfüllen sichtbar wird, und nicht schon nach dem erstmaligen Laden der Seite. Diese Technik beinhaltet allerdings ein Problem, falls ein Feld übersprungen oder aus einem anderen Grund ignoriert wird. Wenn das Feld niemals selektiert wird, greift auch das Error-Styling nicht. Ein Fehler würde erst angezeigt, sobald das betreffende Element erneut Fokus erhält.

Weitere Pseudo-Klassen, die den Umgang mit Fehlermeldungen erleichtern sollen, sind ‘in-range’ und ‘out-of-range’, die für den gemeinsamen Einsatz mit den ‘min’ und ‘max’ Attributen eines numerischen Inputs gedacht sind.

Usability & Accessibility

Um dem User den Umgang mit Formularen zusätzlich zu erleichtern, wurden einige Attribute geschaffen, die den Aspekt der Usability verbessern sollen. ‘autofocus’ etwa setzt den Cursor bereits zu Beginn in das erste Eingabefeld, das ‘placeholder’-Attribut ermöglicht den Einsatz von Platzhaltertext, der Auskunft über die erwarteten Daten gibt

und beim Ausfüllen verschwindet. Durch ‘list’ ist es möglich, Feldern Auswahlmöglichkeiten ähnlich einer Select-Box hinzuzufügen, und ‘autocomplete’ füllt oft gefragte Informationen wie Name oder Emailadresse basierend auf früheren Eingaben automatisch aus.

Generell würde durch den einheitlichen Um-

gang mit einigen der komplexeren Formulardaten und den Konventionen, wie diese einzugeben sind, dem User das Neuorientieren und Kennenlernen von Widgets und anderen scriptbasierten Inputformen der Gegenwart abgenommen.

Die genauere Spezifikation der Eingabefelder bringt auch im Bereich der mobilen Endgeräte Vorteile in der Usability mit sich. Im Browser von Apples iPhone zum Beispiel werden bereits viele der neuen HTML5 In-



angepasstes Tastenfeld bei URL-Input am iPhone

putfelder unterstützt. Bringt der User ein solches Eingabefeld in den Fokus, wird ihm automatisch eine angepasste Version des Tastenfeldes angeboten – handelt es sich etwa um ein Emailfeld, sind eigene Tasten für '@' und '.' reserviert, bei URLs gibt es Tasten für Slash und '.com'. Blackberry und einige Android Smartphones stellen bei HTML5 Inputs wie dem Color- oder Datepicker zudem ihr eigenes Userinterface zur Verfügung, und zeigen bei Bedarf etwa Informationen aus dem Kalender.

Weiters geht mit einer verbesserten Semantik auch eine Steigerung der Accessibility einher. Screenreader und andere Werkzeuge profitieren von der detaillierten Auszeichnung der Formulare und können effektivere Hilfestellung leisten. Zusätzlich ist bei einer Implementierung der neuen Eingabetypen seitens der Browserhersteller ein höheres Augenmerk auf Accessibility zu erwarten, als momentan in den diversen Javascript-PlugIn Varianten existiert.

Browsersupport & Workarounds

So wie bei den meisten Neuerungen, die HTML5 mit sich bringt, existiert leider auch für diese noch kein vollständiger Browsersupport. Chrome, Opera und Safari unterstützen bereits die wichtigsten neuen Elemente, auch die kommende Firefox Version 4 wird verbesserten HTML5 und CSS3 Support haben. Inwiefern die neuen Funktionen in Microsofts Internet Explorer 9 Unterstützung finden, ist jedoch noch fraglich.

Javascript-Lösungen für clientseitige Formularvalidierung werden also noch länger nötig sein, doch bedeutet das nicht, dass ein Einsatz der neuen Inputs noch nicht möglich ist. Sie alle sind rückwärtskompatibel und werden in älteren Browsern einfach als 'type=text' gerendert – so wie sie sonst auch verwendet würden. Zusätzlich ist über entsprechendes Javascript auch die Emulation der HTML5 Funktionen in älteren Browsern möglich. Ein

solches Script würde nach dem Laden des Markups überprüfen, ob Unterstützung der Neuerungen besteht, und bei Fehlen dieser entsprechende Funktionen bereitstellen. Das Framework jQuery bietet so in seinen neuen "jQuery Tools: Form" eine nur 5,6 kb große, komplett auf HTML5 basierende Lösung für Formulare an. Datepicker, Range-Slider und andere neue Inputs werden nötigenfalls durch das Framework emuliert und können so cross-browser tauglich angewendet werden.

Einen anderen Ansatz liefert das inzwischen weit verbreitete Script *Modernizr* (www.modernizr.com), in seiner aktuellen Version 1.6 unkomprimiert 33,6 kb groß. Modernizr testet Browser auf ihren Support der neuen Elemente und Attribute, und ermöglicht dann den einfachen, zielgerichteten Einsatz von Fallback Varianten in älteren Browsern.

Conclusio

Viele der Neuerungen im Bereich von Webformularen können bereits heute Anwendung finden. Obwohl die Umstellung auf HTML5 und die Unterstützung des vollen Funktionsumfangs wohl noch einige Zeit auf sich warten lassen werden, entstehen durch den Einsatz der Technologie schon jetzt viele Vorteile.

Besonders im Bereich der mobilen Endgeräte wie Smartphones oder iPad kann eine deutliche Verbesserung erzielt werden. Noch ist für eine wirklich konsistente Oberfläche in allen Browsern zwar der zusätzliche Gebrauch von Scriptlösungen nötig, doch werden in Zukunft zweifelsohne immer mehr Entwickler auf die neuen Möglichkeiten zurückgreifen - und so langsam aber sicher einen neuen Standard etablieren.

Quellen

Kröner, Peter (2010) "HTML5 - Webseiten innovativ und zukunftssicher". S.143ff München: Open Source Press.

Seddon, Ryan (2010) "Forward Thinking Form Validation" In: A List Apart, 314. Zugriff am 18. Jänner 2011 unter:
<http://www.alistapart.com/articles/forward-thinking-form-validation>

Pilgrim, Mark (2010) "Dive Into HTML5". Zugriff am 17. Jänner 2011 unter:
<http://diveintohtml5.org/forms.html>

W3C HTML5 Spec (Working Draft 13)
<http://www.w3.org/TR/html5/>
Zugriff am 21. Jänner 2011

W3C CSS3 Basic User Interface Module
<http://www.w3.org/TR/css3-ui/>
Zugriff am 21. Jänner 2011

Mozilla Firefox 4 (Beta)
<http://www.mozilla.com/en-US/firefox/beta/technology/> Zugriff am 22. Jänner 2011

Blackberry Developer Blog
<http://devblog.blackberry.com/2010/08/blackberry-html5/> Zugriff am 23. Jänner 2011

