Local Storage und andere HTML5-Highlights in APEX

Andreas Wismann
WHEN OTHERS
D-41564 Kaarst

Schlüsselworte
HTML5, APEX 4.2, Local Storage

Einleitung

Nicht nur in Web-Entwicklerkreisen ist HTML5 längst der arrivierte Standard. Einige HTML5-Templates sind bereits in Oracle Application Express integriert. Features wie die neuen Attribute von Formularelementen, Local Storage und Canvas lassen sich bereits heute in der Mehrzahl der aktuellen Browser problemlos einsetzen. Doch welche praktischen Einsatzszenarien ergeben sich mit "HTML5" für Architekten und Entwickler von APEX-Anwendungen? Der Vortrag zeigt Beispiele dafür, wie man mit den neuen Elementen von HTML5 umgeht, was sie in APEX-Anwendungen (Desktop und mobil) bewirken können und inwieweit man Application Express dabei noch etwas auf die Sprünge helfen kann.

Wann spricht man von HTML5?

HTML5 ist abwärtskompatibel zu früheren HTML-Dokumenttypen. Zudem können Sie HTML5-Dokumente schreiben, ohne zwingend Gebrauch von irgendwelchen neuen Features zu machen. Um ein solches Dokument zu erstellen, genügt es, seinen Dokumenttyp entsprechend auszuzeichnen:

```
<!DOCTYPE html>
```

Nun bewirkt die Änderung des Dokumenttyps allein nicht viel. In einem typischen HTML5-Dokument sollten natürlich auch HTML5-Elemente zum Einsatz gelangen. Hier eine beispielhafte Liste typischer HTML5-Tags, die bei weitem nicht vollständig ist:

Tag	Erläuterung
`<nav></nav>`	Navigationsbereich (keine unmittelbare Auswirkung)
`<section>…</section>`	Inhaltlicher Abschnitt (keine unmittelbare Auswirkung)
`<header>…</header>`	Kopfbereich (keine unmittelbare Auswirkung)
`<footer>…</footer>`	Fußbereich (keine unmittelbare Auswirkung)
`<input type="search">`	Suchfeld (manche Browser heben dieses Feld optisch hervor)
`<input type="date">`	Datumseingabe (manche Browser blenden Datepicker ein)

`<input type="phone">`	Eingabe einer Telefonnummer (manche Browser prüfen die Eingabe auf ein bestimmtes Format)
`<input type="email">`	Eingabe einer E-Mail-Adresse (manche Browser ändern die Tastaturdarstellung)
`<input type="number" step="10">`	Zahleneingabe (hier: in Zehnerschritten. Manche Browser blenden Zahlenauswahl-Bedienelemente ein)
`<input type="range" min="0" max="100">`	Zahleneingabe innerhalb eines Wertebereichs (manche Browser stellen einen Slider dar)

Was liefert APEX bereits mit?
In einigen Seiten-Templates, die mit APEX 4.2 ausgeliefert werden, befinden sich bereits Elemente aus dem Sprachschatz von HTML5. Insbesondere bei den Mobile Themes werden die neuen Input-Elemente verwendet. Man findet beispielsweise auch header- und footer-Elemente, wobei entsprechende CSS-Regeln eine bestimmte Platzierung bzw. Aussehen verleihen.

Welche HTML5-Elemente kann der Anwendungsentwickler selbst hinzufügen und wo?
HTML-Inputfelder können nicht als eigene Templates definiert werden, weil es in APEX – außer für Kalender-Items – leider keine Template-Klasse für Eingabefelder gibt. Möchte man beispielsweise ein Feld für die Eingabe einer E-Mail-Adresse verwenden (input type="email"), so führt der Weg entweder über die Erstellung eines APEX Plugins oder über die Umwandlung der Type-Attribute mittels jQuery unmittelbar nach dem Laden der Seite.

Wesentlich einfacher ist die Verwendung semantischer Abschnitts-Tags in Seiten-Templates, so wie es bereits von der APEX-Entwicklern vorgemacht wurde. Es spricht nichts dagegen, anstelle der mitgelieferten Templates ein vollständig selbst definiertes Template zu verwenden, das mit Hilfe von HTML5 und CSS3 aufgebaut ist. Beispiele für moderne Webdesign-Vorlagen finden sich im Web zahlreich. Ein recht bekanntes und ausgereiftes Projekt ist

http://yaml.de

Ist der Browser überhaupt online?
Regelmäßig passiert es, dass der Benutzer kurzzeitig nicht online ist (Stichwort Eisenbahntunnel). Mit HTML5 und JavaScript (hier: jQuery) kann – in gewissen Grenzen – festgestellt werden, ob das der Fall ist.

```
var onlineStatus;

/** Nach dem Laden der Seite regelmäßig den Onlinestatus abfragen: */
$(document).ready(function() {
  window.setInterval(zeigeOnlineStatus, 1000);
})
```

```
/**
 * Prüft, ob der Browser online ist, stellt das Anzeigeelement
 * mit der id="#onlineIndikator" entsprechend visuell um
 * und setzt die globale Variable onlineStatus auf true|false.
 */
function zeigeOnlineStatus () {
  // eine neue Eigenschaft HTML5-fähiger Browser:
  if(navigator.onLine) {
    $('#onlineIndikator').show(); // dieses Element müssen Sie erstellen
    onlineStatus = true;
  else
    $('#onlineIndikator').hide('slow');
    onlineStatus = false;
}
```

Daraufhin kann in einem HTML5-Formular beim Betätigen eines SUBMIT-Buttons geprüft werden, ob der Server prinzipiell erreichbar ist. Ist der Benutzer offline, so wird gar nicht erst versucht, das Formular abzusenden (was unnötige Wartezeiten ersparen kann) und der User erhält einen entsprechenden Hinweis:

```
$('form').submit(function(){
  if(!onlineStatus) {
    alert('Es besteht momentan keine Verbindung zum Server');
    return false;
  }
  else {
    return true;
  }
});
```

Zwar gibt es für diese Anwendung noch einige Besonderheiten und Fallstricke zu beachten (diese werden im Vortrag demonstriert), doch im Wesentlichen funktioniert es bereits so.

Local Storage: eine mächtige Alternative zu Cookies

In Cookies werden bekanntermaßen Key/Value-Paare gespeichert. Das Datenvolumen von Cookies pro Webdomain ist in vielen Browsern gemäß W3C-Standard begrenzt (maximal 20 Cookies mit insgesamt 4 Kilobyte), was für Schlüsselinformationen meist ausreicht. Auch APEX verwendet Cookies, beispielsweise um den Benutzer zu identifizieren und den Anwendungsverlauf zu steuern.

HTML5-fähige Browser beherrschen darüber hinaus „Local Storage“. Sie können damit wesentlich mehr Informationen speichern als in den Cookies. So lässt sich mit wenig Aufwand eine JavaScript-Routine schreiben, die in der Lage ist, alle in einem Webformular getätigten Eingaben lokal (d.h. auf dem Computer des Anwenders) zu speichern, um sie zu einem späteren Zeitpunkt wieder auszulesen.

Anwendungsfälle hierfür sind beispielsweise:

- Der Benutzer ist vorübergehend nicht online (siehe vorheriger Abschnitt), die eingegebenen Daten werden deshalb zwischengespeichert; sie können später wieder lokal ausgelesen und erneut an den Server gesendet werden
- Der Benutzer kann eine oder mehrere Vorlagen zum Ausfüllen von Formularen erstellen, wobei die Daten auf Knopfdruck in das Formular übertragen werden – unabhängig von einer Implementierung in der Datenbank

Der prinzipielle JavaScript-Code zum Abspeichern eines Schlüssel-/Werte-Paares im Local Storage lautet:

```
localStorage.setItem("schluessel", "wert");
```

Ausgelesen wird mittels

```
localStorage.getItem("schluessel");
```

Um die Daten eines APEX-Formulars vorübergehend lokal abzuspeichern, kann man den folgenden Ansatz wählen:

- Ermitteln aller Items, die abgespeichert werden müssen, beispielsweise als Liste von IDs: „#P50_CUSTOMER_ID, #P50_CUSTOMER_FIRST_NAME, #P50_CUSTOMER_LAST_NAME, …“
- Eine Schaltfläche, mit der ein Benutzer das Abspeichern der Formularinhalte auslösen kann
- Eine jQuery-Routine, welche über diese Items iteriert und alle Werte im Local Storage ablegt

Genau diese Implementierung wird in meinem DOAG-Vortrag gezeigt. Dabei gehe ich auch auf ein Projekt namens jStorage ein; dabei handelt es sich um ein Framework, welches sich in jQuery integriert und den Umgang mit Local Storage, auch im Hinblick auf ältere Browser, erheblich vereinfacht. Das Ergebnis kommt mit bemerkenswert wenig Code aus und ist auch für JavaScript-Einsteiger gut nachvollziehbar.

Weitere HTML5-Highlights, die ich in meinem Vortrag am Mittwoch, 20.11.2013, 15:00 Uhr, Raum 13 im Zusammenspiel mit einer APEX-Anwendung vorstellen möchte:

- Canvas (parametergesteuertes Zeichnen)
- Web Worker (clientseitige Aufgaben im Hintergrund ausführen)
- Geolocation („wo bin ich?“)

Das aktualisierte Manuskript, meine Präsentationsfolien sowie alle Beispiel-Quellcodes können Sie im Anschluss an die DOAG 2013 auf meiner Webseite herunterladen:

http://when-others.com/download/doag/2013

Kontaktadresse:

Andreas Wismann
WHEN OTHERS
Hirschstraße 10
D-41564 Kaarst

Telefon: +49 (0) 2131-314 9966
E-Mail: wismann@when-others.com
Internet: www.when-others.com