

JavaScript-Erweiterungen für UI und UX

Kai Donato
MT AG
Ratingen

Schlüsselworte

JavaScript, Pace, UX, UI, CSS,

Einleitung

Wir als Entwickler stehen regelmäßig vor der Aufgabe, Geschäftsprozesse in APEX abzubilden. Es ist dabei nicht nur wichtig, einen korrekten Ablauf zu gewährleisten, sondern auch für die Bedienbarkeit durch den Endanwender das Beste aus APEX herauszuholen. Ich möchte Ihnen anhand von JavaScript-Erweiterungen zeigen, wie einfach man mit wenig Aufwand für eine bessere User Experience (UX) und ein ansprechendes User Interface (UI) sorgen kann.

Ein Blick über den Tellerrand

Es ist häufig äußerst hilfreich einen Blick über den Tellerrand und die Standardfunktionen der eingesetzten Technologie zu werfen. Einen näheren Blick auf bewährte User-Experience (UX) Techniken kann die Erfahrung, die der Anwender mit Ihrer Applikation macht, positiv beeinflussen. Ein durchdachtes Konzept und ein gutes Zusammenspiel zwischen UI und UX kann der Erfolg ihrer Applikation bedeuten.

Loading... (Visuelle Einflüsse mit JavaScript in APEX)

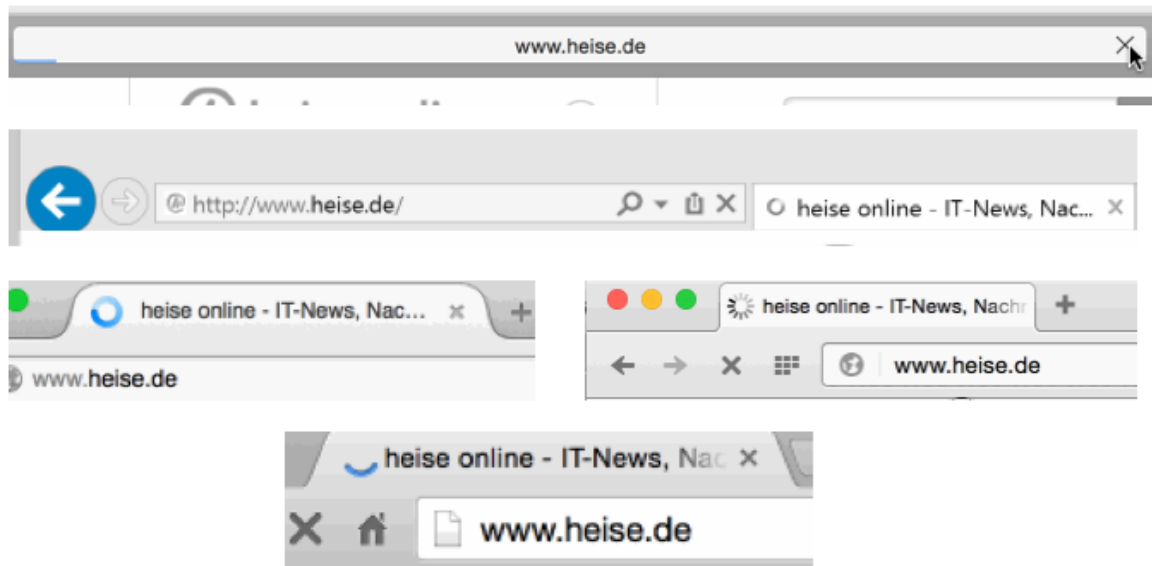


Abbildung 1: Ladeindikatoren

Jeder Browser besitzt eine Art Indikator, der dem Anwender suggeriert, dass gerade ein Prozess durchgeführt wird. Leider wird dieser Indikator nicht bei jedem asynchronen Prozess innerhalb der

Anwendung ausgelöst und lässt den Endbenutzer des Öfteren warten, ohne ihn visuell auf dem Laufenden zu halten.

Durch den gezielten Einsatz eines JavaScript-Frameworks (PACE) lässt sich mit geringem Aufwand ein Lade-Indikator innerhalb der Applikation einblenden. Belässt man die Konfiguration beim Standard, so wird bei jedem JavaScript-Request, der in der Anwendung gestartet wird, ein Indikator angezeigt. Folgende Indikator können einfach per CSS gewählt werden:

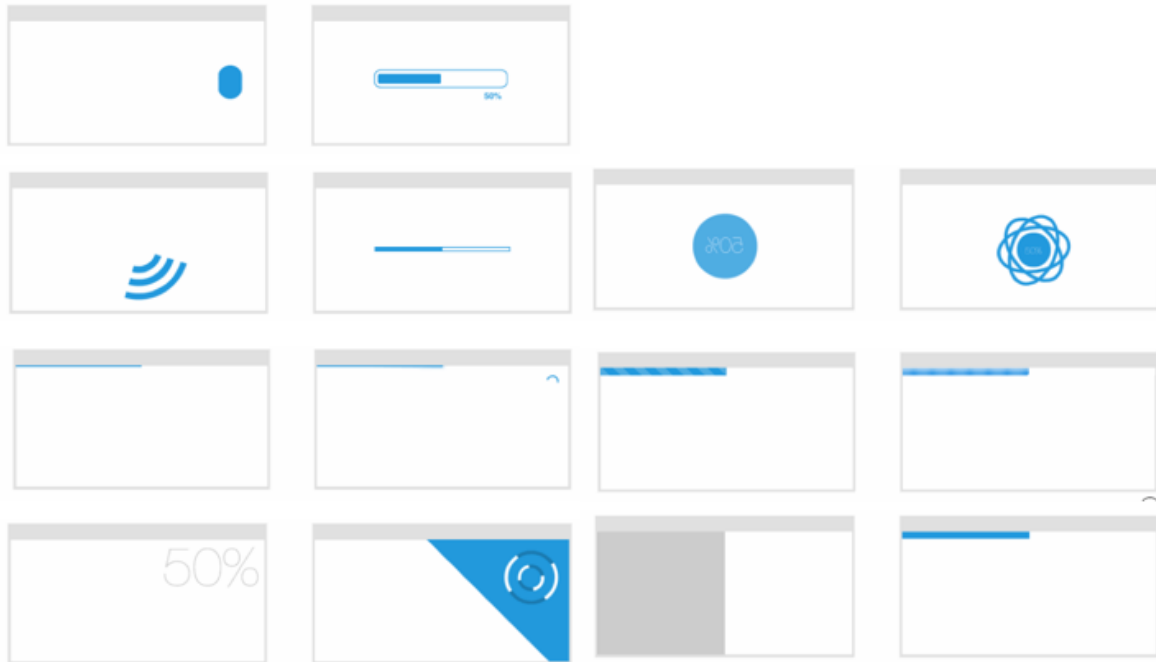


Abbildung 2: PACE-Presets

UX-Einflüsse mit JavaScript in APEX

Neben den visuellen Einflüssen, die den Anwender bei der Verwendung der Applikation unterstützen, ist auch der Einsatz von funktionalen Erweiterungen sinnvoll. Durch die Verwendung von funktionserweiternden JavaScript-Frameworks kann die Benutzung der Anwendung erleichtert werden und möglicherweise, wie im folgenden Beispiel sogar für die barrierefreie Verwendung gesorgt werden.

Mousetrap

Mousetrap kann mit seinen schlanken 2 KB (minimized) in die Applikation implementiert werden und im Anschluss mit Hilfe eines Definitionsblocks im Kopf der APEX-Seite um Tastenkombinationen erweitert werden. Im Definitionsblock kann der Entwickler von Tastenanschlägen bis hin zu Tastenkombinationen oder nacheinander betätigte Tasten erkennen und verarbeiten. Neben der Cross-Browser-Kompatibilität ist die Unterstützung von internationalen Tastatur-Layouts ein klarer Vorteil gegenüber der herkömmlichen Entwicklung.

```

// single keys
 Mousetrap.bind('4', function() { highlight(2); });
 Mousetrap.bind('x', function() { highlight(3); }, 'keyup');

// combinations
 Mousetrap.bind('command+shift+k', function(e) {
   highlight([6, 7, 8, 9]);
   return false;
 });

```

Abbildung 3: Definition von Mousetrap-Events

VanillaMasker

Beim VanillaMasker-Framework handelt es sich um eine JavaScript-Erweiterung, die den Endanwender bei der Eingabe von Daten in Formulare unterstützen soll. Hierbei kann beispielsweise ein Format der einzugebenden Daten bestimmt werden und dieses dann in Echtzeit bei der Eingabe durch den Endanwender überprüft werden. So kann unter anderem dafür gesorgt werden, dass ein Datum oder eine Artikel- bzw. Seriennummer stets korrekt eingegeben wird.

```

// Listen the input element masking it to format with pattern.
VMasker(document.querySelector("input")).maskPattern("(99) 9999-9999");

// Converts value to masked phone
VMasker.toPattern(10999911111, "(99) 9999-9999"); // -> (10) 9991-1111
// Converts value to masked date
VMasker.toPattern(12122000, "99/99/9999"); // -> 12/12/2000
// Converts value to masked document
VMasker.toPattern(99911111101, "999.999.999-99"); // -> 999.111.111-01
// Converts value to masked car plate
VMasker.toPattern('ABC1234', "AAA-9999"); // -> ABC-1234
// Converts value to masked vehicle identification
VMasker.toPattern('9BGRD08X04G117974', "SS.SS.SSSSS.S.S.SSSSS"); // -> 9B.GR.D08X0.4.G.11797

// Pass in an optional placeholder option to represent remaining characters to be entered
VMasker.toPattern('4', {pattern: "(999) 999-9999", placeholder: "x"}); // -> (4xx) xxx-xxxx

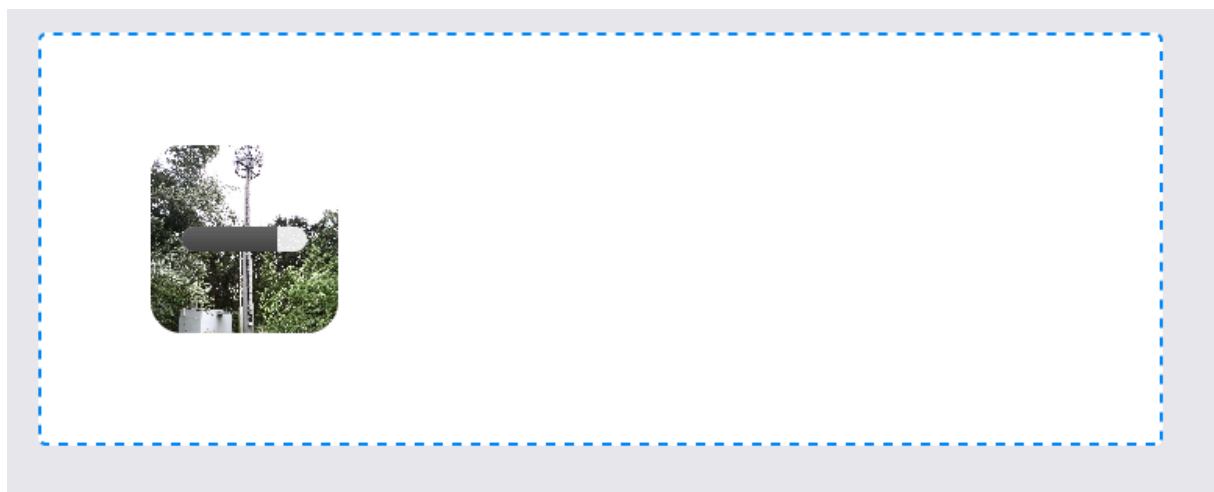
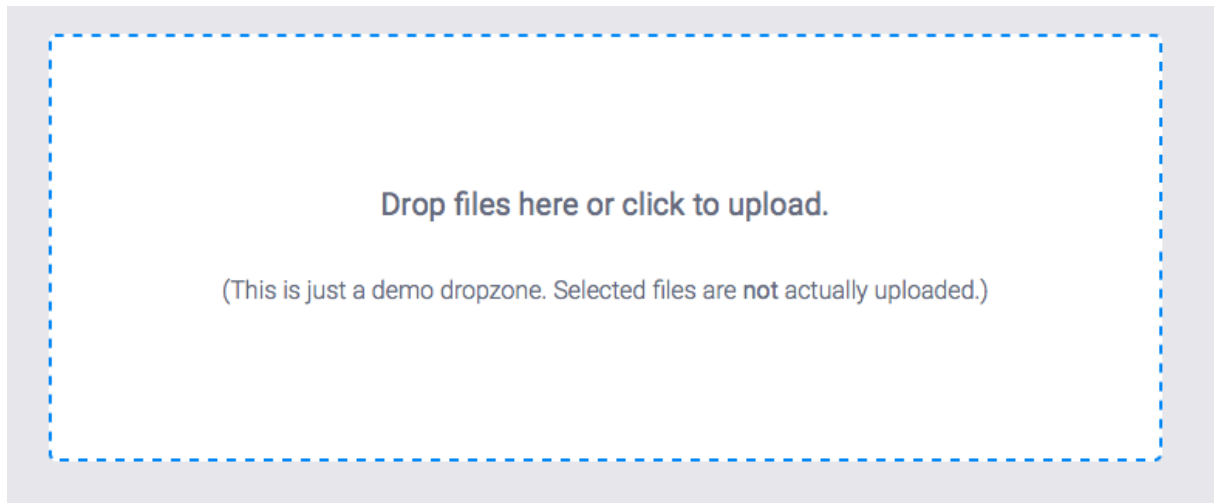
```

Abbildung 4: Festlegen von Formatvorgaben für Eingabefelder

DropzoneJS

Um Webanwendungen so intuitiv wie möglich bedienbar zu machen, kann man sich an bereits bekannte Strukturen halten und beispielsweise funktionsweisen von Desktop-Applikationen übernehmen. So ist es bei DropzoneJS der Fall. DropzoneJS ermöglicht es, Dateien per Drag&Drop in den Browser zu ziehen und nach dem Loslassen der Maustaste mit dem abgelegten Inhalt zu interagieren. So kann der Anwender eine Datei aus einem Ordner auf der Festplatte in einen gekennzeichneten Bereich in der Webapplikation ziehen und sie dort ablegen. Direkt nach dem

Ablegen der Datei beginnt der Browser mit dem Upload und zeigt den Fortschritt auf der abgelegten Datei an.



Mobile Anwendungen

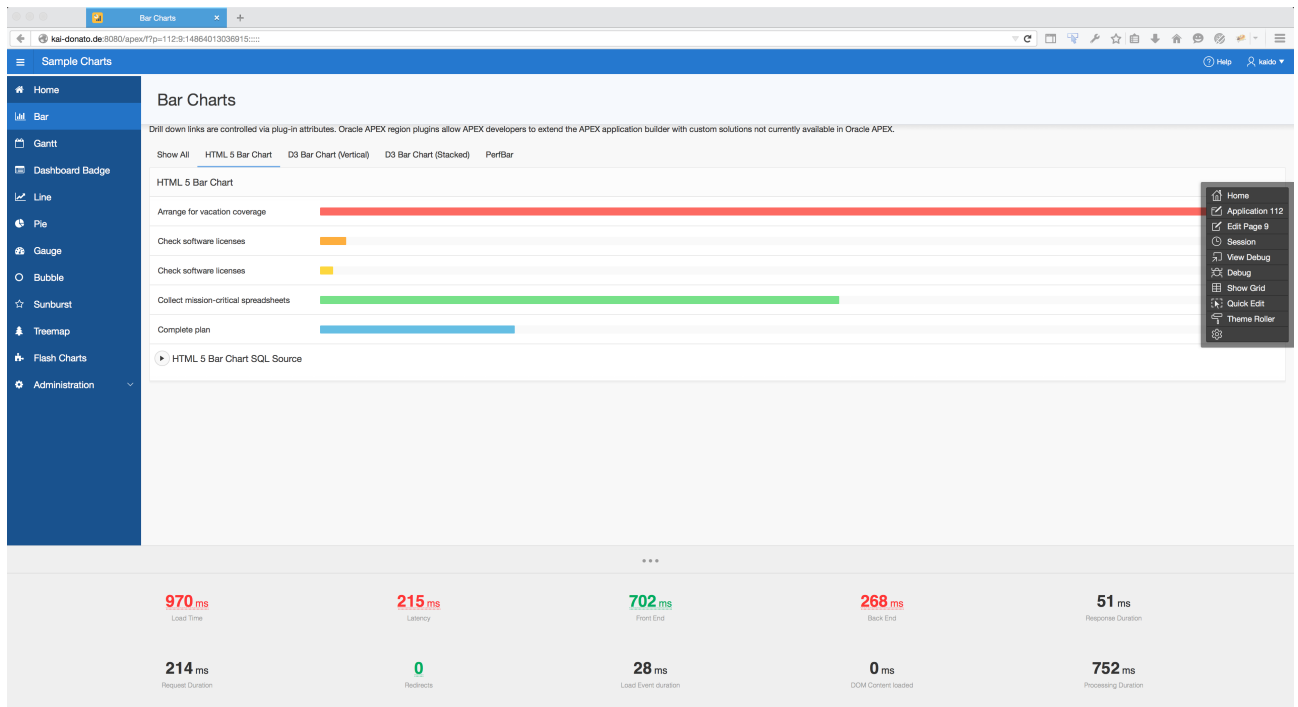
Eine Variante mobile Applikationen zu entwickeln ist, ein fertiges Grundgerüst zu nehmen und dieses dann mit Skriptsprachen wie JavaScript zu erweitern. Das bekannteste Framework hierfür ist „Apache Cordova“, das früher unter dem Namen „PhoneGap“ bekannt geworden ist. Der Vorteil an dieser Technologie ist, dass mit Hilfe von JavaScript auf das mitgelieferte Grundgerüst zugegriffen werden kann und mit Hilfe der mitgelieferten Schnittstellen auf die nativen APIs des mobilen Betriebssystems zugegriffen werden kann. Hierzu gehören beispielsweise Kamerafunktionen, Kalenderzugriff, Bluetooth oder auch das versenden von Emails. Die Unterstützung verschiedener Schnittstellen lässt sich modular aufbauen und via PlugIns hinzufügen und entfernen. Der offizielle Plugin-Katalog des Frameworks umfasst aktuell 1113 Plugins (Stand September 2015).

Ein großer Vorteil ist definitiv der Dienst, den Adobe mit dem Projekt „PhoneGap Build“ ins Leben gerufen hat. Der Dienst ermöglicht es, eine in Cordova bzw. PhoneGap entwickelte Applikation hochzuladen und durch den Dienst für alle Plattformen zu kompilieren. Somit kann ein Entwickler auch ohne Apple-Hardware eine Applikation kompilieren und sogar signieren. Dies ermöglicht im Anschluss die Veröffentlichung im App-Store.

	amazon- fireos	android	blackberry10	Firefox OS	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.0, 8.1, Phone 8.1)	tizen
cordova CLI	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓	✗
Embedded WebView	✓ (see details)	✓ (see details)	✗	✗	✓ (see details)	✓	✗	✗	✗
Plug-in Interface	✓ (see details)	✓ (see details)	✓ (see details)	✗	✓ (see details)	✓	✓ (see details)	✓	✗
Platform APIs									
Accelerometer*	✓	✓	✓	✓	✓	✓	✓	✓	✓
BatteryStatus*	✓	✓	✓	✓	✓	✗	✓	✓ Windows Phone 8.1 only	✓
Camera*	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capture*	✓	✓	✓	✗	✓	✓	✓	✓	✗
Compass*	✓	✓	✓	✗	✓ (3GS+)	✓	✓	✓	✓
Connection*	✓	✓	✓	✗	✓	✓	✓	✓	✓
Contacts*	✓	✓	✓	✓	✓	✓	✓	partially	✗
Device*	✓	✓	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✗	✓	✓	✓	✓	✓
File*	✓	✓	✓	✗	✓	✓	✓	✓	✗
File Transfer*	✓	✓	✓ Do not support onprogress nor abort	✗	✓	✗	✓ Do not support onprogress nor abort	✓ Do not support onprogress nor abort	✗
Geolocation*	✓	✓	✓	✓	✓	✓	✓	✓	✓
Globalization*	✓	✓	✓	✗	✓	✓	✓	✓	✗
InAppBrowser*	✓	✓	✓	✗	✓	✓	✓	uses iframe	✗
Media*	✓	✓	✓	✗	✓	✓	✓	✓	✓
Notification*	✓	✓	✓	✗	✓	✓	✓	✓	✓
Splashscreen*	✓	✓	✓	✗	✓	✓	✓	✓	✗
Status Bar*	✗	✓	✗	✗	✓	✗	✓	✓ Windows Phone 8.1 only	✗
Storage	✓	✓	✓	✗	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓
Vibration*	✓	✓	✓	✓	✓	✗	✓	✓ * Windows Phone 8.1 only	✗

Entwicklungsressourcen

Ein weiterer möglicher Einsatz von JavaScript-Erweiterungen, die dem Endanwender der Applikation zu Gute kommen können, gibt es auch nützliche Erweiterungen, die dem Entwickler nützliche Informationen liefern. Beispielsweise das Framework „PerfBar“. Mit minimalem Aufwand lässt sich das Framework implementieren und mit wenigen Codezeilen vollständig zu konfigurieren. Die einzelnen so genannten Metriken lassen sich nach Bedarf zusammenstellen und liefern unter anderem Informationen zu der Ladezeit der Applikation, Anzahl der geladenen Grafiken und der Latenz.



Kontaktadresse:

Kai Donato
MT AG
Balcke-Dürr-Allee 9
D-40882 Ratingen

Telefon: +49 (0) 173-8937790

E-Mail: kai.donato@mt-ag.com

Internet: www.mt-ag.com