

## **Web 3.0? Trends & Techniken in modernen Web-Applikationen**

**Kai Donato  
MT AG  
Ratingen**

### **Schlüsselworte**

HTML5, CSS3, SASS, LESS, Skeumorph, Flat Design, NodeJS, NPM, BOWER, SSL, HTTP/2

### **Einleitung**

Die Entwicklung des WWWs und den damit verbundenen Techniken ist kaum überschaubar und so schnelllebig, dass man schnell auch mal eine Neuigkeit oder einen Trend verpasst. Ich möchte Ihnen einen Überblick über einen kleinen Teil der aktuell angesagten Technologien verschaffen und somit vielleicht sogar einen Anstoß für interessante Experimente innerhalb Ihrer Entwicklungsumgebungen schaffen. Moderne Technologien können zum einen für einen Produktivitätsschub und zum anderen für ungeahnte Möglichkeiten innerhalb bestehender Projekte sorgen.

### **HTML5 und CSS3**

In den letzten Jahren haben sich zwei Technologien besonderer Aufmerksamkeit erfreut. HTML5 und CSS3 sind fast zeitgleich ins Rampenlicht getreten und werden auch schon in sehr vielen Projekten verwendet. Die wohl bekannteste Plattform, die schon „früh“ im großen Stil mit HTML5 gearbeitet hat, ist YouTube. Der Anbieter, der es Benutzern ermöglicht Videos aus dem Internet zu streamen, hat das hochgeladene Videomaterial zunächst ausschließlich mit Flash-Playern abgespielt. Mittlerweile werden unterstützen die bekannten Browser von Google, Microsoft, Firefox, Opera und Apple die HTML5-Player.

CSS3 ist bei der Entwicklung von Online-Applikationen einer der bekanntesten und nützlichsten Versionssprünge gelungen. Mit der Einführung von Animationen und vielen weiteren nützlichen Funktionen hat CSS3 auch in kürzester Zeit Einzug in alle bekannten Browser gefunden.

### **Sass/Less**

Die zunehmende Komplexität von Internetseiten hat das Bedürfnis nach umfangreicheren Gestaltungsmöglichkeiten geschaffen. Mit der Einführung von „Sass“ (2007) und „Less“ (2009) wurde das Entwickeln von Cascading Style Sheets (CSS) um eine Stufe angehoben. Es war nun möglich auf eine große Palette an Funktionen zuzugreifen, die CSS standardmäßig nicht unterstützte. So wurde beispielsweise möglich, verschachtelte Regeln sowie Variablen zu verwenden. Der wesentliche Unterschied zu einfachem CSS-Code ist die unterschiedliche Schreibweise. Sass und Less werden jeweils mit Ihrer eigenen Schreibweise verfasst und

anschließend in valides CSS kompiliert. Das Entwickeln in der jeweiligen Sprache ermöglicht unter Anderem das Verschachteln von CSS-Eigenschaften:

```
#menu {  
  h1 {  
    font-size: 16px;  
    font-weight: bold;  
  }  
  p { font-size: 10px;  
    a { text-decoration: underlined;  
      &:hover { border-width: 2px }  
    }  
  }  
}
```

wird zu folgendem validen CSS-Code kompiliert:

```
#menu h1 {  
  font-size: 16px;  
  font-weight: bold;  
}  
#menu p {  
  font-size: 10px;  
}  
#header p a {  
  text-decoration: underlined;  
}  
#header p a:hover {  
  border-width: 2px;  
}
```

Wie in diesem Beispiel (Less) zu erkennen ist, wird durch das Verschachteln von Selektoren dafür gesorgt, dass kein redundanter Code geschrieben werden muss. Dies hat unter Anderem auch zur Folge, dass die Gefahr von Typos reduziert wird.

## **Vom Skeumorphismus zum Flat Design**

Wo früher viel Wert auf realistische Interfaces gelegt wurde, hat seit der Einführung von Windows Phone 7 (2010) und Apples iOS7 (2013) das Flat Design Einzug in die weitverbreiteten Betriebssysteme für Computer und Smartphones gehalten. Von realistischen Schaltflächen, die beispielsweise Holzoberflächen oder Glas widerspiegeln sollten, befinden sich jetzt fast ausschließlich einfarbige und simple Schaltflächen bzw. UI-Elemente im Fokus.

## **NODEJS**

Bisher galt JavaScript als Script-Sprache, die weitläufig für die client-seitige Verarbeitung von Daten oder später auch für dynamische Webseiten genutzt wurde. Mit der Weiterentwicklung von AJAX können Daten dann auch asynchron, ohne dass die Seite vollständig neu geladen wird, nachgeladen werden und somit schneller bzw. komfortabler mit der Webanwendung gearbeitet werden.

Seit der Veröffentlichung von NodeJS (2008 Open-Source) wird JavaScript-Code nun auch zur serverseitigen Entwicklung genutzt. Besonders für die Entwicklung von Webservern wird häufig auf NodeJS gesetzt. Durch die große Anzahl von PlugIns (ca. 188600 - Stand September 2015) ist NodeJS aus der Webentwicklung nicht mehr wegzudenken. Die Tatsache, dass NodeJS eine „Non-Blocking“ Architektur besitzt und asynchrone Verarbeitung von Daten vornimmt, ermöglicht es dem Entwickler langwierige Prozesse wie z.B. das Auslesen einer größeren Datei oder längerfristige Netzwerktransfers sowie aufwändige Datenbankprozesse im Hintergrund zu bearbeiten. Währenddessen wird der Programmcode ohne unerwünschte Unterbrechung fortgesetzt und durch einen Callback-Block zu einem späteren Zeitpunkt wieder entgegengenommen. Eine Abspaltung des NodeJS-Projekts resultierte in das io.js Projekt, welches nach kurzer Zeit wieder mit dem Hauptprojekt verschmolz.

## **NPM und BOWER**

Durch die zunehmende Menge an Erweiterungen, die sich modular zu jedem Projekt hinzufügen lassen, haben sich die Entwickler von NodeJS dazu entschlossen einen Paketmanager zu entwickeln. Dieser sollte sich so ähnlich nutzen lassen wie die bekannten Paketmanager in diversen Unix-Distributionen. Dies war die Geburtsstunde von NPM (2009). Mit simplen Befehlen wie z.B.

```
npm install express
```

```
npm install cordova
```

```
npm install less
```

lassen sich Pakete automatisch installieren und in einen „node\_modules“ Ordner ablegen. Diese Installationsroutine übernimmt das Auflösen von Abhängigkeiten sowie die Prüfung auf notwendige Systemvoraussetzungen.

Auch das Team von Twitter hat sich zusammen mit der GitHub-Community an die Entwicklung eines Tools gemacht, das ähnlich wie NPM ein Paketmanager für Front-End Frameworks und Librarys ist. Es soll den Programmierer bei der Front-End-Entwicklung unterstützen und Frameworks wie beispielsweise jQuery oder Angular.js mit einem Befehl in die Entwicklungsumgebung laden. Der Unterschied zwischen diesen beiden sehr ähnlichen Technologien ist die bei „Bower“ verwendete flache Abhängigkeits-Hierarchie. Wo bei NPM weniger Wert auf Ressourcen und Latenz gelegt wird, wird bei Bower strenger darauf geachtet, dass es nicht zu Duplikaten in der Auflösung von Abhängigkeiten kommen kann.

NPM würde beispielsweise für zwei verschiedene Pakete auch die beiden referenzierten Abhängigkeiten beziehen. Referenziert das erste Paket z.B. jQuery in der Version 2.0.1 und das zweite, wesentlich neuere Paket, die Version 2.1.4 so würde NPM beide jQuery Versionen herunterladen und verwenden. Dies sorgt zum einen für ein größeres Projektverzeichnis auf Dateisystem-Ebene und zum anderen für eine größere Datenmenge, die an den Browser übertragen wird.

Für die Entscheidung, welches Tool sich ein Entwickler für sein Projekt entscheidet ist die folgende Frage hilfreich:

*Benötige ich ein Paketmanager lediglich zum Hinzufügen von reinen Front-End Frameworks/Librarys oder benötige ich auch erweiternde Module für meine NodeJS-Umgebung?*

Für ersteres ist Bower mit seiner flachen Abhängigkeits-Hierarchie und dem Fokus auf Front-End-Entwicklung die geeignete Wahl. Benötigt man Module für seine NodeJS Instanz, wie z.B. die Datenbankbindung für Oracle oder einen performanten Webserver, so sollte man NPM nutzen. Trotz gewisser Überschneidungen, was die Pakete angeht, sollte diese Frage jedem Entwickler bei der Wahl des richtigen Paketmanagers helfen.

## **SSL und HTTP2**

(Mindestens) Zwei Ereignisse in der Geschichte des Internets haben die Aufmerksamkeit auf eine Technologie geworfen, die eigentlich schon weit verbreitet, aber noch nicht allgegenwärtig war. Die Rede ist von SSL-Verschlüsselung. Meisten Betreiber von Online-Diensten setzten bereits seit vielen Jahren auf SSL-verschlüsselte HTTP Verbindungen. Der Skandal um Edward Snowden hat die meisten Menschen darauf hingewiesen, dass unverschlüsselte Verbindungen einfacher abgehört werden können. Deswegen legen immer mehr Menschen wert darauf, dass bei der Übertragung Ihrer sensiblen Daten immer ein Hinweis auf eine verschlüsselte Verbindung zu sehen ist. Dies ist bei den meisten Browsern mit einem grünen Schloss gekennzeichnet. Im Gegenzug weisen die Browser den Benutzer auch darauf hin, wenn eine Seite unverschlüsselt Daten (bspw. über ein Formular) gesendet

werden. Darüber hinaus warnen Browser den Benutzer ebenfalls, wenn ein nicht verifiziertes SSL-Zertifikat verwendet wird. Mit der Veröffentlichung der Heartbleed-Sicherheitslücke hat eine Linux-Software, die für die Verschlüsselung verantwortlich ist eine ganze Menge negativ PR geerntet und für Aufruhr unter den Serverbetreibern gesorgt.

Google und Microsoft haben maßgeblich bei der Entwicklung eines neuen HTTP-Standards beigetragen. Zu den Vorteilen der neuen Version 2 (HTTP/2) gehören unter Anderem das Bündeln von mehreren Anfragen, weitere Datenkompressionsmöglichkeiten und das Push-Verfahren, dass eine Datenübertragung, die vom Server initiiert wird, ermöglicht.

**Kontaktadresse:**

Kai Donato

MT AG

Balcke-Dürr-Allee 9

D-40882 Ratingen

Telefon: +49 (0) 173-8937790

E-Mail: [kai.donato@mt-ag.com](mailto:kai.donato@mt-ag.com)

Internet: [www.mt-ag.com](http://www.mt-ag.com)