

Von der Webseite zur Webapplikation

Neue Möglichkeiten in der Webentwicklung
am Beispiel von JavaScript Frameworks

Webanwendungen als Nachfolger von Desktop-Applikationen

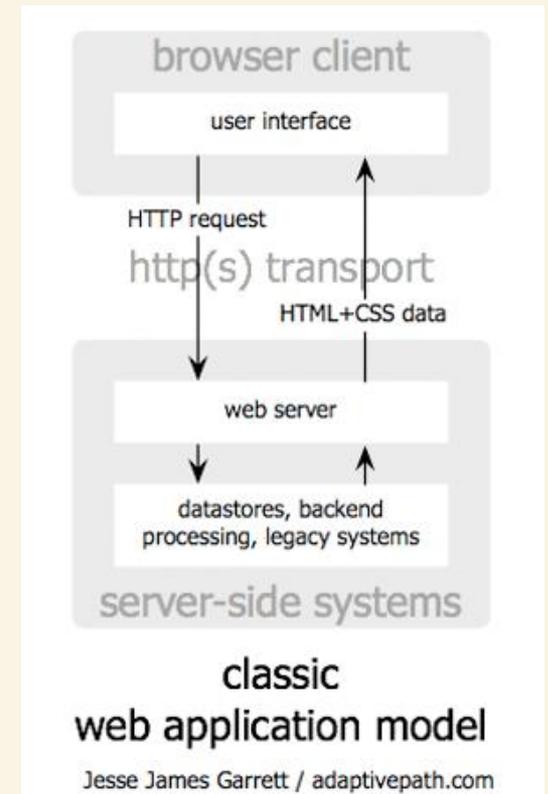
Webseite wird Webapplikation

Herkömmliches Modell

Interaktion bedeutet Request & Reload der Seite

Server stellt Seite mit Daten bereit
Browser dient der Darstellung
Server verarbeitet Events

- Langsam
- Viel Rechenzeit auf dem Server
- JS wird im Browser kaum genutzt
- Enge Bindung zw. Server & Client



Ajax Webapplikationen

Interaktion wird im Browser ausgewertet, schlanke Requests zum Server

Server stellt Seite bereit

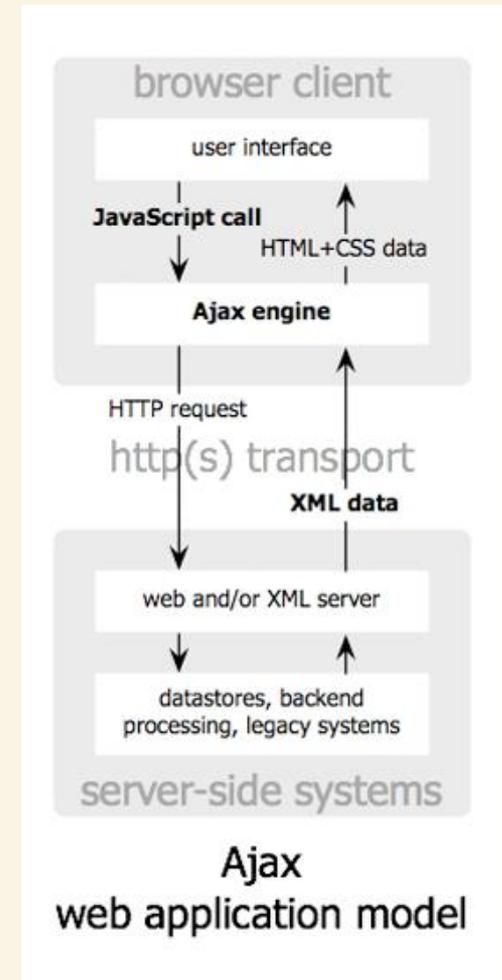
Server stellt Daten bereit

Browser dient der Darstellung

Browser verarbeitet Events

Browser lädt Daten via AJAX vom Server

- + Schnell
- + Seite stets verfügbar
- + Wenig Rechenzeit auf dem Server
- + JS wird im Browser genutzt
- + Keine direkte Bindung zw. Client & Server
- Erstes Laden dauert länger



Strukturierung mittels JavaScript Frameworks

JavaScript Frameworks



- Gibt Struktur vor, kaum Inhalt
 - Imperativer Ansatz
 - One-Way-Databinding
 - Seit 2010 in Arbeit
 - V 1.0: 20.03.2013
-
- Referenzen:



- Viele Hilfsmethoden
 - Deklarativer Ansatz
 - Two-Way-Databinding
 - Seit 2010 in Arbeit
 - V 1.0: 13.06.2012
-
- Referenzen



Basecamp



Youtube on PS3

★ Star 14,740 Version 1.0.0

★ Star 11,226 Version 1.1.5



Model

- Definiert Vorlage für einen Datensatz

Collection

- Menge von Datensätzen
- Kann über eine Schnittstelle synchronisiert werden

View

- Beschreibt die Darstellung eines Datensatzes
- Definiert Events und Reaktionen



View

- Der View wird im HTML-Code definiert
- Eigene HTML-Attribute bilden die Logik ab
- *ng-repeat* erzeugt eine Schleife
- *ng-model* bindet einen Datensatz an ein HTML-Tag

Controller

- Steuert das Verhalten
- Stellt *\$scope* Variable bereit (beinhaltet aktuelle Werte)
- Gültig für einen Bereich im HTML-Code

Beispielapplikation

Umfrage mit erweiterbaren Antworten

- 3 Use Cases (*Abstimmen, Hinzufügen, Filtern*)
- Implementierung jeweils mit AngularJS und BackboneJS

Feststellungen / Ziele

- Unterschiede in der Implementierung
- Einarbeitungsdauer

Struktur, Code, Resultat

Live Präsentation

Erkenntnisse

Unterschiede



BACKBONE.JS

- Auch typische Aufgaben werden selbst programmiert (filter)
- Viel Quellcode, aber 100% getrennt vom JS
- One-Way-Databinding
- Klare Strukturierung, hohe Wiederverwendbarkeit



ANGULARJS by Google

- Hilfestellung für viele Aufgaben (ngrepeat, ngfilter)
- Deklarative Logik im HTML, wenig JS
- Two-Way-Databinding
- Direkte Bindung an den HTML-Code

Erweiterbarkeit



BACKBONE.JS

- Sehr unterschiedliche Qualität bei Erweiterungen
- Erweiterungen verfügbar auf <https://github.com/documentcloud/backbone/wiki/Extensions,-Plugins,-Resources>

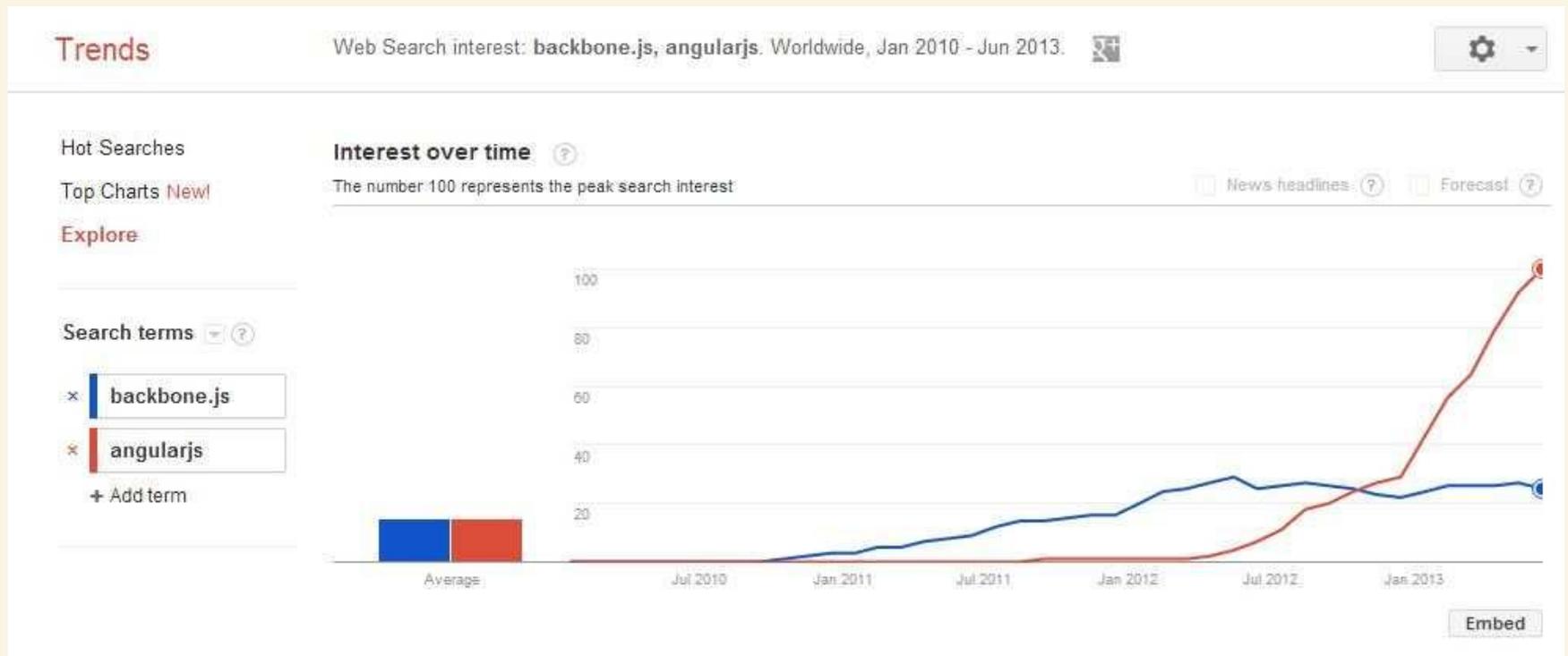


ANGULARJS by Google

- Sehr unterschiedliche Qualität bei Erweiterungen
- Erweiterungen verfügbar auf <http://ngmodules.org/>

Popularität

Google Trends: backbone.js, angularjs (22.06.2013)



Resultat

- Einsatz eines JavaScript Frameworks empfehlenswert
- Entscheidung ob AngularJS oder BackboneJS (oder ein weiteres) muss im Projekt getroffen werden
- Beide Frameworks steigern Organisation, Wiederverwendbarkeit und Wartbarkeit von JavaScript erheblich
- Schnelle Einarbeitung in beide Frameworks möglich

Ausblick

- Trennung von Client und Server Code wird immer wichtiger
 - JavaScript als „Programmiersprache des Webs“
 - Rasche Weiterentwicklung von JS Frameworks
 - Weiterentwicklung von JavaScript
 - Performanceoptimierung in modernen Browsern
- Einsatz eines JavaScript Frameworks wie BackboneJS oder AngularJS muss bei der Entwicklung einer Webapplikation evaluiert werden.**

**Vielen Dank für Ihre
Aufmerksamkeit**