

Sebastian Bothe - DAVID Software GmbH

Web-Applikationen mit JavaScript

NodeJS - AngularJS -
MongoDB

Agenda

- ❖ Einleitung und Demo
- ❖ Überblick: Architektur
- ❖ MongoDB - Datenbank
- ❖ Node.js - Server
- ❖ AngularJS - Client
- ❖ Exkurs: CSS vereinfachen mit less
- ❖ Zusammenfassung / Ausblick

Motivation

- ❖ JavaScript Frameworks immer verbreiteter
- ❖ Neue Konzepte
- ❖ Verlagerung von mehr Verantwortung zurück in den Client
- ❖ Blick über den Tellerrand

JavaScript - klassisch

Client



Struktur



Verhalten

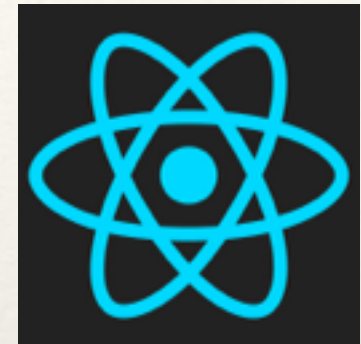


Optik

JavaScript wie es viele kennen...

```
fs.readdir(source, function(err, files) {  
  if (err) {  
    console.log('Error finding files: ' + err)  
  } else {  
    files.forEach(function(filename, fileIndex) {  
      console.log(filename)  
      gm(source + filename).size(function(err, values) {  
        if (err) {  
          console.log('Error identifying file size: ' + err)  
        } else {  
          console.log(filename + ' : ' + values)  
          aspect = (values.width / values.height)  
          widths.forEach(function(width, widthIndex) {  
            height = Math.round(width / aspect)  
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)  
            this.resize(width, height).write(destination + 'w' + width + '_' + filename, function(err)  
              if (err) console.log('Error writing file: ' + err)  
            })  
          }.bind(this))  
        }  
      })  
    })  
  })  
})
```

JavaScript Frameworks



DEMO: Vorschlagsverwaltung

/d3rdud3/Vorschlagswesen



Warum JavaScript als Alternative ?

Warum JavaScript-Frameworks als Alternative ?

- ❖ Delegation von Aufgaben an den Client
- ❖ Einfache Installation über Paketmanager (npm)
- ❖ Geringe Vorbereitungszeit
- ❖ Breite community
- ❖ Schnelle, sichtbare Erfolge
- ❖ Trennung von Entwicklung und Design
- ❖ Keine oder geringe Lizenzkosten

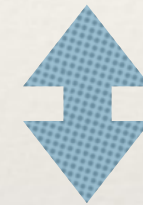
Mögliche Anwendungsbereiche

- ❖ Web-Applikationen (Client side)
- ❖ Performante Webservices
- ❖ Rapid Prototyping

Überblick: Architektur

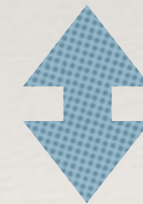
Architektur

Client



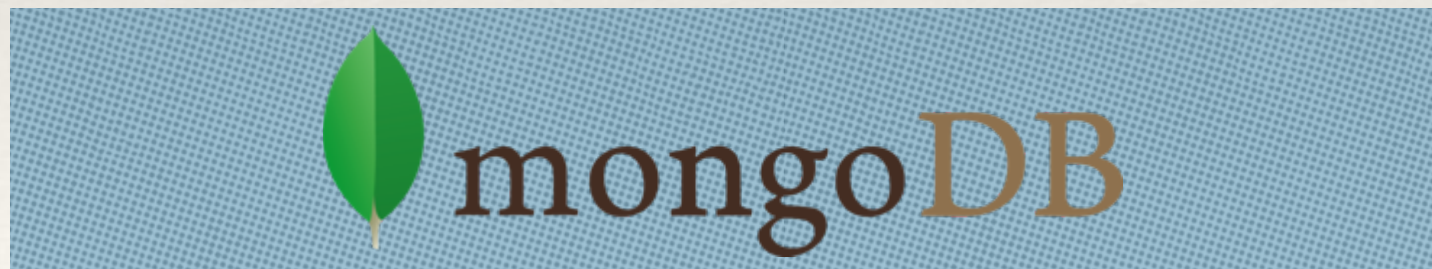
JSON

Server



JSON

Datenbank





mongoDB

mongoDB

MongoDB

- ❖ NoSQL Datenbank
- ❖ Schemafrei / Keine festen Tabellenstrukturen
- ❖ Dokumentenorientiert / Key-Value Storage
- ❖ Einfache Kommunikation aus Node.js (JSON)
- ❖ Umfangreiche Dokumentation

Insert

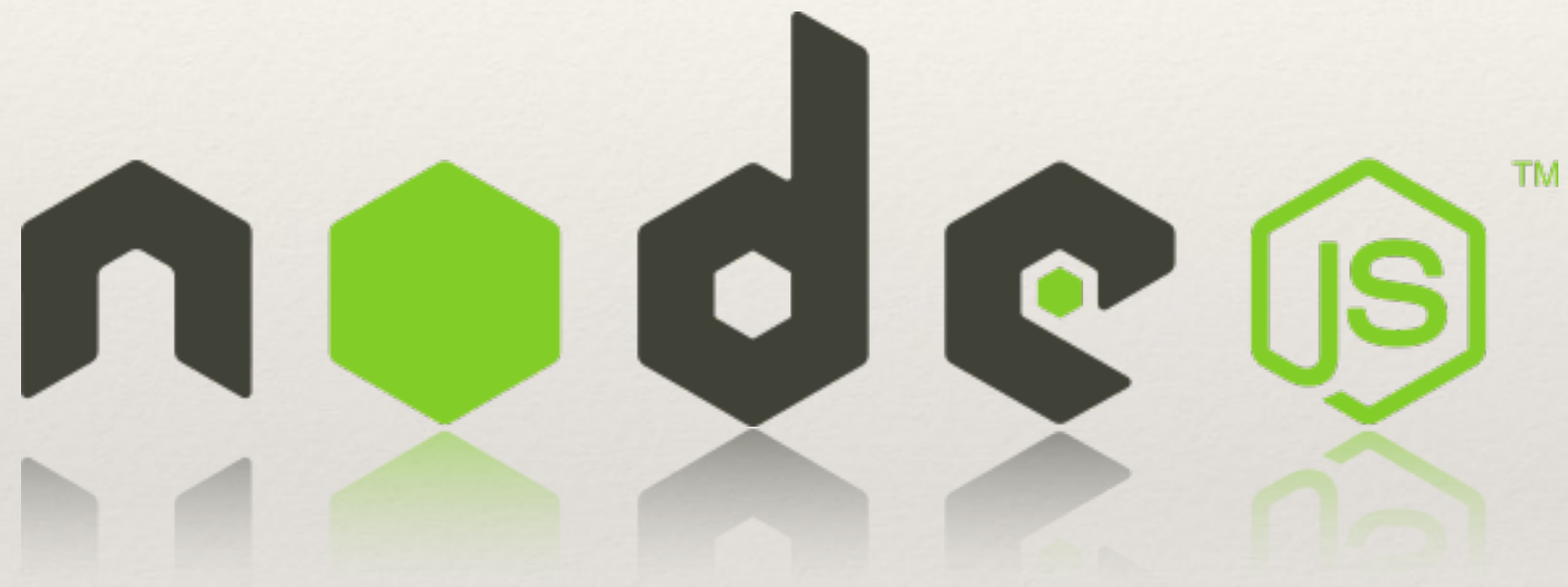
```
db.entries.insert(
{
  author:      'MaxMustermann',
  content:     'suggestion1',
  comments:    [
    {author: 'commenter1', comment: 'very nice'},
    {author: 'commenter2', comment: 'bad one'}
  ],
  rating:      5,
  fav: true,
  archived: false
})
```

Update

```
db.entries.update(
{
  author: "MaxMustermann"
},
{
  $set:
  {
    content: "anotherSuggestion"
  }
})
```

Select

```
db.entries.find({author: 'MaxMustermann'})
```



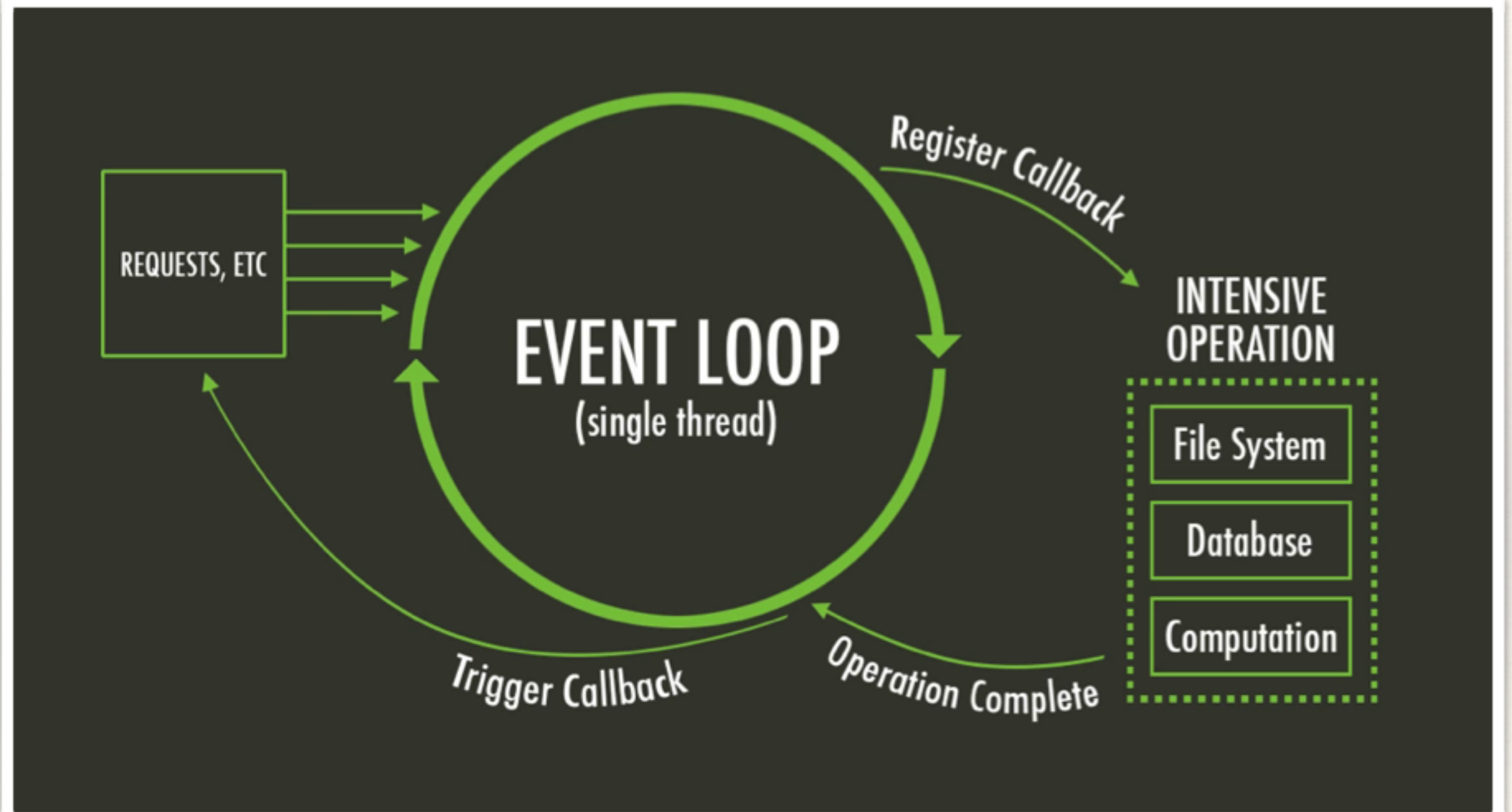
Node.js

- ❖ Basierend auf V8 JavaScript Engine
- ❖ Serverseitiges JavaScript
- ❖ Modularer Aufbau
- ❖ Asynchrone Verarbeitung

Node.js Konzept

- ❖ Non-Blocking I/O
- ❖ Event Loop nimmt Anfragen entgegen
- ❖ Weitergabe zur Ausführung
- ❖ Event Loop läuft im Hintergrund weiter
- ❖ Callbacks werden nach Verarbeitung ausgeführt

Event Verarbeitung



2 Beispiele: Webserver in Node.js

```
var http = require('http');

http.createServer(function (req, res)
{
  res.writeHead(200,
    {'Content-Type': 'text/plain'});
  res.end('Hello DAVID');
}).listen(1337, '127.0.0.1');

console.log('running at http://127.0.0.1:1337/');
```

```
var express = require('express')
var app = express()

app.get('/helloDavid', function (req, res)
{
  res.send('Hello DAVID!')
})

var server = app.listen(3000, function ()
{
  console.log('listening at http://localhost:3000', host, port)
})
```

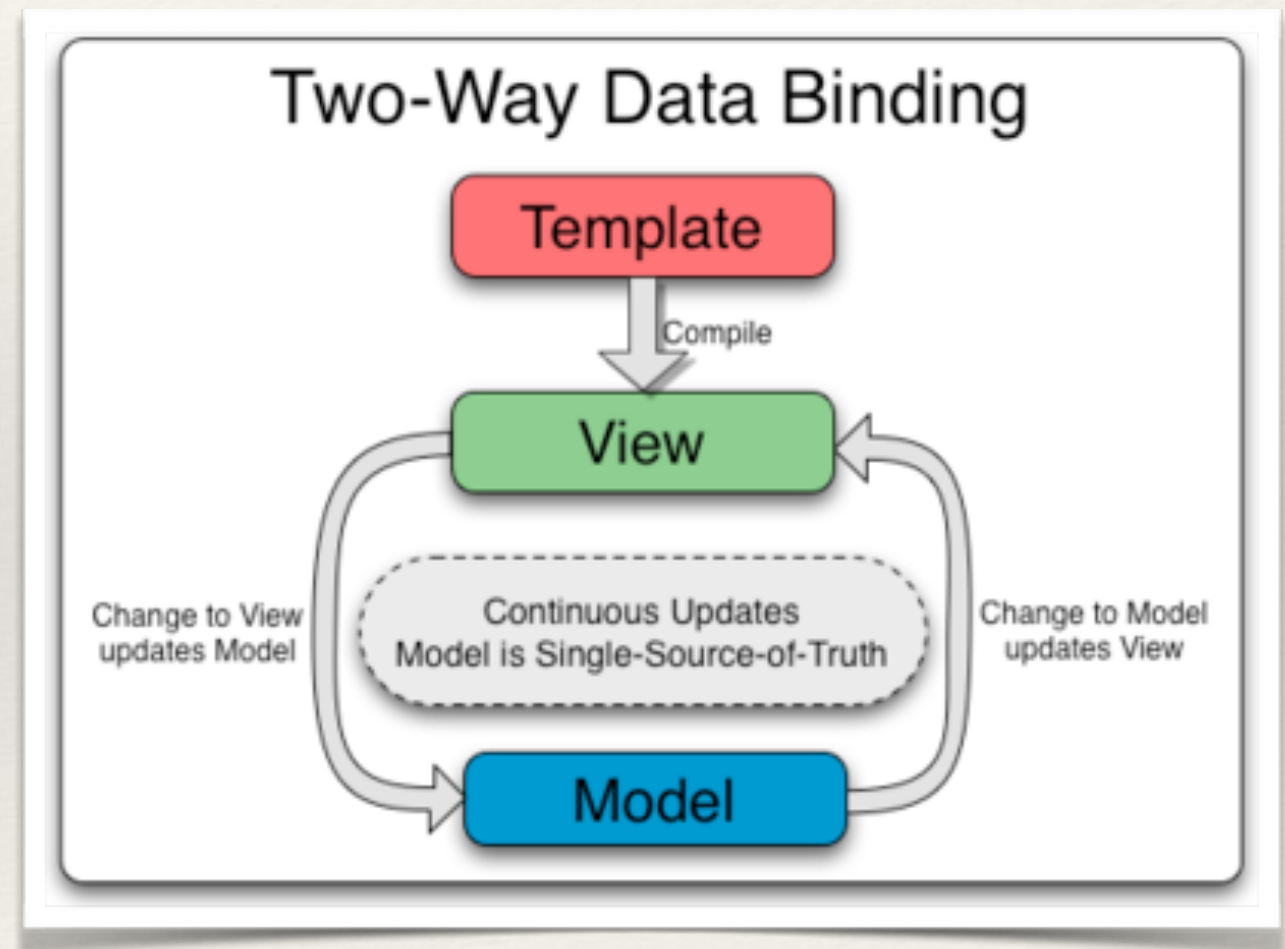



Angular JS

- ❖ Client side MVC Framework
- ❖ Unterstützt die Darstellung von Daten und Verhalten in HTML
- ❖ Modularer Aufbau (Module / Controller / Services / Direktiven)
- ❖ Vereinfachung und Strukturierung der Zusammenarbeit zwischen HTML und JavaScript

Angular JS Konzepte

- ❖ Zwei Wege Datenbindung
- ❖ Realisiert über Scopes



Angular JS Konzepte

- ❖ Direktiven im HTML Code z.B.
- ❖ `<li ng-repeat=„elem in archived“>`
- ❖ Dependency Injection in Controllern / Services
- ❖ Templating / Dynamisches nachladen von Inhalten
- ❖ Eigene Direktiven / Services möglich
- ❖ Unit und Integrationstests möglich

HTML Page

```
<div class="panel-body">
  <ul>
    <li ng-repeat="elem in archived | filter:favoriteSearch | orderBy : 'created' : true">
      <p>
        <span class="bold" >{{elem.created | date:'medium'}}</span>
      </p>
      <p>
        <span class="bold" >Eingereicht von: </span> <span>{{elem.author}}</span>
      </p>
      <p>
        <span class="bold">Vorschlagstext: </span>
      </p>
      <p>
        <span>{{elem.content}}</span>
      </p>
      <div class="archiveDiv">
        <button class="btn btn-default btn-md" ng-click="deleteSuggest(elem._id)">Vorschlag löschen</button>
      </div>
    </li>
  </ul>
</div>
```

Controller

```
angular.module('myApp.controllers').controller('archiveController',
function($scope,$http,notifications)
{
    $scope.archived = [];

    $http.get('http://127.0.0.1:3000/archive').success(function(data)
    {
        $scope.archived = data;
    });
});
```

Exkurs: CSS vereinfachen:



LESS

- ❖ CSS zur Entwicklungszeit vereinfachen und aufräumen
- ❖ LESS „Kompiliert“ zu CSS
- ❖ Clientseitig und Serverseitig einsetzbar
- ❖ Variablen / Regeln innerhalb anderer Regeln / Verschachteln von Regeln / Funktionen

Beispiele:

LESS

```
@myFavoriteColor: #4D926F;

#header {
  color: @myFavoriteColor;
}

h2 {
  color: @myFavoriteColor;
}
```

```
.myStyle {
  width: percentage(0.5);
  color: saturate(#f04615, 5%);
}
```

CSS

```
#header {
  color: #4d926f;
}

h2 {
  color: #4d926f;
}
```

```
.myStyle {
  width: 50%;
  color: #f6430f;
}
```

Tools und mehr Beispiele

- ❖ Twitter recess als command line tool
- ❖ `d3rdud3 / Playground / tree / master / less`




Code der Demo-Applikation

/d3rdud3/Vorschlagswesen



Ausblick:

- ❖ Angular.js on Meteor
- ❖ Realtime web-socket sync mit Angular als frontend MVC Framework
- ❖ Demo bei Interesse
- ❖ /d3rdud3 / AngularOnMeteor 
- ❖ Bei Interesse: <http://angular-meteor.com>

Interessante Links

- ❖ <https://docs.angularjs.org/tutorial>
- ❖ <http://www.nodejs.org/documentation/>
- ❖ <http://www.mongodb.com/de>
- ❖ <http://getbootstrap.com>
- ❖ <http://expressjs.com>

Literatur

DAVID-Wiki: Literaturempfehlungen

Vielen Dank für Ihre
Aufmerksamkeit