

# Webprojekte mit AngularJS und JHipster

## Lessons learned



**Michel Mathis**  
mp technology AG  
November 2015

# Über mp technology



swiss made  
software

- **mp technology AG, Zürich – [www.mptechology.ch](http://www.mptechology.ch)**
- Individual-Software für Intranet und Internet und Mobile seit 2003
- Wir begleiten von Projekte von der Idee zum Erfolg
  - Beratung, Analyse, Konzept, Architektur, Implementierung, Wartung
- Auszug Kundenliste



**MIGROS**



**orell füssli**



- Kontakt: Patrick Pfister, [pfister@mptechology.ch](mailto:pfister@mptechology.ch) , +41 44 296 67 01

# Über mich



- Senior Software Engineer @ mp technology AG, Zürich
  - Web- and Mobile- Software Entwicklung
  - [www.mptechtechnology.ch](http://www.mptechtechnology.ch)
- Meine Schwerpunkte:
  - Java
  - Datenbanken
  - Angular JS

**WE'RE  
HIRING!**

**Java/AngularJS  
SW Engineer**

**iOS Mobile  
SW Engineer**

mathis@mptechtechnology.ch

# Agenda

- Vorstellung des konkreten Projekts
- JHipster
- Der Server
- Der Client
  - AngularJS: Knackpunkte
  - Client-side Build
  - Tuningmöglichkeiten
- Security
- Lessons learned

# Projekt

- [www.insightbee.com](http://www.insightbee.com)
- Zwei Teile
  - e-Commerce system (Mobile und Desktop)
  - Backend
- Start: 30. April 2014 (erster Commit)
- Bis zu acht Entwickler gleichzeitig
- 10238 Commits (Anfang Oktober)
- 61853 Java LOC
- 34758 Javascript LOC



Quality Insights, Quicker.

[PRICING & SAMPLES](#) | [CORPORATE PLANS](#) | [RESOURCES](#) | [CASE STUDIES](#) | [FAQ](#) | [CONTACT](#) | [LOG IN](#)

# Business research just got a whole lot easier

Reports tailor made for you by our analysts. Delivered globally at double the speed and half the price.

[Order a report](#)



[LEARN MORE](#)

## Why use us?

# How it works


[Order a report](#)

Introducing InsightBee



Introducing  
InsightBee





[< Back](#)

## Select a report type



COMPANIES

[View sample](#)



EXECUTIVES

[View sample](#)



INDUSTRY SECTORS

[View sample](#)



BUSINESS QUESTION

[View sample](#)

Enter the name of the company you would like us to research

[Continue](#)

Tip: To enter multiple companies, separate by semicolon (;)

### About company reports

Our company reports are tailored to you and come in three sizes. We will also tailor your report based on the context provided. We will follow up with you directly to clarify details.

"I was impressed with InsightBee. In addition to the smooth online user experience, the team performed well in profiling a private company by combining desk research with targeted interviews."

Vivek Menon  
Marketing Director, JLL






[View sample](#)

Need to order multiple reports? View our [Corporate Plans](#)



Choose the level of detail you require or [Build your own table of contents](#)

		 Quick 4-5 pages Select	 Standard 6-8 pages Selected	 In-depth 15-20 pages Select
Price	CHF ▼	CHF 236	CHF 515	CHF 957
Report samples		<a href="#">View sample</a>	<a href="#">View sample</a>	<a href="#">View sample</a>
Executive summary		×	✓	✓
Company overview		✓	✓	✓
Financials		✓	✓	✓
Key developments		✓	✓	✓
SWOT analysis		×	✓	✓
Company strategy		×	×	✓
Industry analysis		×	×	✓

CHF 515.00

ETA: 19:00 UK Time, 10 Nov [Cancel](#)

Continue

Confirm password

Country

☐ Add an address for invoicing (optional)

## Payment method



We accept Mastercard, Visa, Amex Debit/Credit cards

**Please note:** Your card will be authorised for the amount indicated. The funds will be taken within the next two hours once we have verified the order. It is possible that the actual cost charged will be lower than the amount authorised. In the unlikely event that your request is not achievable for the cost or in the time frame selected we will contact you.

Name on card

Credit card number

Expiry date

 ▼ ▼

CVC

☐ Store payment card securely 

☐ I agree to the InsightBee [terms & conditions](#)

By submitting your details, you are consenting to receiving marketing messages from Evalueserve UK Limited trading as InsightBee. We will not sell your data to any third party. Tick this box if you would prefer not to receive these messages. ☐

We will not sell your data as part of a marketing list to any third party.

**Total CHF 515.00**


ETA: 19:00 UK Time, 10 Nov

Place my order

Earn rewards of up to USD 1000 each time you refer a contact to try InsightBee. [Click here to learn more](#)

×

# My Reports

 New report ▼

All ▼

 Search

TITLE / NAME	DELIVERY DATE			
ABB Ltd.	Due 2 Sep 2015 - 12:00 UK	Share	<a href="#">New report like this</a>	<a href="#">Invoice</a>
Vodafone custom ppt	Due 3 Mar 2015 - 08:00 UK	Share	<a href="#">New report like this</a>	<a href="#">Invoice</a>
Vodafone Plain word	Due 2 Mar 2015 - 13:00 UK	Share	<a href="#">New report like this</a>	<a href="#">Invoice</a>
Rieter	Due 12 Feb 2015 - 08:00 UK	Share	<a href="#">New report like this</a>	<a href="#">Invoice</a>
Disney	28 Nov 2014 - 11:40 UK	Share	<a href="#">New report like this</a>	<a href="#">Invoice</a>

Links

Display 3 ▼ day(s) per page

5 Nov 2015

[Previous](#)

Today

Next


[illegible]

Orders > Order #1001 > Report #1 >

# Report: 1001/1

Open in K-hive

Un-publish

Date created [UK]	11 Nov 2014 - 09:23	Report type	COMPANY
ETA [UK]	12 Nov 2014 - 21:00	Reporting on	Company 2
Order No	 1001	Role	Information technology
Report No	1	Industry	Real estate
Frequency		Report package	In-depth
Scheduling		Report template option	PLAIN_POWERPOINT
Order status	Not published	User comments	
Report status	Published	test	
Report published [UK]	29 Jun 2015 - 13:45		
Invoice	UK14-IB-0001001		
Price [USD]	1080		
Client email			

## Update

Display title

Company 2

Analyst comments

## Publications

Downloaded	No
Files	<a href="#">1-a.docx</a> <a href="#">1-b.docx</a> <a href="#">1-a.docx</a>

The filenames must be prefixed with the report no: "1-"

# Mobile clients

- Durchgängige Verwendung von Responsive Design
- Dasselbe HTML/CSS für alle Clients
- Zusammenarbeit mit Design-Firma
- Übername von HTML-Design: Aufwände bei der Integration von Javascript
- Mehrere unterstützte Plattformen: Testing ist aufwändiger
- Javascript Performance im Auge behalten

# Grüne Wiese für e-Commerce-Projekt

- Anforderungen
  - Modernes Aussehen
  - Einfache, benutzerfreundliche Bedienung
  - Effiziente Entwicklung
- Überlegungen
  - Server-side templating oft ein “Murks”, wenn es dynamischer werden soll
  - Komponentenframework wie Vaadin? Schwierig wegen Design.
  - JavaScript: Erfahrungen vorhanden mit Backbone-basiertem Projekt
  - AngularJS?
  - JHipster: Modernes Java kombiniert mit state of the art Client-Technologie: let's give it a try!



# JHipster

# JHipster

- Yeoman-Generator, mit dem ein Projekt basierend auf Spring Boot und AngularJS erstellt und erweitert werden kann (ähnlich Ruby on Rails)
- Durchgängige Build-Pipeline
- Vereint eine Menge von bewährten Konzepten: Twitter Bootstrap, AngularJS, Sass, Bower, Grunt, Karma, Spring Boot, Maven, Spring Security, ...
- Generatoren zum Erstellen von Controllern, Entities, Services
- Datenbank-Support

# Der Server

# Die Serverseite

- Java 8/Maven
- Spring Boot: Convention over Configuration
- Konfiguration in Java-Klassen
- Embedded Servlet Container
- Schnittstelle zum Client via REST/JSON
- Spring gehört zum Standard-Setup bei uns

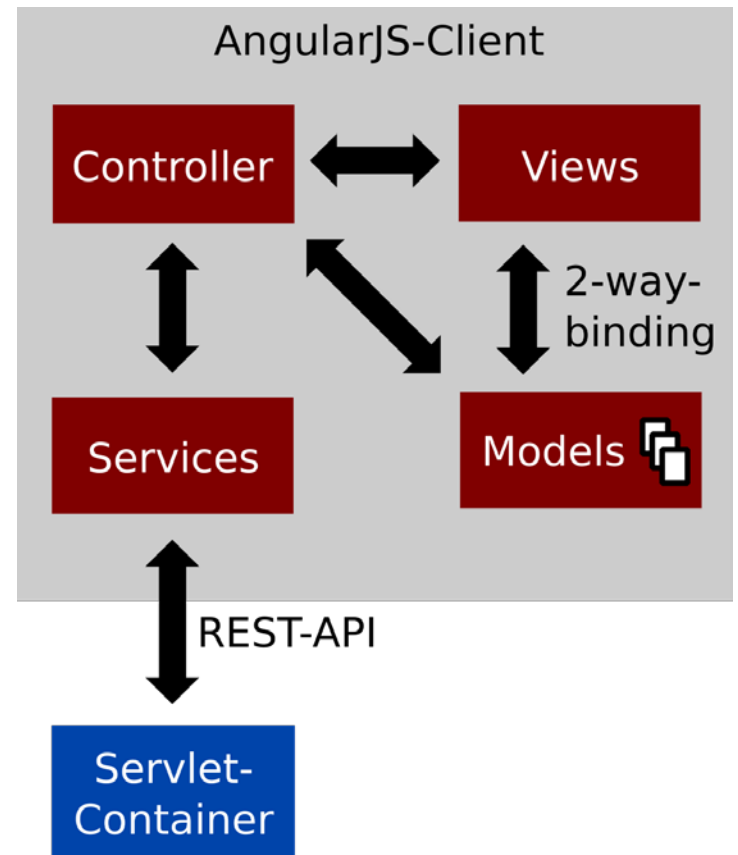
# Datenbank-Support

- Anpassen des Datenbankschemas mit Liquibase
- Standalone-Tool, kann in Java integriert werden
- Datenbank-unabhängige DDL-Beschreibung in XML
- Automatische Tabellen-Generierung aus annotierten Java-Modellen (abhängig von Datenbank-Produkt)

# Der Client

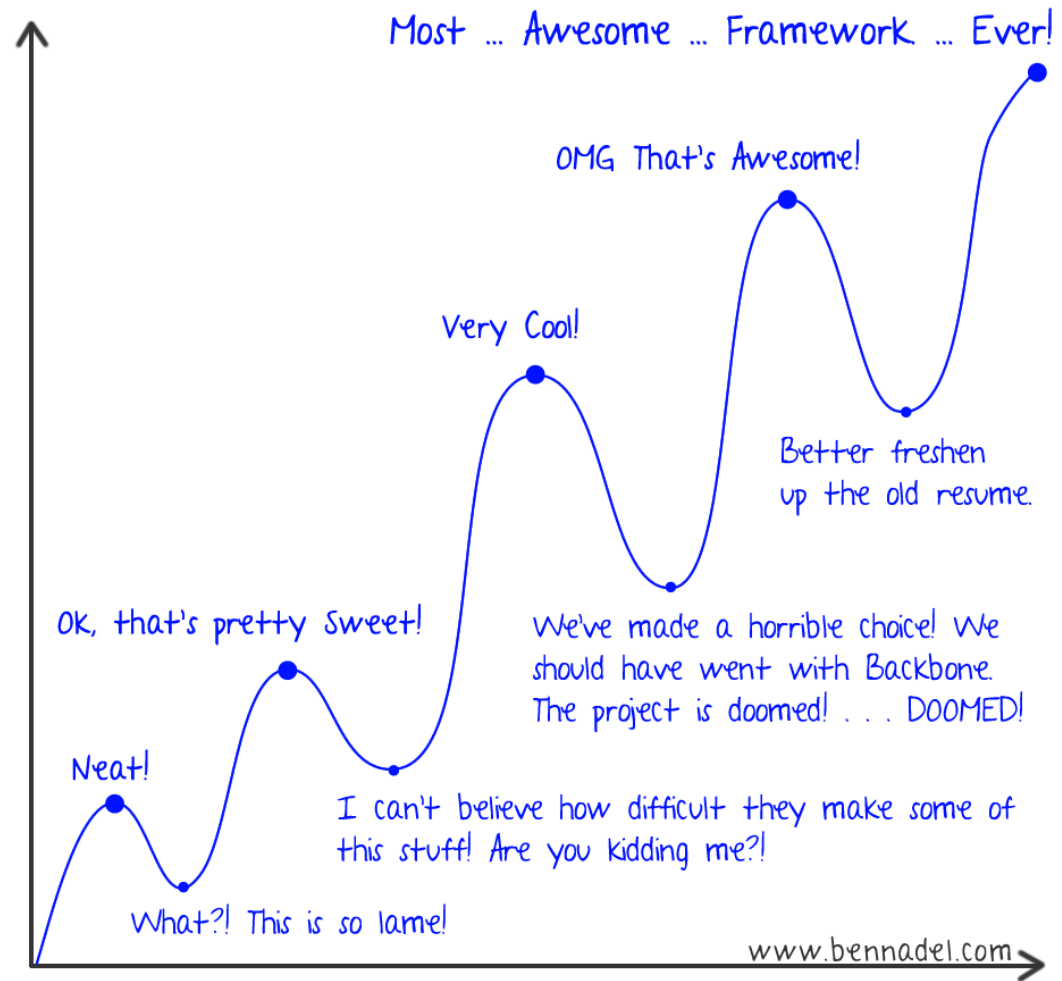
# AngularJS

- Startpunkt: JHipster-App
- Grundstruktur ist relativ übersichtlich: Controller, Services, Views
- Two-way binding zwischen Model und View
- Aber: viel unbekanntes Terrain für Java-Entwickler





# AngularJS



My Feelings About AngularJS Over Time

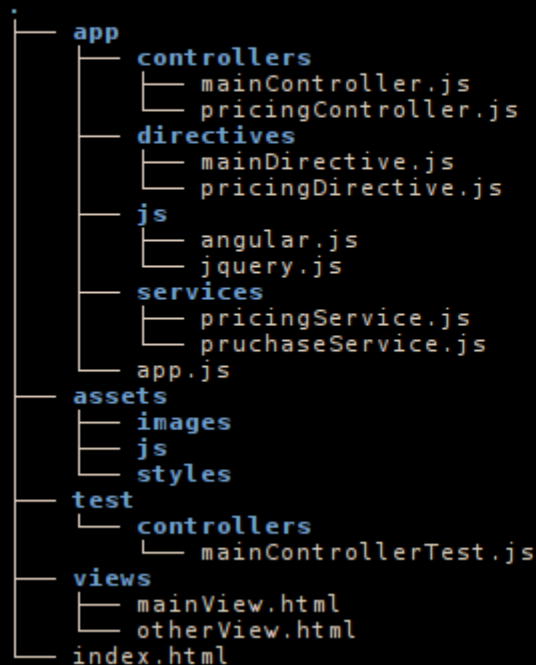
# Knackpunkt: Umgebung kennen lernen

- Ordner, Dateien, Strukturen: Beeinflussen Entwicklung, Build und Testing. Include-Reihenfolge kann entscheidend sein (Angular-Module)!
- Neue Tools, mit denen man vertraut werden muss:
  - Node/Grunt für Client-Side build
  - Bower für dependency management
  - Karma/Jasmine für Client (Unit-) Tests

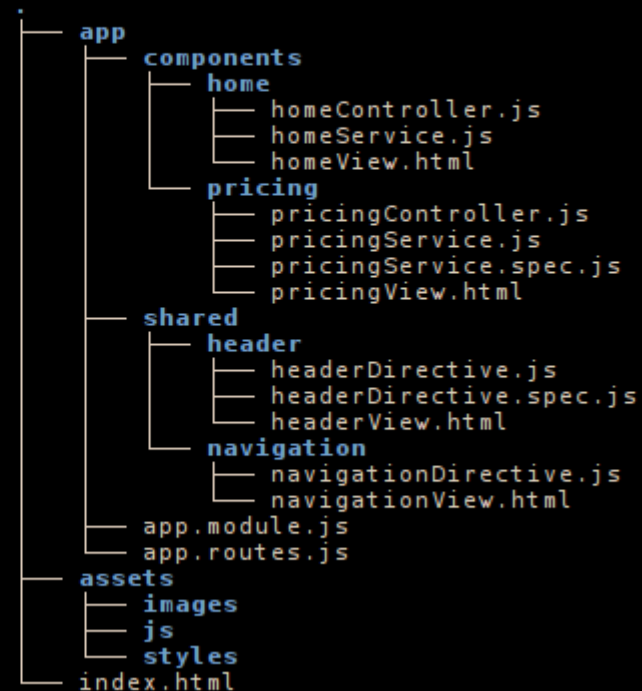
# Knackpunkt: Umgebung kennen lernen

- Ordnen
- Entwerfen
- Realisieren
- Modifizieren
- Neuen
- Multiplizieren

Variante A:



Variante B:  
(empfohlen)



# Knackpunkt: Umgebung kennen lernen

- Ordner, Dateien, Strukturen: Beeinflussen Entwicklung, Build und Testing. Include-Reihenfolge kann entscheidend sein (Angular-Module)!
- Neue Tools, mit denen man vertraut werden muss:
  - Node/Grunt für Client-Side build
  - Bower für dependency management
  - Karma/Jasmine für Client (Unit-) Tests

# Knackpunkt: Angular Views

- Gewöhnungsbedürftige Syntax
- Nested Scopes:
  - Scopes sind nicht direkt sichtbar
  - Vorsicht bei Zuweisungen von primitiven Werten (the dot rule)
- Template-Syntax nicht HTML5-konform
- Prerender.io für SEO-Support
- Template-Markup kann unter Umständen sichtbar werden (bei Fehlern, ngCloak hilft beim Initialisieren)

```

<tr ng-repeat="orderItem in filteredOrderItems(filters.status, filters.searchString)"
    ng-class="{pending: orderItem.orderItemStatus === 'ORDERED', notRead:
isNotRead(orderItem)}">
    <td><span class="nr"></span></td>
    <td class="name" >
        <a ng-show="orderItem.orderItemStatus === 'PUBLISHED' && orderItem.deliverReports ==
true"
            href="/mydownloads/{{orderItem.id}}">{{orderItem.title}}</a>
        <a class="deliveryNotes" ng-show="orderItem.orderItemStatus === 'PUBLISHED' &&
orderItem.deliverReports == false"
            ng-click="showDeliveryNotes(orderItem)" href="{{orderItem.title}}</a>

        <span ng-show="orderItem.orderItemStatus !== 'PUBLISHED'">{{orderItem.title}}</span>
        <label ng-if="orderItem.title != orderItem.name"
            class="glyphicon glyphicon-info-sign" isb-tooltip
            data-isb-tooltip-title="Full Title/Name"
            data-isb-tooltip-description="{{orderItem.name}}">&nbsp;</label>
    </td>
    <td class="ordered_by" ng-if="filters.includeCorporateOrders">
        <div class="circle-avatar filled">
            <span class="initials">{{orderItem.order.account.user.initials}}</span> <span
class="label-wrap"><span
            class="name">
                {{orderItem.order.account.user.firstName}}
                {{orderItem.order.account.user.lastName}}</span></span>
        </div>
    </td>
</tr>

```

# Knackpunkt: Asynchronität

- Beispiel: Anzeigen einer Liste
  - Router definiert Controller und View
  - Controller-Code wird ausgeführt und Template geladen und angezeigt
  - Laden der Daten im Controller, Template wird befüllt, sobald Daten verfügbar
- Alternative: Template erst anzeigen wenn Daten geladen. Aber: User-Feedback nicht vergessen!
- Promise-API ist ein must!



# Knackpunkt

- Beispiel: AngularJS
  - Router definiert
  - Controller- und View
  - Controller und View werden geladen und angezeigt
  - Laden der Daten erfolgt sobald Daten verfügbar sind
- Alternative: AngularJS
- Promise-API

```
$routeProvider.when('/users', {  
  templateUrl: 'views/user/list.html',  
  controller: 'userListcontroller'  
});
```

```
angular.controller('userListController',  
function(userResource) {  
  $scope.users = userResource.query({sort: 'asc'});  
});
```

```
<table>  
  <thead>  
    <tr>  
      <th>Name</th>  
      <th>Actions</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr ng-repeat="user in users">  
      <td>{{user.name}}</td>  
      <td><a  
        href="#/users/{{user.id}}/edit">Edit</a></td>  
    </tr>  
  </tbody>  
</table>
```

# Knackpunkt: Asynchronität

- Beispiel: Anzeigen einer Liste
  - Router definiert Controller und View
  - Controller-Code wird ausgeführt und Template geladen und angezeigt
  - Laden der Daten im Controller, Template wird befüllt, sobald Daten verfügbar
- Alternative: Template erst anzeigen wenn Daten geladen. Aber: User-Feedback nicht vergessen!
- Promise-API ist ein must!

## Knackpunkt: viele Alternativen

- Beispiele in der Dokumentation sind oft trivial
- Hinweise zu best practices fehlen manchmal: z.B. `controller as`
- Javascript ist eine sehr flexible Sprache UND AngularJS bietet manchmal Alternativen: z.B. Dependency Injection
- Angular Style Guide:  
<https://github.com/johnpapa/angular-styleguide>

# Knackpunkt: viele Alternativen

- Beispiel aus dem Style Guide:

```
/* avoid */
function dataService() {
  var someValue = '';
  function save() {
    /* */
  };
  function validate() {
    /* */
  };

  return {
    save: save,
    someValue: someValue,
    validate: validate
  };
}
```

```
/* recommended */
function dataService() {
  var someValue = '';
  var service = {
    save: save,
    someValue: someValue,
    validate: validate
  };
  return service;

  ///////////

  function save() {
    /* */
  };

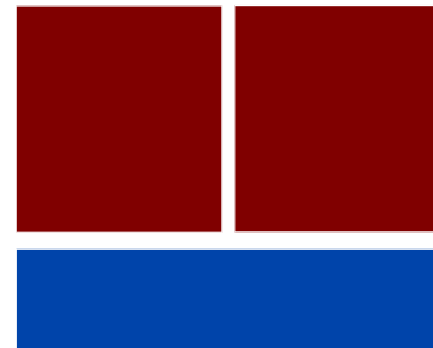
  function validate() {
    /* */
  };
}
```

# Knackpunkt: schöne URLs

- `http://insightbee.com/#/users --> http://insightbee.com/users`
- HTML5 history API to the rescue (history `pushState/popState`)
- Ermöglicht schöne URLs trotz Verarbeitung im Client
- Server-side rewriting nötig
- Automatischer Fallback auf Hash-basierte URLs für alte Browser
- Kann nötig sein für Search Engine Optimization

# Knackpunkt: Client oder Server?

- Überlegungen
  - Security
  - Performance
  - Datenfluss
  - Die Seite zu wechseln ist nicht “Null Aufwand”
- Beispiel: Generierung eines Texts mittels Template
- App-split muss vertikal nicht durchgängig sein



# Tuningmöglichkeiten in AngularJS

- Generell sind Erweiterungspunkte vorgesehen (Error-handling, http-Interceptors)
- Direkter Zugang zum Code
- Es gibt diverse Plugins/Erweiterungen, um einzelne Aspekte zu verändern (z.B. Routing)
- Beispiele
  - Aufstarten der Applikation (angular-deferred-bootstrap)
  - Pre-loaden von HTML-templates (grunt-angular-templates)
  - Automatischer Retry bei Verbindungsabbrüchen



## Client Tests (Unit tests)

- Via Maven in CI integriert
- Test-Runner: Karma
- Test-API: Jasmine
- Headless Browser: PhantomJS (ähnlich zu Chrome)
- Debugging im Browser möglich

## Client-side tools

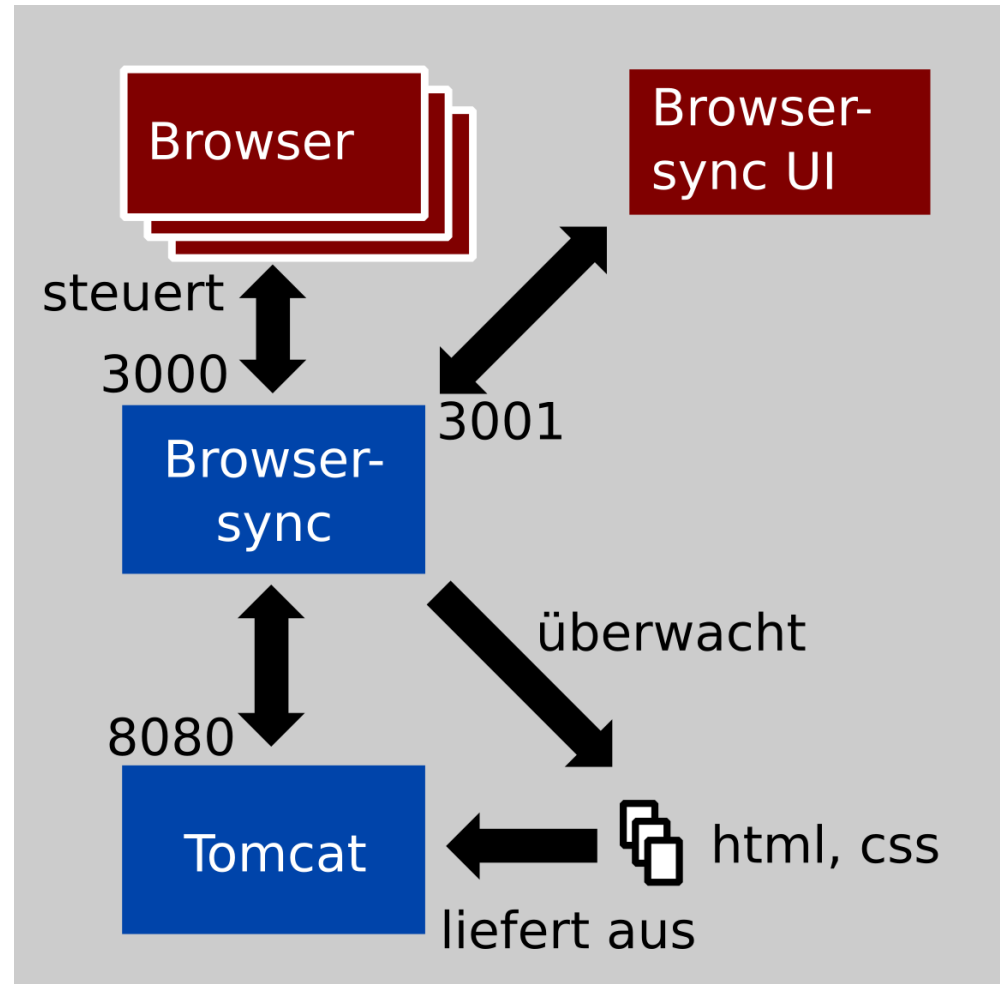
- Node.js, Node Package Manager, Grunt, Bower
- Integriert in Maven
- Gibt es für alle Systeme
- Am besten unterstützt auf Unix/MacOS  
(Windows: Installers, Shell, Path too long)

# Grunt

- Runtime-Optimierung des Codes
- Ziemlich umfangreiche Processing-Pipeline:
  - Vereinen von Dateien
  - Minimize/uglify Javascript code
  - Optimieren von Grafiken
  - Templates HTML-konform machen
  - Cache-Control: Dateinamen werden aufgrund des Inhalts generiert
  - Tests ausführen
- Auch hilfreich während der Entwicklung (Tests, Compile sass)
- Endversion muss getestet werden

# Grunt: Livereload/Browsersync

- Automatischer Browser-Reload
- Browsersync unterstützt mehrere Browser
- Auch für Unit-Tests
- Bei JHipster fertig konfiguriert



# Security setup

- Datensicherheit muss auf dem Server sichergestellt werden (Spring security, konfiguriert mit Config-Builder)
- Rollenkonzept auf dem Client, um richtige Bereiche anzuzeigen
- Client-Implementierung von JHipster: ein bisschen zu kompliziert?
- Client-Assets (JS-Code, Views, Bilder) sind nicht geschützt, enthalten in der Regel aber auch keine kritischen Daten

# Lessons learned: JHipster

- JHipster gut für initialen Setup
- Kontrolle über sämtliche Aspekte bleibt erhalten
- SQL-Tabellengenerierung: nicht ganz ausgereift
- Code-Generatoren praktisch nicht verwendet

# Lessons learned: AngularJS

- Konzepte sind gut zu durchschauen, das führt zu raschen ersten Erfolgen
- Framework ist weit verbreitet: viel Infomaterial vorhanden
- Viele Erweiterungen verfügbar
- Probleme liegen oft im Detail
- Tools are your friends
- Webentwicklung ist erwachsener geworden (aber sie wächst noch immer)

# AngularJS 2.0

- Developer preview (alpha)
- Komponentenbasiert (weniger Code, bessere performance)
- Implementiert in TypeScript (ES6 + Types + Annotations)
- Modern (ausgerichtet auf moderne Browser)
- <https://angular.io/>
- <https://splintercode.github.io/is-angular-2-ready/>



# Interne Umfrage zu AngularJS

- Schwache Typisierung von Javascript wird nicht so problematisch gesehen
- Qualität der Dokumentation wird neutral beurteilt
- Unterschiedliche Sprachen im Projekt sind für die meisten kein Problem
- Vier von fünf würden wieder AngularJS verwenden
- Entwicklungsumgebungen:
  - Java: vorwiegend Eclipse, z.T. IntelliJ
  - Javascript: Atom/Sublime neben der Java-IDE

# Weiterführende interessante Themen

- Multi-Browser testing: browser-sync
- JS code style: JSHint und jscs
- Verbessertes Client-Side routing: ui-router
- <https://github.com/angular/angular.js/wiki/Understanding-Scopes>
- <http://www.toptal.com/angular-js/top-18-most-common-angularjs-developer-mistakes>
- <https://github.com/philipd/angular-deferred-bootstrap>
- Alternativer Start für AngularJS: ng-boilerplate