


Webapp vs. Mobile App -  
gesammelte Berufserfahrungen  
Update

---

LVA Anwendungen in  
Wirtschaft und Technik



# Inhalt

---

→ Einleitung

→ Vorstellung der Personen

→ Web App

→ Three- Tier -Achitectures

→ Mobile App

◆ Challenges

◆ Show-Case: Hybride App

◆ Anguar JS

→ Gesammelte Berufserfahrungen

# Das Team

---

→ Mag. Ulrike Walch

→ 1998 Studium der  
Wirtschaftsinformatik in Wien

→ 18 Jahre Berufserfahrung

→ DI Josef Ornetsmüller

→ 1989 Studium der Informatik in  
Linz

→ 27 Jahre Berufserfahrung

→ [https://www.xing.com/profile/Josef\\_Ornetsmueller](https://www.xing.com/profile/Josef_Ornetsmueller)

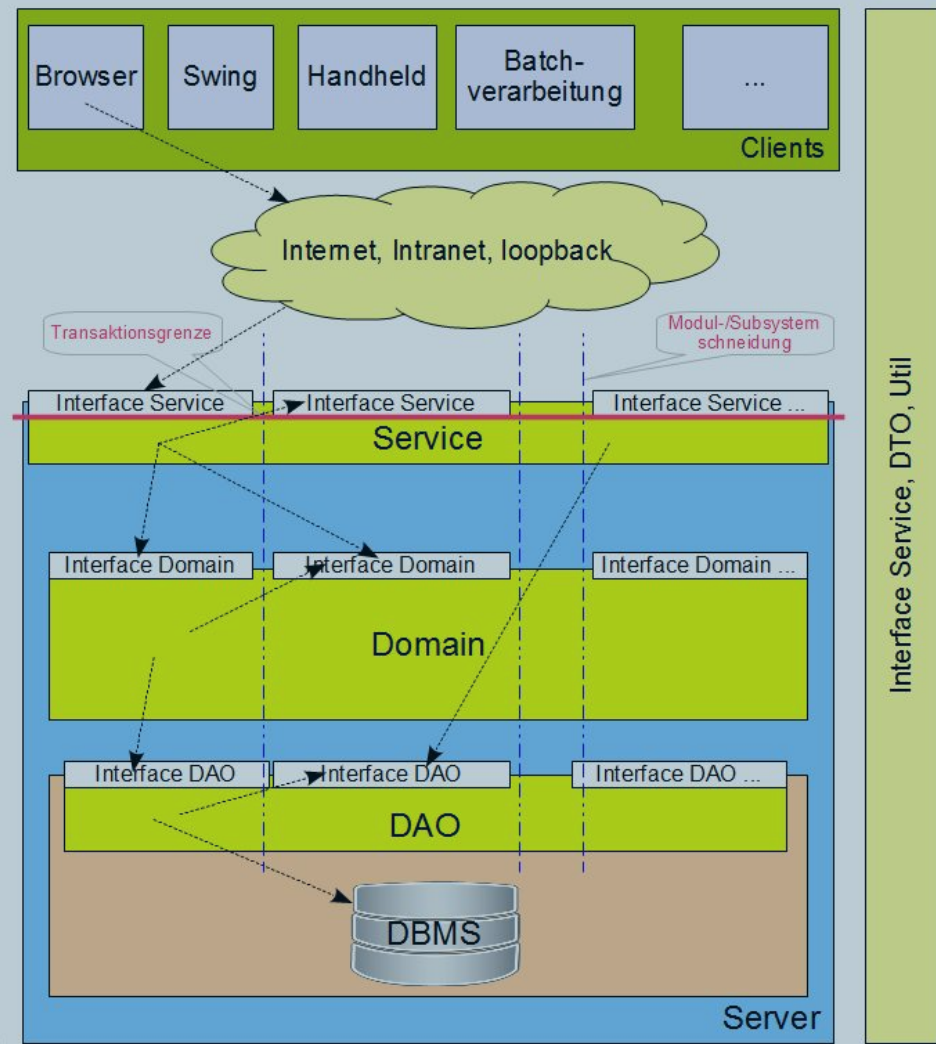


# Webapp

---

3-Tier

# Three Tier Architecture

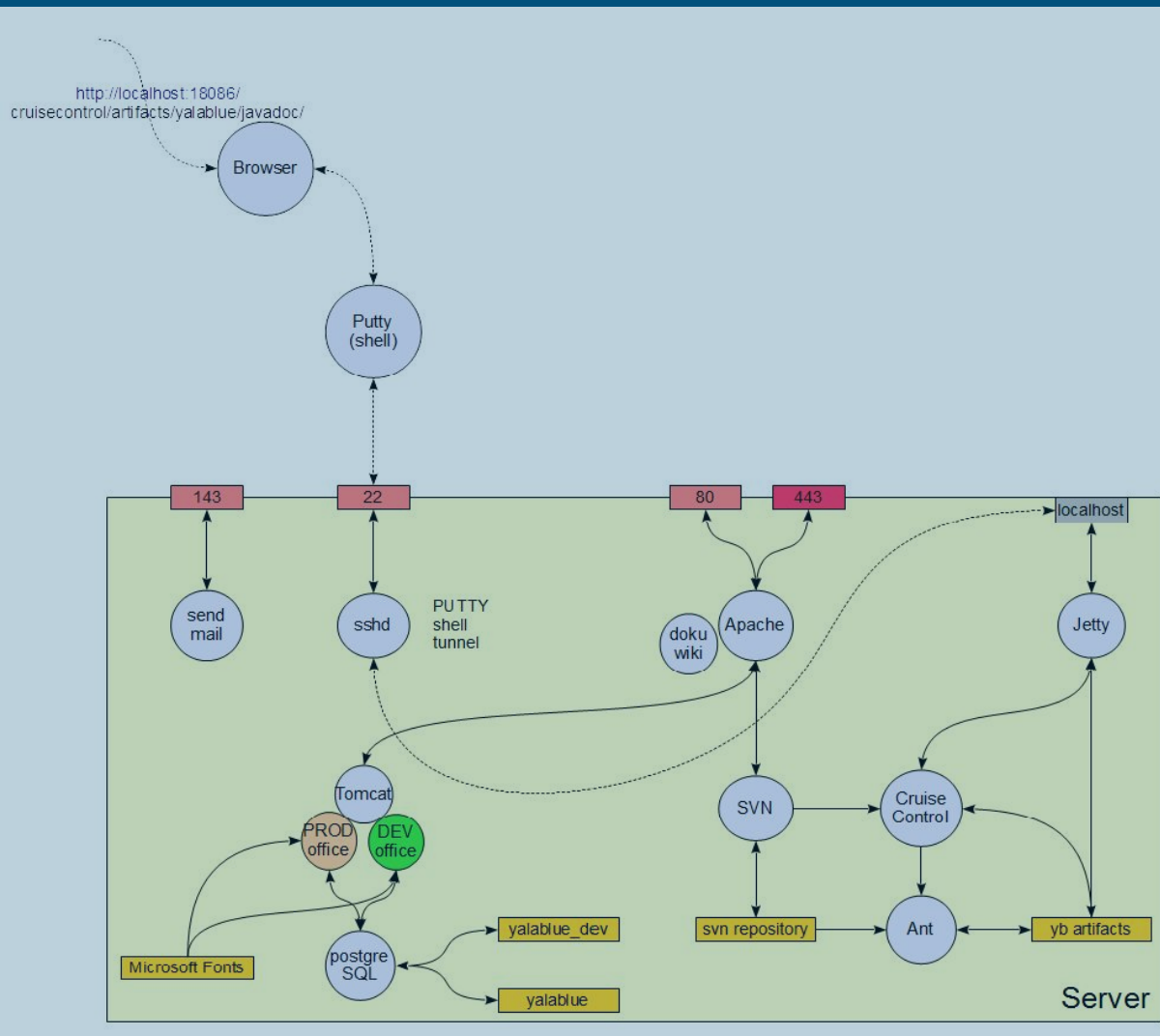


# Build

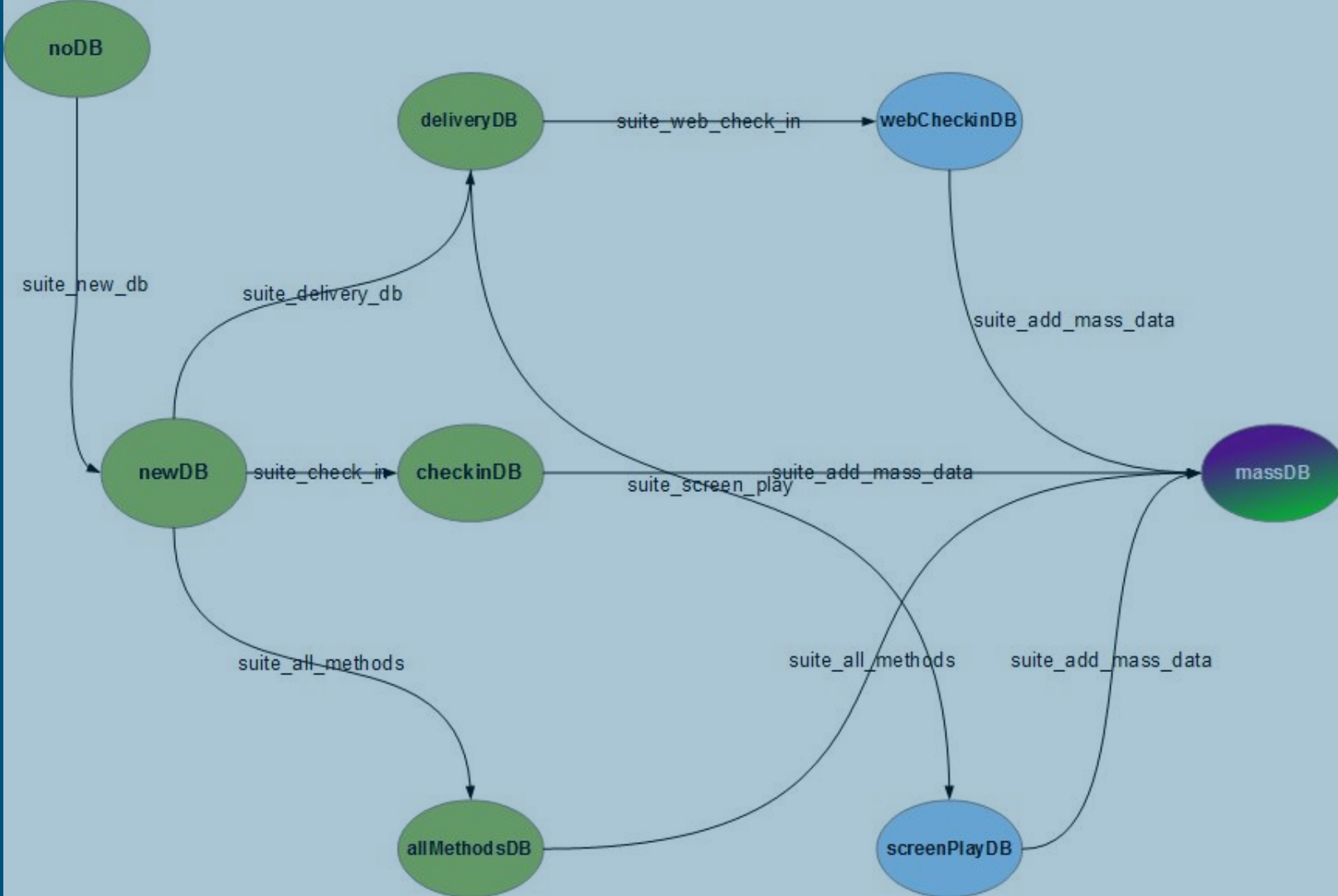
---

- Build und Test laufen schichtweise ab DAO, Domain, Service; Client
- Jenkins, Cruisecontrol; Ant, Maven
- Artefakte wie WAR-files, gefüllte Datenbanken, Javadoc, ... werden erzeugt
- Toolbox für Entwickler: Ant/Maven: clean, build, test, Javadoc, native2ASCII, ...
- Git, SVN, CVS
- Staging: DEV, TEST, PROD
- ...

# Buildserver Akteure



# Test





# 3rd Party Software

---

- Java SDK
- Apache
- Ant
- Maven
- Connection Pooling
- Code coverage
- Jenkins
- DokuWiki
- Eclipse J2EE
- Google Analytics
- Hibernate, Lucene
- JQuery: Calendar, ...
- LIQUIBASE
- Log4J
- Mailing
- Putty
- Spring
- TestNG
- Tomcat
- Wicket
- guava
- postgresSQL
- Eclipse
- Jasper Reports
- ...

# Bugfix/Analyse

---

- Firefox; Firebug; HTML, CSS, JavaScript Debugger; Bildschirmgrößen
- Google Chrome; Inspektor
- Internet Explorer; DOM Explorer
- Code coverage
- Eclipse Debugger, remote
- Log files
- Tool jvisualvm
- Wicket: debug mode
- Eigene eingebaute Tools (Cockpit)
- ...

# REST - HATEOAS | Roy Fielding

---

- REST *Representational State Transfer*
- Light: Webservices; simple stateless architecture
- Heavy: SOAP, WSDL, Registrier DB
- Zugriff reduziert : GET, POST, PUT, DELETE, ...
- `rs/rest/cars?maxResults=4`
- HATEOAS *Hypermedia as the Engine of Application State*
- in/out: interface verhandelbar: version; XML, JSON, HTML, TEXT, ...
- Fehler: HTTP Codes
- Client bekommt Links/Navigation von Server
- Testen: curl, wget
- wildfly RESTeasy



# Mobile App

---

Challenges, Showcase & Angular JS



# Mobile App: Challenges

---

- Klassische Browser-Anwendungen für große Businessanwendungen
- Kennzeichen:
  - ◆ großes Datenvolumen, das aufbereitet und dargestellt werden muss
  - ◆ komplexe Workflows und aufwendige User-Interaktionen
  - ◆ aufwendiges User Interface
- Im einfachsten Fall: Mobile Client für bestehenden Server

# Mobile App: Challenges

---

- Viele verschiedene Hersteller, Betriebssysteme, Betriebssystemversionen
- Geringe Bandbreiten
- Netzverfügbarkeit
- Geringe Devicegröße:
  - ◆ Wieviele Daten können sinnvoll dargestellt werden?
  - ◆ Wie kann der Workflow sinnvoll gestaltet werden (Userinteraktion)?
  - ◆ Anpassung des Designs
- Verfügbare Technologien
- ...

# Mobile App: Grundsätzliche Entscheidungen

---

- Browser-Anwendung portieren ja/nein?
- 
- Wenn Mobile App ja:
  - ◆ Zielgruppe, Zielplattform?
  - ◆ Native oder hybride App?
- 
- Native: Technologiestack durch die Plattform vorgegeben
- Hybrid: HTML5/CSS3 mit native features via Cordova/PhoneGap



# Hybrid Mobile App

---

Showcase





# Google Chrome Remote Debugging

The screenshot displays the Google Chrome Remote Debugging interface. On the left, a mobile application titled "Mein Inventar" is visible. The app features a search bar with the text "im Inventar suchen" and an "NFC" icon. Below the search bar, there are sections for "Meine Materialboxen" (2 items) and "Mein Material" (15 items). The material list includes items like "fischer L-Boxx Spreizdübel S6/SX6", "Klauke Elektroinstallations L-BOXX 102", "fischer Spreizdübel S 6", "fischer Spreizdübel SX 6 X 30 mit Rand", "Hand-Schneidwerkzeug", "Kabelmesser, ohne Klinge", "Presswerkzeug", and "Rohrkabelschuh M10 16 mm², o.". A plus sign icon is visible at the bottom right of the material list.

The right side of the interface shows the Chrome DevTools interface. The "Sources" panel is open, displaying the JavaScript code for the application. The code is located in the file `file:///android_asset/www/index.html#itemList`. The code defines a function `myPurchaseCtrls.js` that handles purchase actions. It includes logic for calculating quantities, translating units, and sending purchase orders via email or a service. The code is as follows:

```
53 var quantity = !$scope.data.materialList[1].quantity ? 1 : $scope.data.materialList[1].quant
54
55 purchaseList += '<p>' + quantity + " " + $translate.instant('UNITS') + " " + $scope.data.mat
56
57
58 if ($scope.data.materialList[1].shopURL) {
59   purchaseList += '<br><a href="" + $scope.data.materialList[1].shopURL + ">" + $translat
60 }
61
62
63 purchaseList += '<p>' + $translate.instant('PURCHASE_ORDER_TEXT3') + '<br>'
64
65 $cordovaEmailComposer.isAvailable().then(function() {
66   var props = {
67     app: 'gmail', // gmail account must be available on the mobile device!
68     attachments: [],
69     subject: $translate.instant('PURCHASE_ORDER_LIST'),
70     body: purchaseList,
71     isHTML: true
72   };
73   $cordovaEmailComposer.open(props).then(function() {
74     // TODO each promise is rejected by plugin
75   }, function(reason) {
76     LogFactory.logMsg(ctrl1, "E-Mail view has been dismissed, reason=" + reason);
77   });
78   ExtItemService.postPurchase($scope.data.materialList).then(function() {
79     MyMaterialDbService.resetPurchase($scope.data.materialList).then(function() {
80       $scope.closePurchase();
81     });
82   });
83 });
84 }, function () {
85   LogFactory.logErr(service, "E-Mail service is not available.");
86 }
87 );
88 }
89 }
90 })
91
```

The "Console" panel at the bottom shows a series of database operations performed by the application:

```
<top frame>
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=6: number|name|doPurchase|src=2050001416285|Rohrkabelschuh M6 10 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=7: number|name|doPurchase|src=2050001416292|Rohrkabelschuh M8 10 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=8: number|name|doPurchase|src=2050001416339|Rohrkabelschuh M10 16 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=9: number|name|doPurchase|src=2050001416353|Rohrkabelschuh M8 25 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=10: number|name|doPurchase|src=2050001416360|Rohrkabelschuh M10 25 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=11: number|name|doPurchase|src=2050001416384|Rohrkabelschuh M10 35 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=12: number|name|doPurchase|src=2050001416414|Rohrkabelschuh M10 50 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=13: number|name|doPurchase|src=4006209700068|fischer Spreizdübel SX 6 X 30 mit Rand|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=14: number|name|doPurchase|src=4006209501061|fischer Spreizdübel S 6|1|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: SELECTED id=15: number|name|doPurchase|src=2050001416377|Rohrkabelschuh M8 35 mm², o. Sichtl.|0|SORTIMO console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: UPDATED id|doPurchase=13|1 console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: UPDATED id|doPurchase=3|1 console-via-logger.js:173
im: MyMaterialDbService TABLE MY_MATERIAL: UPDATED id|doPurchase=3|0 console-via-logger.js:173
```

# Hybride Mobile App: Technologiestack

---

- [Ionic Framework](#): UI-Framework
  - [Angular JS](#): JavaScript Framework; MVC & Testing
  - [Apache Cordova \(Adobe PhoneGap\)](#): Device abstraction layer
  - [SQLite](#): Relational database on smart devices
- 
- [Angular 1](#): 1.5
  - [Angular 2](#): Still Beta

# Round-Trip vs. Single-Page Application

---

## → Round trip:

- Klassische Web-Anwendung, die auf dem HTTP-Protokoll basiert (Request / Response)

## → Single page:

- Initiales HTML-Dokument, Views werden dann über AJAX häppchenweise geladen

# Angular JS

---

- Single Page: index.html
- View change via Angular routing mechanism

```
<script src="js/addItem/services/barcodeService.js"></script>
<script src="js/addItem/services/extItemService.js"></script>
<script src="js/addItem/services/nfcService.js"></script>

<script src="js/options/profileCtrl.js"></script>
<script src="js/options/communityCtrl.js"></script>
<script src="js/options/appConfigCtrl.js"></script>
</head>
```

```
<body class="platform-android platform-cordova platform-webview">
  <ion-nav-view></ion-nav-view>
</body>
</html>
```

# Angular JS: Promises

---

→ [API reference \\$q](#):

◆ “Service that helps you run functions asynchronously”

→ JavaScript = single threaded

→ einzelne Teile im View können je nach Aufgabenstellung später “eingebledet” werden

→ Angular Libraries basieren auf dem Prinzip der Promises

→ z.B. REST calls via [API reference \\$http](#)

# Beispiel REST API

- REST = Representational State Transfer
- Architekturstil für Web Services
- Fokus auf Sicherheit & Skalierbarkeit



## SES Backend API

### customers : the customers API

Show/Hide | List Operations | Expand Operations

POST /customers

Get customers

POST /customers/customer

Create new customer.

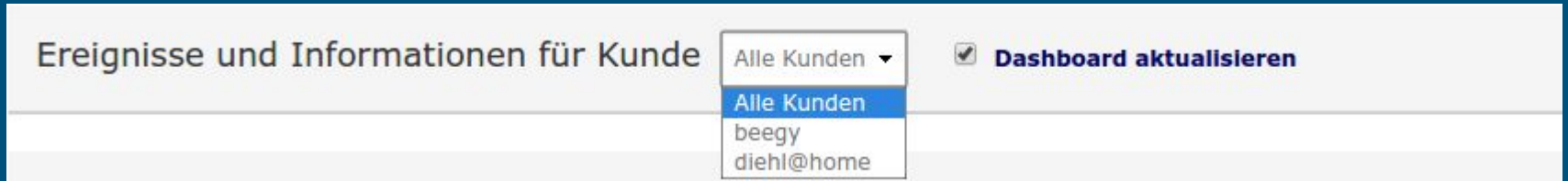
PUT /customers/customer

Updates a customer.

# UI: Befüllen einer DropdownBox

---

- Angular Service \$http liefert ein Promise
- “Hier hast du das Versprechen, dass ein Ergebnis zurückkommt, aber ich kann Dir nicht sagen wann und ob es ein gültiges Ergebnis oder ein Fehler sein wird.”



# REST endpoint for getting all customers

```
.factory('Customers', function(..) {
  return {
    customers : function() {
      var deferred = $q.defer();

      var url      = '/customers';
      var method   = 'POST';

      ServicesHelper.getHttpPromise(url, method).then(function(result) {
        deferred.resolve(result);
      }, function(reason) {
        $log.error('Customers.customers: ' + JSON.stringify(reason));

        deferred.reject([]);
      });

      return deferred.promise;
    }
  };
});
```



# Testing a web app based on Angular JS

---

## Unit Test & Code coverage

Karma

TestRunner for JavaScript

lcov

## e2e Test

Protractor

e2e – Tests based on Selenium  
(Tests written for DOM elements)



# Gesammelte Berufserfahrungen



# Ihre Zukunft (J)

---

- 7 €/Monat für Ihre Zukunft
- Sie sind anderer Meinung
- Grenzen Sie sich ab vs. 'Staubsauger'
- Ihre Idee findet keine Anerkennung, sie sind...
- Wieviele neue 'Brocken' packt das Projekt
- Firmentyp/Manager: Ich darf alles vs. kaum was
- 6 Monate Schonfrist, hat die neue Firma
- Wie motivieren sie ich für den nächsten Tag?
- SCRUM Hype now
- Intrapreneurship/Binnenunternehmertum

# Zuerst kommt... (J)

---

...das Wichtige dann das  
Richtige

...das Richtige dann das  
Wichtige

# Technische Skills (J | U)

---

- Software Design: Pattern, Tiers
- Domain Driven Design; Test Driven Design
- Programmiersprachen
- Betriebssysteme: Windows, Unix/Linux
- Ihre UNIX-Box im Internet verwalten sie selbst
- Staging: DEV, TEST, PROD, ...
- Datenbanken (SQL, DBMS)
- Regular Expression
- CVS, SVN, Git
- Hype: SCRUM
- Server-Technologien:
  - ◆ J2EE EJB
  - ◆ Spring, Hibernate, JPA, JDO
- Client-Technologien:
  - ◆ JSP, JSF, Wicket
  - ◆ HTML, CSS, JavaScript
  - ◆ jQuery Mobile
  - ◆ Angular JS
  - ◆ Cordova/Phone Gap

# Soft Skills (J | U)

---

- KISS
- Focus on point
- Fragen stellen
- Teamfähigkeit
- Veranschaulichung von Problemen und Lösungsansätzen
- Humor
- Artifact über CRUD nachdenken
- Kreativität
- Durchstich
- Kundenkontakt
- Einarbeitung in Fachlichkeit (Domain)
- Probleme in Arbeitspakete zerlegen
- Am falschen Weg umkehren
- Das gemeinsame Ziel verfolgen

# Dokumentation (J)

---

- JavaDoc sinnvoll nutzen
- Nahe am Code
- Tools generieren automatisch
- Sie lesen viel Code
- Sie finden einen alten Dokumentationsfriedhof
- Unwartbar; Zuviel Aufwand
- Confluence und JIRA
- Übertreiben Sie nicht
- Token – Querbeet
- Wie 'tief' – ist 6 Monate im Team
- Literate Programming
- Die Idee
- Was dem nächsten Leser hilft

# Up-to-Date bleiben (J)

---

- Konzeptwissen Halbwertszeit von 10 bis 15 Jahren, Produktwissen anderthalb bis zwei Jahren
- Lesen lesen lesen; Video schauen
- Konferenzen besuchen
- Flexibilität bewahren
- Zertifizieren Sie sich
- Nützen Sie die Gelegenheit zu Technologieevaluierungen, Tutorials
- Fragen Sie Kollegen, profitieren Sie von den Fehlern, die andere schon gemacht haben



# Programmier-Alltag I (U)

---

- Wie schnell kann ich den Fehler reproduzieren?
- Wie schnell bin ich an der „frisch“ zu programmierenden Stelle?
- Wie lange brauche ich um eine „Seite“ Code zu verstehen?
- Wer darf nicht auf Urlaub gehen, damit ich weiter arbeiten kann?
- Bin ich von externen Systemen abhängig (DB, Server) oder kann ich autark arbeiten (lokale IDE)?
- Gibt es fachliche Ansprechpartner, aktuelle fachliche Dokumentation oder ist die Dokumentation der Code mit den „sprechenden“ Variablennamen?..

# Programmier-Alltag II (J)

---

- Ist der Code noch refactorfähig?
- Kann ich sorgenfrei Bugfixen, einen Feature Request abarbeiten?
- Habe ich überwiegend Spaß an der Arbeit?
- Gibt es einen letzten Breakpoint?
- Wie finde ich mich in der Packagestruktur zurecht?
- Typen:
  - ◆ 'IchWeißAlles'
  - ◆ Mitschwimmer
  - ◆ Buddy
  - ◆ ...

# Fragen

---

