

# FWP Aktuelle Technologien zur Entwicklung verteilter Java- Anwendungen

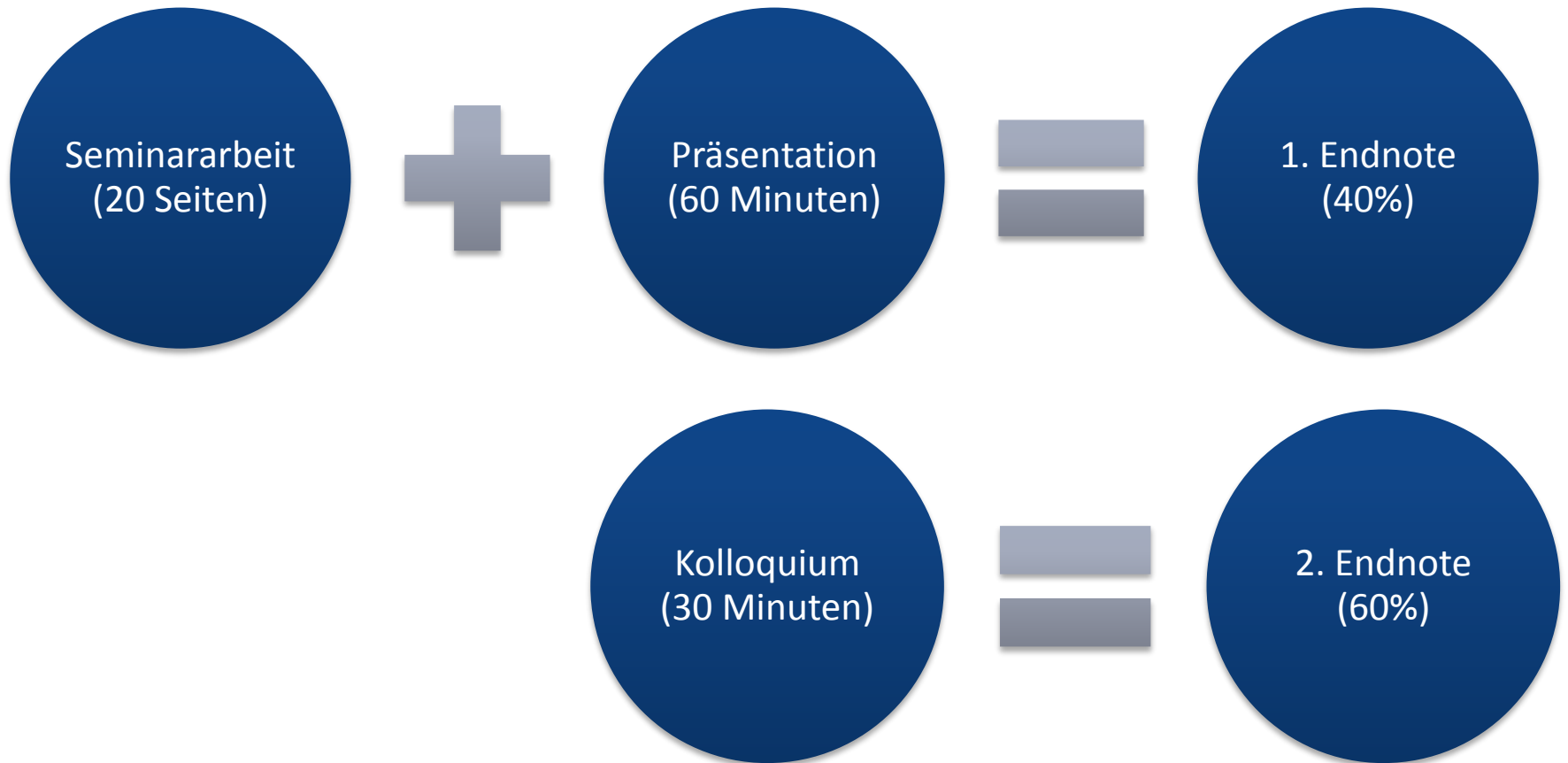
Ablauf, Inhalt und Themen SS2016

# **ABLAUF UND TERMINE**

# Ablauf und Termine

- Wöchentliche Vorlesung (4 SWS)
  - ◉ Start am 18.03.2016
  - ◉ Freitag von 15.15 – 18.30 Uhr Raum 0.005
- Themen werden zu Semesterbeginn vergeben und einzeln bearbeitet
- Pro Thema eine Seminararbeit (20 Seiten) und eine Präsentation (60 Minuten)
- Abschließend Kolloquium (30 Minuten)

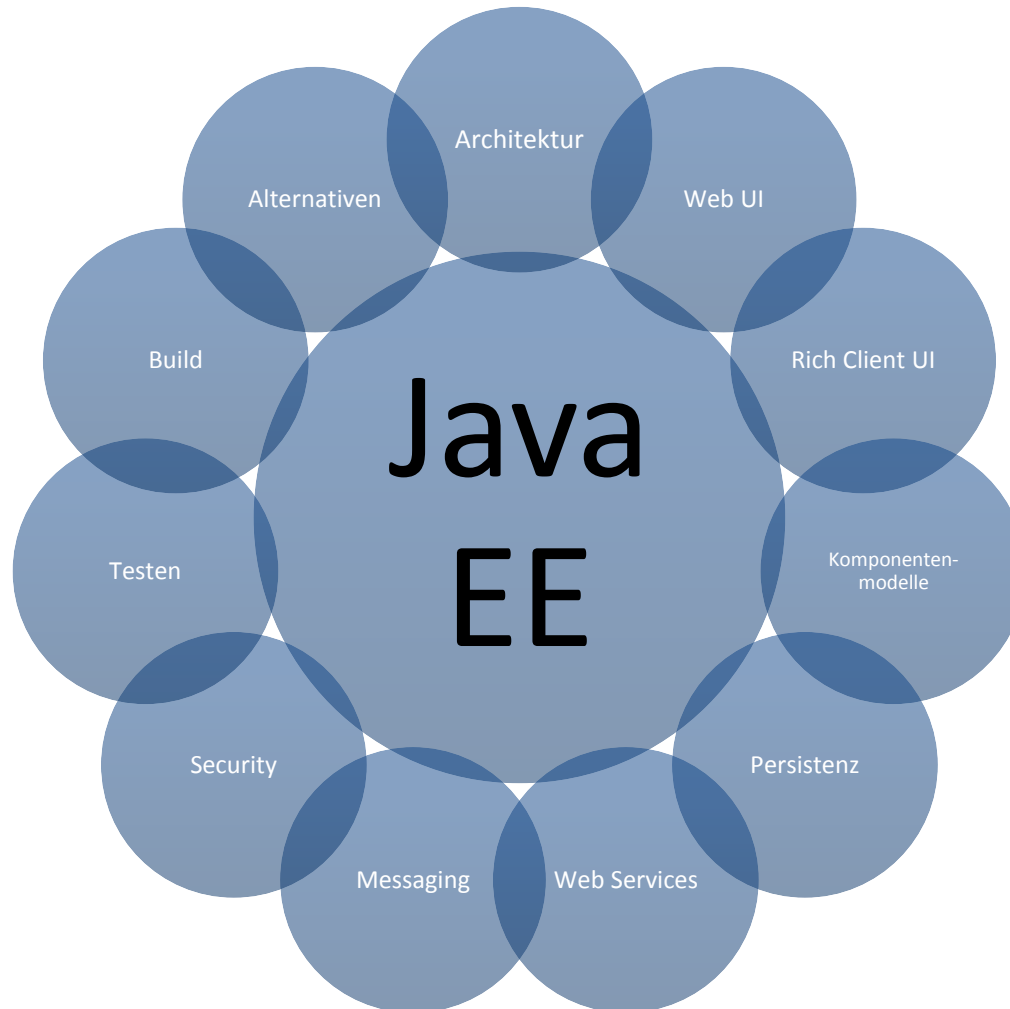
# Meilensteine und deren Bewertung



Das gesamte Themenangebot im Überblick

# **THEMEN 2016**

# Themenschwerpunkte



# Die Themen im Überblick (I)

- Was bringt `JAVA EE 7`?
- `SPRING` und `JEE`: Alternative oder Ergänzung?
- Benutzeroberflächen mit `JSF 2.2`
- Rich Internet Applications mit `REACT`
- Rich Clients mit `ECLIPSE RCP`

# Die Themen im Überblick (II)

- Server an Client - Push mit WEBSOCKETS
- Leichtgewichtige Webservices mit REST
- Messaging mit Java (JMS)
- Integration von Datenbanken mit SPRING  
JDBC
- Datenzugriffskomponenten mit JPA
- NOSQL und Java EE – Ein perfektes Paar?

# Die Themen im Überblick (III)

- NODEJS als Alternative zu JEE
- SICHERHEIT in unternehmenskritischen Applikationen
- Testen mit ARQUILLIAN
- Integration von SOCIAL MEDIA in Java Apps
- ENTITY-CONTROL-BOUNDARY als Komponentenmodell

# Die Themen im Überblick (IV)

- LANG LAUFENDE PROZESSE mit Java EE
- BATCHVERARBEITUNG mit Java
- CONTINUOUS DELIVERY: Keine Angst vor Releases
- Komplexität bewältigen mit DOMAIN DRIVEN DESIGN
- RESPONSIVE WEB-DESIGN - eine Web-App für alle Devices

Die einzelnen Themen im Detail

# **THEMEN 2016**

# Was bringt Java EE 7?

Die Java-Plattform Enterprise Edition (Java EE) ist der Industriestandard für die Erstellung und den Betrieb unternehmenskritischer Anwendungen in Java. Die aktuelle Version ist die Version 7.

- ⊙ Erläutern Sie die wesentlichen neuen Features, die mit Java EE 7 eingeführt wurden.
- ⊙ Vergleichen Sie – sofern möglich – die neuen bzw. weiterentwickelten Features der Version 7 mit denen der Version 6.

## Einstiegsliteratur und Internetquellen:

Eric Jendrock et. al.

*The Java EE 7 Tutorial*

<https://docs.oracle.com/javaee/7/tutorial/index.html>

Oracle September 2014

# Spring und JEE: Alternative oder Ergänzung?

In der Java-Community hat sich Spring als Vertreter der leichtgewichtigen Container für POJO-basierte Komponenten einen festen Platz erobert.

Vergleichen Sie die Architekturkonzepte von Spring 4.x mit denen der JavaEE 7 Spezifikation:

- ⊙ Komponentenmodell
- ⊙ Dependency Injection
- ⊙ Aspect Oriented Programming (AOP)
- ⊙ Integration von Infrastrukturdiensten

Nennen Sie die Schwächen und Stärken der Spring-Plattform im Vergleich zu JEE.

## **Einstiegsliteratur und Internetquellen:**

*Spring Framework Homepage*

<http://projects.spring.io/spring-framework/>

Rod Johnson, *Expert One-on-One J2EE Design and Development*,

Wrox 2002, ISBN: 978-0-7645-4385-2

<http://www.wrox.com/WileyCDA/WroxTitle/productCd-0764543857.html>

Rod Johnson & Jürgen Höller,

*Expert One-on-One J2EE Development without EJB*,

Wrox 2004, ISBN: 978-0-7645-5831-3

<http://www.wrox.com/WileyCDA/WroxTitle/productCd-0764558315.html>

# Webbasierte Benutzeroberflächen mit JSF 2.2

Java Server Faces (JSF) hat sich als offizieller Standard zur Erstellung von webbasierten Benutzerschnittstellen mit Java durchgesetzt.

- ⦿ Schildern Sie die Grundlagen und Konzepte des JSF-Komponentenmodells auf der Basis der JSF 2.2-Referenzimplementierung aus dem Glassfish-Projekt.
- ⦿ Beschreiben Sie anhand eines einfachen Beispiels, wie sich mit JSF 2.2 moderne webbasierte Benutzerschnittstellen mit HTML5 erstellen lassen.

- ⦿ Arbeiten Sie die Stärken und Schwächen von JSF im Vergleich zu JavaScript-basierten RIA-Technologien heraus

## **Einstiegsliteratur und Internetquellen:**

Ed Burns, Chris Schalk

*JavaServer Faces 2.0: The Complete Reference*

McGraw-Hill 2010

ISBN 978-0-07-162509-8

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part III Kapitel 7-16*

<https://docs.oracle.com/javasee/7/tutorial/jsf-intro.htm#BNAPH>

Oracle September 2014

*Oracle's JavaServer Faces Homepage*

<http://www.oracle.com/technetwork/java/javasee/javaserverfaces-139869.html>

# Rich Internet Applications mit React

Die steigenden Anforderungen an web-basierte Benutzeroberflächen haben die Entwicklung von Web-Applikationen radikal verändert. JavaScript-basierte Frameworks wie AngularJS, ExtJS oder React ermöglichen es heute, komplexe Benutzeroberflächen mit der gleichen User Experience wie herkömmliche Desktop-Anwendungen zu erstellen.

- ⊙ Entscheiden Sie sich für ein populäres JavaScript-basiertes WebUI-Framework (außer AngularJS).
- ⊙ Stellen Sie dessen Grundlagen und Konzepte am besten anhand von Beispielen dar.
- ⊙ Bewerten Sie kritisch die Stärken und

Schwächen.

## **Einstiegsliteratur und Internetquellen:**

React Homepage

<https://facebook.github.io/react/>

ExtJS Homepage

<http://www.sencha.com/products/extjs/>

EmberJS Homepage

<http://emberjs.com/>

# Rich Clients mit Eclipse RCP

Mit der Eclipse Rich Client Platform bietet die Eclipse Foundation eine Technologie an, mit der sich Benutzeroberflächen erstellen lassen, die in Bezug auf Erweiterbarkeit und User Experience keine Wünsche offen lassen. Mit der aktuellen Version 4 (Luna) haben nun auch moderne Konzepte wie Dependency Injection in die Eclipse-Welt Einzug gehalten. Widgets lassen sich mit CSS-ähnlichen Dateien gestalten, das Renderkit ist frei konfigurierbar (SWT oder JavaFX).

- ◉ Beschreiben Sie die Grundkonzepte der Eclipse RCP Technologie.

- ◉ Zeigen Sie, wie sich mit Eclipse RCP komplexe Benutzerschnittstellen realisieren, installieren und aktualisieren lassen.
- ◉ Erläutern Sie, wie sich Java EE-Applikationen in diese Benutzerschnittstellen integrieren lassen.

## **Einstiegsliteratur und Internetquellen:**

Lars Vogel

*Eclipse 4 RCP: The complete guide to Eclipse application development*

Lars Vogel 2013, ISBN: 978-3943747072

Lars Vogel

Eclipse 4 RCP – Tutorial

<http://www.vogella.com/tutorials/EclipseRCP/article.html>

Eclipse RCP Wiki

[http://wiki.eclipse.org/Rich\\_Client\\_Platform](http://wiki.eclipse.org/Rich_Client_Platform)

# Server an Client – Push mit WebSockets

Im traditionellen Request-Response-Model von HTTP wird die Kommunikation ausschließlich durch den Client getrieben. Der Server kann keine Daten an den Client senden, die der Client nicht explizit angefordert hat. Dieses bewährte Vorgehensmodell erweist sich bei den heutigen Anforderungen an webbasierte Benutzeroberflächen zunehmend als limitierender Faktor.

Das [WebSocket-Protokoll](#) hebt diese Restriktionen durch das Bereitstellen eines Voll-Duplex-Kommunikationskanals zwischen Client und Server auf. Kombiniert mit anderen Technologien wie HTML5 und JavaScript ermöglicht WebSockets Web-Applikationen mit einer

bisher unerreichten User Experience.

- Schildern Sie die Grundlagen der WebSocket API.
- Beschreiben Sie, wie sich mit Push über WebSockets hochdynamische webbasierte Benutzeroberflächen realisieren lassen.

## **Einstiegsliteratur und Internetquellen:**

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part III Kapitel 18*

<https://docs.oracle.com/javasee/7/tutorial/websocket.htm#GKJIQ5>

Oracle September 2014

JSR 356: Java™ API for WebSocket

Homepage zu JSR 356 auf JCP.org

<https://jcp.org/en/jsr/detail?id=356>

# Leichtgewichtige Webservices mit REST

Das Bereitstellen von Webservices durch und deren Integration in Java-EE Applikationen mit klassischen Methoden bedeutet in den meisten Fällen harte Arbeit. In den letzten Jahren hat sich mit den RESTful-Services eine leichtgewichtige Alternative verbreitet, die mit Java EE 6 erstmals als offizieller Standard eingeführt und mit Java EE 7 nochmals erheblich erweitert worden ist.

- ⊙ Beschreiben Sie die Grundlagen dieser sogenannte RESTful Services und wie diese mit Java EE 7 implementiert werden können.
- ⊙ Erläutern Sie den Unterschied zur

klassischen Webservice-Technologie und zeigen Sie die Stärken und Schwächen der beiden Technologien auf.

## **Einstiegsliteratur und Internetquellen:**

Eric Jendrock et. al.

*The Java EE 7 Tutorial Kapitel 29, 30, 31*

<http://docs.oracle.com/javaee/7/tutorial/jaxrs.htm>

Oracle September 2014

# Messaging mit Java

Die Koppelung über Messaging-Systeme ermöglicht die flexibelste Art der Integration verteilter Anwendungen. In Java steht für den Zugriff auf Messaging-Systeme der hersteller-neutrale JMS-Standard zur Verfügung.

- ⦿ Schildern Sie die Grundlagen von messaging-basierten Systemen, die über JMS integriert werden (Queues, Topics, Connection Factories, Nachrichtentypen)
- ⦿ Zeigen Sie, wie sich mit JMS und EJB Nachrichten über Messaging-Systeme senden und empfangen lassen.

## **Einstiegsliteratur und Internetquellen:**

ORACLE Homepage über JMS

<http://www.oracle.com/technetwork/java/jms-136181.html>

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part IX Messaging*

[https://docs.oracle.com/javasee/7/tutorial/part\\_messaging.htm#GFIRP3](https://docs.oracle.com/javasee/7/tutorial/part_messaging.htm#GFIRP3)

Oracle September 2014

Gregor Hohpe, Bobby Woolf

*Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*

Addison-Wesley Longman,

ISBN 0321200683

# Sicherheit in unternehmenskritischen Applikationen

In unternehmenskritischen Applikationen spielt die Sicherheit vor unzulässigen Zugriffen eine entscheidende Rolle. Die Java EE-Spezifikation sieht hier ein umfangreiches Set von Möglichkeiten vor, Applikationen sicher zu machen.

- ⦿ Beschreiben Sie zunächst die Grundlagen der Sicherheit im Sinne von Java EE
- ⦿ Erläutern Sie anhand von Beispielen, an welchen Stellen ein Java EE-Anwendungen mit welchen Mitteln gehärtet werden kann
- ⦿ Führen Sie auf, wie sich eigene Sicherheitsmechanismen in eine Java

EE-Umgebung integrieren lassen

## Einstiegsliteratur und Internetquellen:

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part X Security*

[http://docs.oracle.com/javaee/7/tutorial/parts\\_eurity.htm](http://docs.oracle.com/javaee/7/tutorial/parts_eurity.htm)

Oracle September 2014

Java EE Management and Security

Technologies Homepage von ORACLE

<http://www.oracle.com/technetwork/java/javaee/tech/management-139662.html>

# Integration von Datenbanken mit Spring JDBC

Auch wenn inzwischen die Java Persistence Architecture (JPA) als Standard für Persistenztechnologie gesetzt ist, gibt es doch Anwendungsfälle, in denen JPA schlichtweg einfach zu umständlich ist oder nicht weiterhelfen kann. Leider ist die JPA zugrundeliegende JDBC-API nicht sonderlich leicht anzuwenden. Glücklicherweise bietet das Spring-Framework hier eine eigene Abstraktionsschicht, die sich wesentlich leichter anfühlt und vor allen Dingen das Aufkommen klassischer Programmierfehler praktisch unmöglich macht.

- ⦿ Beschreiben Sie zunächst die Probleme und Fallstricke der nativen JDBC-API.

- ⦿ Stellen Sie dem die Grundlagen der Spring JDBC API gegenüber.
- ⦿ Erläutern Sie, wie sich mit dem Spring JDBC Template relationale Daten im Rahmen einer Java EE-Applikation integrieren lassen.

## **Einstiegsliteratur und Internetquellen:**

*JDBC(TM) Database Access*

aus der Reihe The Java Tutorials

<http://docs.oracle.com/javase/tutorial/jdbc>

Oracle September 2014

Spring Framework Reference

Kapitel 13 Data access with JDBC

<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/jdbc.html>

# Datenzugriffskomponenten mit JPA 2.1

Die Java Persistence Architecture (JPA) hat sich inzwischen als Standard für die Integration von relationalen Daten in Java Applikationen etabliert.

- ⦿ Beschreiben Sie die Grundlagen von JPA
  - Entities und Entity Manager
  - Persistence Context und Persistence Unit
  - Object-Relational-Mapping (ORM) mit Annotations
  - Queries
- ⦿ Zeigen Sie, wie sich Datenzugriffskomponenten auf Basis von JPA implementieren lassen

## Einstiegsliteratur und Internetquellen:

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part VIII Persistence*

<https://docs.oracle.com/javasee/7/tutorial/part/persist.htm#BNBPY>

Oracle September 2014

Adam Bien

*Real World Java EE Patterns*

*Rethinking Best Practices (2<sup>nd</sup> Ed)*

press.adam-bien.com 2012

ISBN 978-1-300-14931-6

Christian Bauer, Gavin King

*Java Persistence mit Hibernate*

Carl Hanser Verlag GmbH & Co. KG; Auflage: 1

(4. Oktober 2007)

ISBN 978-3446409415

# NoSQL und Java EE – Ein perfektes Paar?

Seit Jahrzehnten sind relationale Datenbanken die erste Wahl, wenn es um ernsthafte Datenspeicherung geht. Unser Informationszeitalter mit immer rasanter wachsenden Datenmengen bringt die relationalen Datenbanken jedoch an ihre Grenzen. Abhilfe schaffen sollen nicht-relationale „NoSQL“ Datenbanken, die der Datenflut mit neuen Konzepten entgegentreten.

- ⦿ Erläutern Sie zunächst die grundlegenden Konzepte von NoSQL-Datenbanken und vergleichen Sie diese mit der relationalen Technik.
- ⦿ Wählen Sie aus dem Angebot eine

bekannte NoSQL Datenbank aus.

- ⦿ Beschreiben Sie deren besonderen Eigenschaften.
- ⦿ Zeigen Sie, wie sich diese NoSQL-Datenbank in eine Java EE Anwendung integrieren lässt.

## **Einstiegsliteratur und Internetquellen:**

Prasmod J. Sadalage, Martin Fowler

*NoSQL Distilled*

Addison Wesley Pearson Education 2013

ISBN 978-0-321-82662-6

# NodeJS als Alternative zu JEE

In den letzten Jahren hat JavaScript als Programmiersprache für server-basierte eine wahre Renaissance gefeiert. In diesem neuen Szenario ersetzt NodeJS als serverseitige JavaScript Runtime den klassischen JEE-Application-Server.

- ⊙ Erläutern Sie zunächst die grundlegenden Konzepte von NodeJS und vergleichen Sie diese mit der klassischen JEE-Technik.
- ⊙ Betrachten Sie hierbei bitte genau die Aspekte Transaktions-Management und Security.
- ⊙ Bewerten Sie kritisch die Stärken und Schwächen von NodeJS heraus.

- ⊙ Skizzieren Sie die Einsatzszenarien, in denen NodeJS glänzen kann.

## **Einstiegsliteratur und Internetquellen:**

Node.js Homepage  
<https://nodejs.org/>

# Testen von Java Applikationen mit Arquillian

Arquillian ist eine innovative und erweiterbare Test-Plattform auf Java-Basis, die es Entwicklern ermöglicht, einfach automatisierte Integrationstests, funktionale Tests und Akzeptanz-Tests für server-seitige Java-Komponenten zu erstellen.

- ⦿ Beschreiben Sie zunächst, wo bei der Entwicklung server-seitiger Java-Anwendungen die Herausforderungen bezüglich automatisiertem Testen liegen.
- ⦿ Schildern Sie anschließend die wesentlichen Konzepte und Eigenschaften von Arquillian und wie

sich diese von denen klassischer Testmethoden unterscheiden.

- ⦿ Demonstrieren Sie den konkreten Einsatz von Arquillian mit einem Beispiel.

## **Einstiegsliteratur und Internetquellen:**

Arquillian Homepage

<http://arquillian.org/>

# Integration von Social Media in Java Applikationen

Soziale Netzwerke wie Facebook, Google+, LinkedIn, XING sind feste Bestandteile unserer heutigen Netzkultur und durchdringen praktisch alle Bereiche unseres täglichen Lebens.

Dementsprechend häufen sich auch die Anforderungen, klassische Unternehmensanwendungen mit diesen sozialen Medien zu vernetzen.

- ⊙ Beschreiben Sie zunächst Szenarien, in denen die Integration von Anwendungen mit sozialen Netzwerken Sinn macht.
- ⊙ Schildern Sie dann, wie diese Integration mit Java-Mitteln

angegangen werden kann.

## **Einstiegsliteratur und Internetquellen:**

Social Media API for the Java Platform

<http://java.net/projects/java-social/pages/Home>

# Entity-Control-Boundary als Komponentenmodell für Geschäftskomponenten

Die konsequente Verwendung eines einheitlichen Komponentenmodells stellt üblicherweise einen wesentlichen Erfolgsfaktor für die Entwicklung großer Java EE-Anwendungen dar. Im Bereich der Geschäftskomponenten hat sich das Entity-Control-Boundary-Modell als vielversprechender Kandidat erwiesen: es ist zum Beispiel wesentlicher Bestandteil der Component Architecture 4 (CA4) der BMW Group.

- Beschreiben Sie zunächst, was sich hinter diesem Komponentenmodell verbirgt und welche Rollen die einzelnen Beteiligten spielen.

- Zeigen Sie auf, wie sich dieses Komponentenmodell mit Java EE-Mitteln umsetzen lässt.
- Bewerten Sie das Komponentenmodell bezüglich seiner Stärken und Schwächen.

## **Einstiegsliteratur und Internetquellen:**

Adam Bien

*Real World Java EE Patterns*

*Rethinking Best Practices (2<sup>nd</sup> Ed)*

press.adam-bien.com 2012

ISBN 978-1-300-14931-6

# Lang laufende Prozesse mit Java EE

In fast jeder nicht-trivialen Applikation muss man früher oder später Aktivitäten ausführen, die bezüglich ihrer Laufzeit den gewöhnlichen Rahmen sprengen (> 120 Sekunden). Bis Java EE 7 gab es keine spezifischen APIs, solche Aufgaben mit Hilfe des Executor-Frameworks sicher auf eine standard-konforme Art und Weise zu lösen.

Hier können die mit Java EE 7 neu hinzugekommenen `Concurrency Utilities` (JSR-236) Abhilfe schaffen: `ManagedExecutorServices` und `ManagedScheduledExecutorServices` lassen sich nun als Ressourcen im Application Server konfigurieren und aus Applikationen heraus nutzen.

- ◉ Beschreiben Sie die wesentlichen Features der `Concurrency Utilities` API.
- ◉ Erläutern Sie, wie sich damit langlaufende Prozesse standard-konform umsetzen lassen.
- ◉ Vergleichen Sie Lösungen auf Basis der `Concurrency Utilities` mit `@Schedule` EJBs und `@Asynchronous` EJB-Aufrufen.

## **Einstiegsliteratur und Internetquellen:**

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part XI Kapitel 56*

<https://docs.oracle.com/javaee/7/tutorial/concurrency-utilities.htm#GKJIQ8>

Oracle September 2014

JSR 236: `Concurrency Utilities` for Java™ EE  
Homepage zu JSR 236 auf JCP.org

<https://jcp.org/en/jsr/detail?id=236>

# Batchverarbeitung mit Java

Die Verarbeitung von großen Datenmengen ist nicht mehr ausschließlich mainframe-basierten Systemen vorbehalten, sondern wird aus Kostengründen zunehmend auch auf java-basierten Plattformen durchgeführt. Mit Java EE 7 ist nun erstmals eine Standard-API für Batchverarbeitung veröffentlicht worden.

Beschreiben Sie zunächst die Grundlagen der Batchverarbeitung:

- ⊙ Massendatenverarbeitung
- ⊙ Jobs / Steps
- ⊙ Checkpoint / Restart

Erläutern Sie, wie sich mit Java Batch Processing Massendatenverarbeitung umsetzen lässt.

## **Einstiegsliteratur und Internetquellen:**

Eric Jendrock et. al.

*The Java EE 7 Tutorial Part XI Kapitel 55*

<https://docs.oracle.com/javaee/7/tutorial/batch-processing.htm#GKJIQ6>

Oracle September 2014

# Continuous Delivery: Keine Angst vor Releases

Ein neues Software-Release heraus zu bringen ist oft ein schmerzhafter, riskanter und zeitfressender Prozess. Der Continuous Delivery-Ansatz versucht hier durch die Automatisierung des Build-, Installations- und Testprozesses sowie eine verbesserte Zusammenarbeit zwischen Entwicklern, Testern und Operatoren Abhilfe zu schaffen.

- ⊙ Schildern Sie, wie sich mit **Continuous Delivery** Änderungen mit deutlich verkürzter Durchlaufzeit in Produktion bringen lassen.
- ⊙ Beschreiben Sie den wesentlichen Aufbau einer **Deployment Pipeline**, die

das Kernstück von Continuous Delivery bildet.

- ⊙ Stellen Sie eine Liste von möglichen Produkten zusammen, mit denen sich die einzelnen Stationen einer **Deployment Pipeline** realisieren lassen.

## **Einstiegsliteratur und Internetquellen:**

Jez Humble, David Farley  
*Continuous Delivery*  
Addison Wesley 2011  
ISBN 978-0-321-60191

Jez Humble, David Farley  
Continuous Delivery Homepage  
<http://continuousdelivery.com>

# Komplexität bewältigen mit Domain Driven Design

Domain Driven Design (DDD) hat sich in der Softwareentwicklung-Community inzwischen als fester Begriff etabliert. DDD bietet ein Vorgehensmodell auf Basis von etablierten Patterns, mit deren Hilfe auch komplexe Domänenmodelle in großen Unternehmen verständlich, beherrschbar und änderbar bleiben. Bei DDD liegt der Fokus insbesondere auf einer allumfassenden Sprache, die sowohl die fachlichen als auch die technischen Experten sprechen und leben und die vom Domänenmodell direkt umgesetzt wird.

- ◉ Beschreiben Sie kurz die grundlegenden Konzepte von DDD.

- ◉ Zeigen Sie anhand von essentiellen Patterns aus der DDD-Welt, wie DDD bei der alltäglichen Entwicklung von Applikationen weiterhelfen kann.

## **Einstiegsliteratur und Internetquellen:**

Eric Evans

*Domain Driven Design*

*Tackling Complexity in the Heart of Software*

Addison-Wesley Longman, Amsterdam, 2003

ISBN 978-0-321-12521-7

Vaughn Vernon

*Implementing Domain Driven Design*

*Tackling Complexity in the Heart of Software*

Addison-Wesley 2013

ISBN 978-0-321-83457-7

Homepage der DDD Community

<http://dddcommunity.org/>

# Responsive Web-Design – eine App für alle Devices

Mobile Endgeräte haben dem klassischen Desktop-PC als Endgerät für das Internet längst den Rang abgelaufen. Daher muss man heutzutage beim Design einer Web-Applikation berücksichtigen, dass die Web-Applikation auch auf Endgeräten mit unterschiedlichen Bildschirm-auflösungen optimal bedienbar bleiben. Diesen Design-Ansatz bezeichnet man als Responsive Web Design.

- ⦿ Beschreiben Sie kurz die grundlegenden Konzepte von Responsive Web Design.
- ⦿ Erläutern Sie, wie sich mit Fluid Grids, JavaScript und CSS Responsive Web

Design umsetzen lässt.

- ⦿ Zeigen Sie anhand eines Frameworks Ihrer Wahl (Bootstrap etc.), wie man Responsive Web Design in eine Web-Applikation integrieren kann.

## **Einstiegsliteratur und Internetquellen:**

Bootstrap Homepage

<http://getbootstrap.com/>

# Kontakt



## **Michael Theis**

Lehrbeauftragter Hochschule München

email [michael.theis@hm.edu](mailto:michael.theis@hm.edu)

mobile + 49 170 5403805

web <http://www.tschutschu.de/Lehrauftrag.html>