

Course code: **JEE7/JSF**

Course title: **Modern web applications built with JEE7 (JSF2.2, EJB3.2, JPA2.1, CDI1.1, AJAX)**

Days: 5

Description:

Course intended for:

The training is aimed at web application developers, willing to supplement their skills on latest technologies, as well as for people who know Java and who want to start an adventure with web applications using latest technologies. The training is carried out using the latest Java Enterprise Edition - version 7. For those wishing to work with older and established technologies we recommend WEB2 training.

Course objective:

Participants will learn how to design and implement applications using Java EE platform technologies. The goal of the training is to gain the ability to create scalable applications, in accordance with the arcane art: from business components to highly ergonomic and functional interfaces suited to the WEB 2.0 trends.

In particular: Participants will learn in JSF2.2 in details, including RichFaces, IceFaces. They will learn how to design and implement the business logic layer using EJB 3.2 combined with the JPA2.1 mechanisms of object-relational mapping. They will also learn the CDI specification.

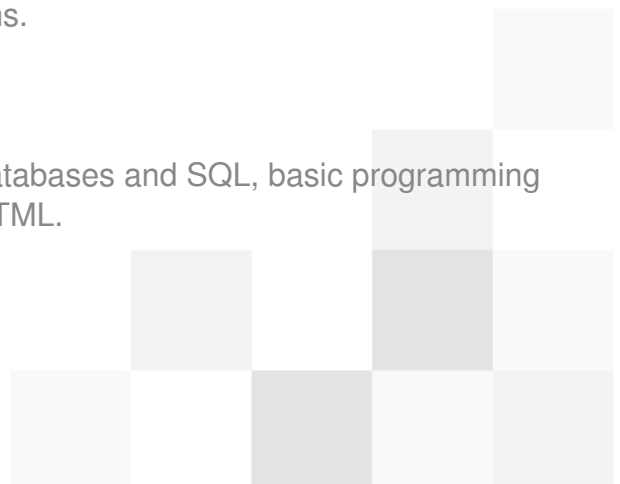
Course strengths:

The training program covers issues related to a web application development. In contrast to other trainings, this course is not focused on a single technology, but it covers most important issues of selected technologies. As a result, after completing the training, participants will be able to implement complete web applications by themselves. Training program is constantly updated due to a rapid development of web solutions.

Requirements:

The participants are required basic knowledge of databases and SQL, basic programming skills in Java, basic knowledge of JavaScript and HTML.

Course parameters:



5*7 hours of lectures and workshops at a ratio of 1:3. During workshops, in addition to simple exercises, participants will design and implement web application using JEE7, JS, HTML.
Group size: max. 8-10 people.

Course curriculum:

1. Introduction

- I. Java Enterprise Edition
- II. JEE most commonly used components: EJB, CDI, JPA, web services, websockets, servlets.
- III. Design and implementation with JEE
- IV. AngularJS as a MVC framework, thin versus thick client

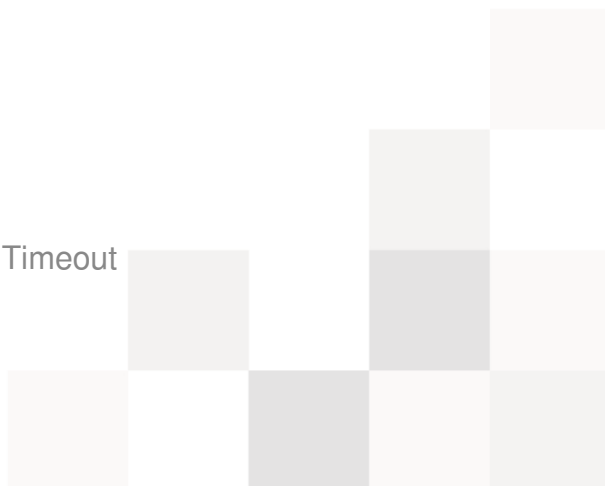
2. Managed Beans specification

3. Business logic layer - EJB 3.2

- I. EJB 3.x component types
- II. How does it work?
- III. Session components
 - i. Stateless
 - ii. Statefull
 - iii. Singleton
 - iv. Lifecycle, lifecycle callbacks
 - v. Accessing EJB
 - A. Locally
 - B. Locally without an interface
 - C. Remotely



- D. Web service
 - vi. Asynchronous EJBs
 - vii. Concurrency
 - A. Container role
 - B. SFSB and concurrency
- IV. Working with resources
 - i. JNDI lookups
 - ii. Inversion of Control
- V. Interceptors
- VI. Exceptions
- VII. Message Driven Beans
 - i. JMS2.0
 - ii. How does it work?
 - iii. MEPs
- VIII. Security
- IX. Transactions
 - i. Container managed transactions (CMT)
 - ii. Bean managed transactions (BMT)
 - iii. Client managed transactions
 - iv. JTA a JTS, CORBA
- X. TimerService
 - i. Timer and TimerService
 - ii. @Schedule, @Schedules, @Timeout



XI. EJB and web services

- i. Web services basics
- ii. JAX-WS/JAX-RS and EJB/CDI

XII. Deployment

- i. EJB module (jar)
- ii. Enterprise Archive (ear)
- iii. Web Archive (war)
- iv. Full vs light (web) server profile
- v. EJB embedded container

XIII. Testing EJB

XIV. EJB Lite and web applications

4. Persistence layer and Java Persistence API 2.1

I. ORM conception

II. Entities

- i. simple
- ii. embedded
- iii. accessing fields
- iv. collections

III. Associations

IV. Keys

- i. Simple
- ii. Complex
- iii. Automatic generation of keys



V. PersistentContext

- i. Container managed and application managed
- ii. Transaction and Extended

VI. Entity lifecycle

VII. DML

VIII. Transactions

IX. Queries

i. JPQL

- A. Syntax
- B. Query and TypedQuery
- C. Named queries
- D. Native queries

ii. JPA2.1 Criteria API

- A. Criteria API

iii. serach-within and serach-by-example

iv. Native queries mapping

X. Lifecycle listeners

XI. Testing JPA

XII. Deployment

- i. Java SE/dektop
- ii. Web module
- iii. EJB module

XIII. Validation



- i. Bean Validation 1.1
 - A. Introduction to Bean Validation
 - B. Validating fields, methods
 - C. Parametrizing validations
 - D. Built-in validations
 - E. Custom validations
 - F. Validation process
 - a. Fields, objects, graphs
 - b. Groups and sequences
 - c. Blocking field access

XIV. JPA2.1 and Bean Validation

- i. Configuration
- ii. Validations and database restrictions

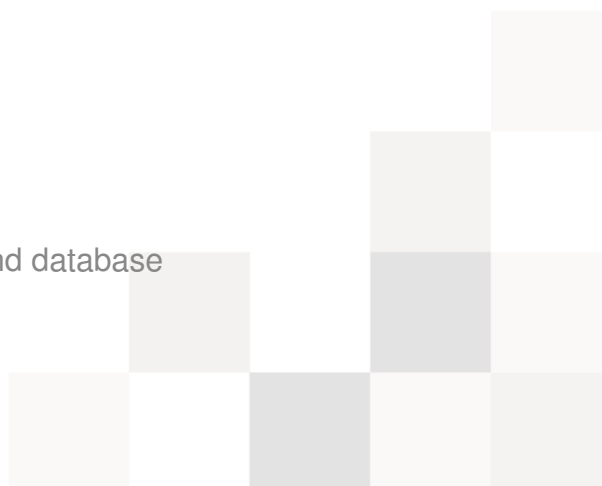
XV. Inheritance

XVI. Locking

- i. optimistic locking
- ii. pessimistic locking
- iii. Extensions in JPA2.1

XVII. 2nd level cache

- i. The idea
- ii. Configuration
- iii. Entities
- iv. Interaction between cache and database



v. Hibernate Cache (option)

vi. JPA2.1 Cache

vii. Issues

XVIII. JPA providers (Hibernate, Toplink, EclipseLink, OpenJPA)

XIX. Hibernate in details (option)

i. Hibernate API

ii. Features

iii. Hibernate: Search, Shards, Validator, Envers

XX. Design patterns

5. Java Servlet 3.1

I. Introduction

II. Lifecycle of a servlet

III. Sharing data

IV. Servlet initialization

V. Process HTTP requests

VI. Filtering

VII. File upload, media types

VIII. WebContext

IX. Asynchronous processing

X. Nonblocking I/O

XI. HTTP 1.1

XII. Security

6. GUI



- I. Introduction to JSF
 - i. WebApp typical problems
 - ii. Component-event model
- II. Request cycle
 - i. Request types
 - ii. Phases
 - iii. Full and partial processing
- III. Component model
- IV. Classes and interfaces
- V. Scopes
 - i. What is JSF scope?
 - ii. Configuration
 - iii. Custom scopes
- VI. Expression Language
- VII. Views
 - i. JSF tags
 - A. HTML
 - B. Core
 - ii. View Declaration Language
 - iii. JSF facelets
 - A. Benefits
 - B. Templating
 - C. Tags



D. Composite Components

- iv. JSF and JSP
- v. JSF and REST
- vi. JSF and HTML5

VIII. Conversation models

- i. Built-in converters
- ii. Custom converters

IX. Validation

- i. Built-in validators
- ii. Custom validators
- iii. Default validator
- iv. Bean Validation 1.1 and JSF
- v. Validation events
- vi. Complex validation scenarios

X. Events

- i. Events and listeners
- ii. Action events
- iii. Phase events
- iv. System events

XI. Navigation

- i. Classic
- ii. Short
- iii. Conditional



- iv. Dynamic
 - v. Pull and push MVC

XII. Internationalization

XIII. Web interface issues

- i. Wizards
- ii. Pagination
- iii. Master-Detail, LOV

XIV. Error handling

XV. Resources

- i. Resource Request
- ii. EL

XVI. Behavior

XVII. Customizations

- i. Custom component
- ii. Custom renderer
- iii. Custom tag

XVIII. Deployment model

XIX. JSF testing with JSFUnit/Selenium

XX. Security

7. Web application using AJAX and JSF

I. AJAX communication models

II. JSON

- i. Introduction



ii. Java API for JSON Processing

iii. JSON and Restful web service

III. Prototype and JQuery libraries

IV. JSF2

i. f:ajax and Ajax Behavior

ii. Partial processing

iii. DOM based operations

iv. Queueing

v. Error handling

V. JSF libraries

i. PrimeFaces

ii. OmniFaces

iii. RichFaces

iv. PrettyFaces

v. IceFaces

8. CDI (WebBeans)

I. Introduction

i. Bean

ii. Context

iii. Scope

iv. Injection

II. Beans

i. Managed beans



- ii. Session means
- iii. Specialization
- iv. Active and inactive beans

III. Scopes

- i. Built-in scopes
- ii. Conversation scope
- iii. Dependent scope
- iv. Nonstandard scopes
- v. Default scope
- vi. Lifecycle of scope objects

IV. Injection

- i. Where to inject
- ii. Qualifiers, Alternatives, Stereotypes
- iii. Named injections

V. Producers

VI. Events and listeners

VII. CDI and EJB

