# Vaadin 8.17 with Spring

#### Environment/Lab set up:-

1. Java 8
2. Eclipse Neon 3
3. Web logic 12.2 or above
4. Oracle 11g xe
5. Spring 4 and Spring Boot

Specifics: Please provide install instructions to the participants so that only issue of the installation can be resolved during the course.

# Course Outline

#### Introduction and goals

Course goals and overview: the course is to provide a hands on experience on using Vaadin 8.17 version and use it along with spring /spring boot.

At the end of the course participants should be able to construct Rich internet application using Vaadin 8.17 and Spring frame work.

# Day 1

# Topic -1: Use of Maven

What is Maven?

Maven with command line and Eclipse

Project Object Model (POM)

Build lifecycles and phases I – clean  
Build lifecycles and phases II – site  
Build lifecycles and phases III – default

Plugins

Plugins example

Maven repositories

Multiple modules

Parent pom vs aggregator pom

[Using WARs](javascript:void(0))

# Topic 2 :-Eclipse Link

**Goal:** Educate EclipseLink users on basic JPA usage.

**What is Eclipse Link**

JPA Terminology

Anatomy of a JPA Application

**Configuration**

persistence.xml, annotations, and orm.xml

Data sources and native connection pooling

**Mapping**

Direct Identity: Id, IdClass

Basic

Entity Relationships: OneToOne, ManyToOne, ManyToMany

Embeddable & EmbeddableId

ElementCollection/Map

MappedById (JPA 2.0)

Sequencing

Version Locking

**Metamodel (JPA 2.0)**

**Querying**

Java Persistence Query Language (JPQL)

NamedQuery

NativeQuery

CriteriaQuery

**Transactions: Persistence Context**

EntityManager & extended Persistence Context

JTA and EntityTransaction

Entity Lifecycle: Managed, Detached, New

Cascade MERGE, PERSIST, REMOVE, ALL

Concurrency protection - Optimistic and Pessimistic Locking

# Day2

# Spring Framework - XML Configuration

Architecture of Spring framework

First Spring application

Spring containers

Bean scopes

Bean init and destroy

Dependency injection

Annotations I

Annotations II - java based annotations autowire qualifier configuration

@Qualifier annotation

**Spring Boot**

Why Spring Boot came to be?

First spring boot application

Using annotations in Spring Boot

Building with Maven

Building with Eclipse

Spring boot with vaadin integration

Spring and Spring Boot summary

# Day 3

# Vaadin Framework Basics

During this course we can build these two screens after covering the basics or build this screen as part of covering the basics. **Please expand the image for a better view.**

Further how to use the screens designed in Vaadin designer needs to be covered in this session.

|  |  |
| --- | --- |
| cid:image001.png@01D367C0.F3DD3A60 | cid:image002.png@01D367C0.F3DD3A60 |

Vaadin architecture

Label

TextField - basics

TextField - events

TextArea

Buttons

CheckBox

OptionGroup

ComboBox

Grid

Grid selections

Tree

TabSheet

Layouts introduction

Layouts - HorizontalLayout

Layouts - VerticalLayout

Layouts - AbsoluteLayout

Layouts - FormLayout

Layouts - GridLayout

Panel

Data binding

Validators

Navigation

Events and Listeners

Vaadin with spring summary

# Day 4

# Wiring All Together - Web Application (Configuration)

Configuring the application with Maven I - multi modules

Configuring the application with Maven II - version

Configuring the application - spring boot I

Configuring the application - spring boot II

Integrating with Vaadin

Building with Maven

Configuring the application - deploy to Weblogic

Creating the main layout

Logo layout

Menu layout

Enum constants

# Custom Navigation:

<Shivaram> Please outline what will be covered here as part four parts. That will help us refine the content.

<Shivaram> Please ensure that both the types of Navigation is covered, that is event based Navigation (kindly look at the enclosed example) and spring based Navigation is also covered.

Custom navigator I

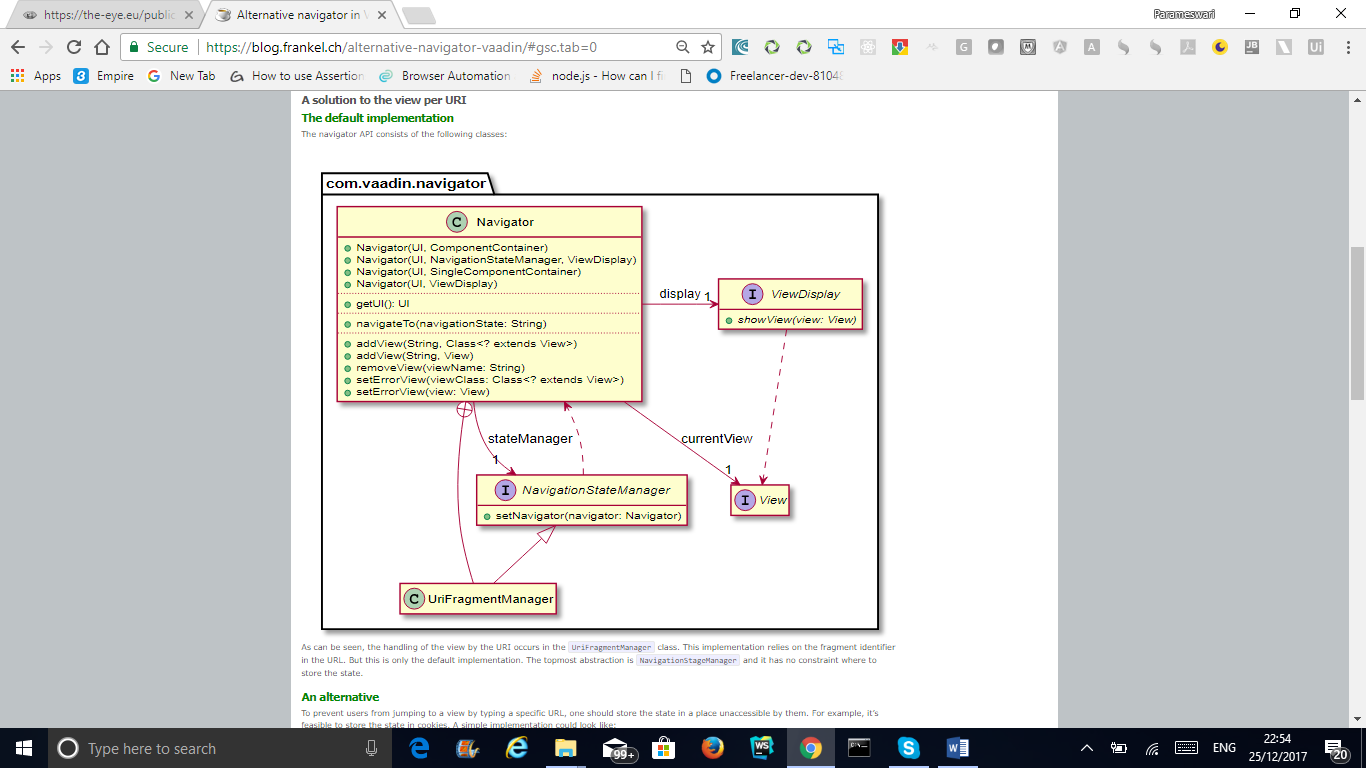
Navigation

Custom navigator II

Navigation

Custom navigator III

Navigation



Custom navigator IV

Spring View

|  |
| --- |
|  |
| @SpringUI |
|  | @Theme(ValoTheme.THEME\_NAME) |
|  | public class NavigationUI extends UI { |
|  |  |
|  | private final SpringViewProvider viewProvider; |
|  |  |
|  | @Autowired |
|  | public NavigationUI(SpringViewProvider viewProvider) { |
|  | this.viewProvider = viewProvider; |
|  | } |
|  |  |

<Shivaram> For the case study please take up a inventory management application. I have enclosed a copy of code which is developed; please ensure to include this in the case study.This will be a part of our discussion. We have the license for the Bakery application which Vaadin provides as a demo.

# Transactions

Transaction introduction

Using @Transactional

# Day 5

# Spring Security

# Authentication VS authorization

<Shivaram> The authentication should include a screen based authentication and when request is from some other application and login credentials are sentin header as part of HTTP request.

------------------ AUTHENTICATION -----------------------------

[Overview of authentication](javascript:void(0))

Spring security configuration

Spring security configuration II

Setting up the database: USER

Repository and service update

Login form implementation I

Login form implementation II

Login form implementation III - authentication

Signup layout

Signup service and repository, Wiring it together