

Course code: **SPRING/WKT**

Course title: **Modern Web applications based on Spring, Hibernate, Apache Wicket and Jetty**

Days: 5

## Description:

### Course intended for:

The training is intended for Java programmers wanting to find out how to develop Web applications on the basis of the container model of the user interface, supported with the business logic layer, implemented by Spring Framework using access to a database through the ORM (Hibernate) mechanism.

### Course objective:

The training objective is to get familiar with the principles of building of Web applications, using such issues as: object-relational mapping and development of a layer of access to data using Hibernate, development of the service layer using a Spring IoC container, the user interface software thanks to the Wicket component framework.

In particular, the participants are to:

- Get familiar with the Dependency Injection pattern and learn to use it in practice with support of the IoC Spring Framework container
- Find out how to configure components from the XML Spring Framework container level and using Java annotations
- Get familiar with the rules of object-relational mapping and learn to design the persistency layer on the basis of Hibernate, starting from simple entities to complex issues, such as inheritance
- Learn how to read and save entities, ensure operation transactionability
- Learn how to care about cohesiveness of data and performance of the application
- Learn to build a user interface using Wicket
- learn to use the ready Wicket library components and create their own

- find out how to make the user interface more comfortable by using AJAX
- get familiar with the issues of security and reliability of Web-based applications
- learn how to launch the applications developed on Jetty servers

## Course strengths:

The curriculum includes a comprehensive and detailed overview of Web application development. Unlike most trainings, it is not focused on a single technology, but it encompasses the key issues from the selected technologies. Thanks to this, the training participants, after the training, will be able to develop full Web applications on their own, developing their knowledge further individually as needed. The curriculum is being constantly updated to take into account the emerging trends.

## Requirements:

The training participants are required to have the Java programming skill (to be learned at the course J/JP), to have the basic knowledge of relational databases and SQL, and the basic knowledge of HTML. They should also be able to develop Web applications using Servlet/JSP technologies (to be learned at the course J/WEB1).

## Course parameters:

5\*8 hours (5\*7 net hours) of lectures and workshops (with a visible emphasis on workshops). During the workshops, apart from simple exercises, a simple complex application is designed and implemented.

Group size: no more than 8-10 participants.

## Course curriculum:

### 1. Introduction – a review of issues to be discussed

- Layer architecture of IT systems
- Patterns: IoC, DI, MVC
- Data layer: JDBC, DAO, JPA
- Web layer: HTML, AJAX
- System integration issues, Webservices



- Aspect oriented programming: AOP
- Issues associated with testing and security

## 2. Business logic layer – Spring IoC

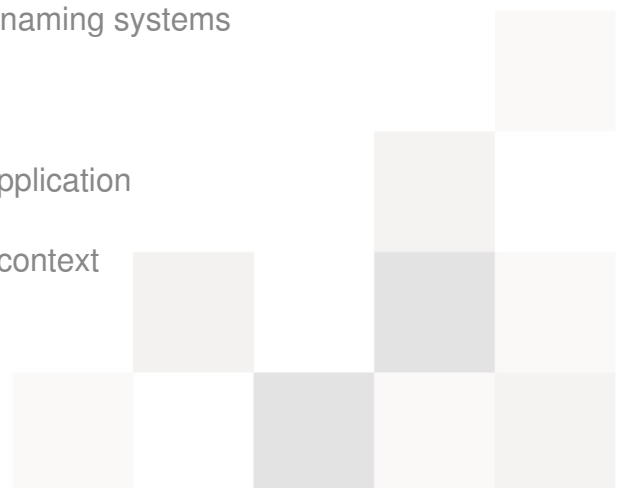
- Inversion of Control and Dependency Injection patterns
- Application structure in Spring Framework
- Container launching
- Defining of components
  - Constructors, factories
  - Simple properties
  - Complex properties, collections
  - Combining of collections
  - inheritance
  - p-namespace
- specification of correlations
- lazy initialization
- Autowiring
- Resources
- Cooperation with container
- Range of components
  - singleton, prototype, request, session
  - own ranges
- Injection of methods
- Bean life cycle and listening



- Closing of container
- Separation of a part of the configuration to property files
- location

### 3. Data layer – Spring ORM and Hibernate

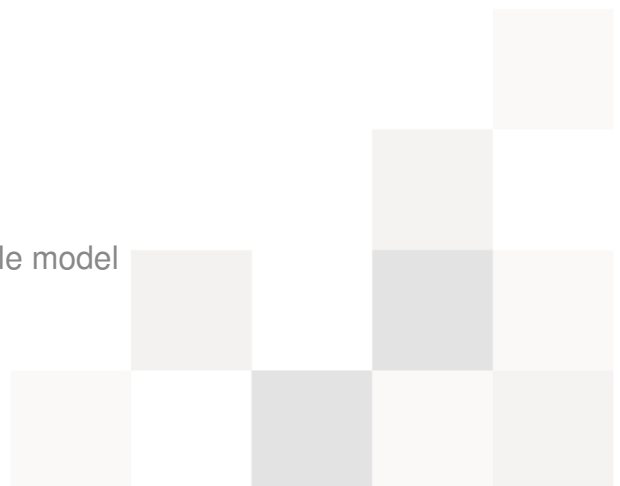
- The object-relational mapping concept
- Entity classes
  - Simple classes
  - Embedded classes
  - Fields and access to fields
- Associations
  - Unidirectional
  - Bidirectional
  - Cascade propagation
- Keys
  - Simple
  - Complex
  - Automatic key generation
- Object composition
- Inheritance
- Naming of objects in database, own naming systems
- Persistency context:
  - Managed by container and application
  - Configuration of persistency context



- Integration with Spring Framework
- Configuration DAO
- Entity life cycle
- Operations on data
- Transactions
  - Transaction isolation levels
  - Transaction subsystem in Spring
  - Declarative transaction specification
  - Pessimistic and optimistic blocking
  - Long conversations
- HQL queries
- API criteria
- Entity life cycle listening
- Threats

#### 4. Web layer – Wicket

- Basic application architecture
  - Configuration
  - Request processing
  - Java code role
  - Html role
- Data model concept
  - Standard model
  - Resource control – detachable model

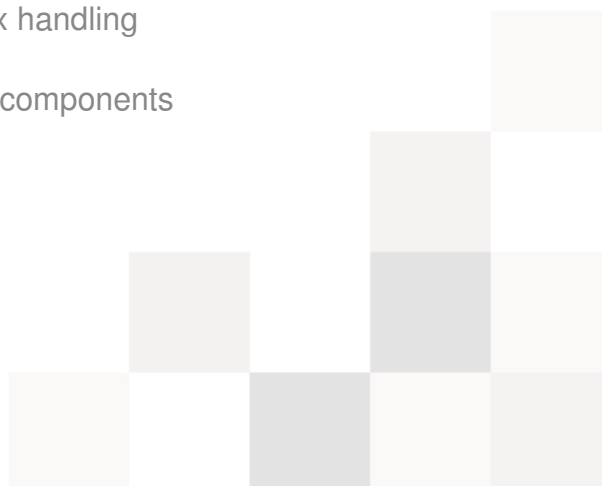


- Embedding of models
- UI basic components
  - Text display
  - Link management
  - Repeaters
  - Component attribute modification
- Handling of forms
  - Form structure
  - Form components
  - Submit management
  - Data validation
  - feedback
- Grouping of components
  - Inheriting of templates
  - Panel
  - Border
  - Fragment

## 5. Ajax in Wicket

- Ajax handling in Wicket
  - Components with in-built Ajax handling
  - Adding Ajax handling to own components
  - Form handling using Ajax

## 6. Development of own components



- Benefits of development of own components
  - Working with resources (css, scripts)
  - Broadening of component capabilities
  - Own form components
  - Distribution of components

