

Course code: **DIST**

Course title: **Ensuring high availability of Web applications on the basis of Spring, Terracotta, HA-JDBC and open database systems**

Days: 2

Description:

Course intended for:

The training is intended for designers and programmers of Web applications, who would like to develop their skills by mastering building of high availability and scalability applications.

Course objective:

The trainers will discuss in detail various architectures of distributed systems. The participants will get familiar with open technologies, allowing for building of distributed applications resistant to failures. They will learn how to design the application architecture to make sure that the capability can be easily increased by adding subsequent servers. Strong emphasis will be put on the aspects of scalability of databases.

Course strengths:

After the training, the participants will be ready to design high-availability Web applications and modify the existing applications to make them scalable and failure resistant.

Requirements:

The participants are expected to have the Java Web application development skills (to be learned at the course J/WEB1 and J/WEB2), to be familiar with Spring-Framework and JPA or Hibernate.

Course parameters:

2*8 hours (2*7 net hours) of lectures and workshops (with a visible emphasis on workshops).

Group size: no more than 8-10 participants.

Course curriculum:



1. Introduction – various cluster architectures and their impact on reliability and scalability; the CAP theorem
2. Ensuring reliability and scalability of the data layer:
 - I. Hot-standby and stream replication – the example of PostgreSQL
 - II. Multi-master replication – the example of HA-JDBC + PostgreSQL
 - III. Master-slave replication – the example of Slony-I + PostgreSQL
 - IV. Introduction to Apache Cassandra
3. Data buffering
 - I. Configuration of distributed cache (memcached or ehcache)
 - II. Buffering and tuning of GC in JVM, data storage outside stack
 - III. Buffering in databases (the example of PostgreSQL and/or Cassandra)
4. Scalability and reliability of WEB layer:
 - I. Load-balancing – the example of HA-Proxy
 - II. Session replication and load-balancing of HTTP in Apache Tomcat

