sages

Course code: **J/REST**

Course title: **RESTful web services in Java**

Days: 2

Description:

Course intended for:

The training is aimed at Java programmers, system analysts, and architects who want to create advanced RESTful web services.

Course objective:

Aim of the training is to familiarize its participants with advanced technologies of RESTful web services. Trainees will learn details of http protocol, techniques of RESTful web services and its clients implementation, methods and standards for describing, securing, increasing efficiency of RESTful services. Trainees learn details of those standards as JAXB 2.0, Servlet 3.0, JAX-RS 1.1. Practical part of the training will be carried out on a pile of JBoss AS and JBoss Resteasy, but on request can be executed on another compatible stack e.g. .: Apache Tomcat and Apache CXF. The training is carried out in several variants - for programmers, system analysts, architects. Depending on the training group, the main emphasis is on theoretical issues or practical implementation and integration solutions.

Requirements:

Participants are required at least basic programming skills in Java, basic knowledge of web services and XML processing technologies. Basic knowledge of JEE is also recommended.

Course parameters:

2*7 hours of lectures and workshops at a ratio of 1/3. During the workshops examples presenting advanced capabilities and flexibility of web services are implemented. Group size: max. 8-10 people.

Course curriculum:

1. Introduction
    I. key concepts and standards of web services: XML, DTD, XSD, XSLT, SOAP,

WSDL, WADL, JAX-RPC, JAX-WS, JAX-RS, JAXP, SAX, StAX, DOM, JAXB, JAXR, UDDI, SAAJ, JSON, AJAX, REST itd.

II. what is REST?

III. RESTful versus SOAP web services

2. Transport layer

I. detailed discussion of HTTP

II. the implementation of session in web applications

III. web services and transport protocol: SOAP over HTTP, JMS, SMTP / POP3, TCP, UDP; XML / JSON / XHTML browser over HTTP; other e.g.: WebSocket, MSMQ

3. Metadata, configurability and flexibility

I. WSDL 2.0, WADL

II. JAX-R, UDDI, REST registries and repositories

4. Security

I. authentication and authorization at http, configuration of servlet security, JAAS, OAuth

II. standards: PKI and X.509, SSL, HTTP headers responsible for security, XML Digital Signature and so on.

5. Reliability

I. REST reliability

6. Performance and efficient processing of XML and JSON

I. methods used to increase processing efficiency at the level of the transport layer protocol, application layer protocol

II. standards: MTOM, XOP, MIME, SMIME, JAXP, SAX, StAX, DOM, JAXB, XPath, XSLT, etc.

III. JSR 222 or Java Api for XML Binding (JAXB) 2.0 in details

i. serialization, externalization, marshalling
ii. JAXB architecture
iii. mapping Java classes to XML Schema
iv. marshalling, unmarshalling
v. validation
vi. modification of the standard mappings using annotations

7. Implementing REST services based on Servlets
   I. JSR 315, Servlet 3.0 in details
      i. annotations and deployment descriptor
      ii. safety
      iii. asynchrony
      iv. other
   II. web application filters
   III. web application event listeners
8. JAX-RS as a implementation of REST
   I. JSR 311, Java API for RESTful Web Services (JAX-RS) 1.1 in details
      i. annotations and deployment descriptor, basic concepts
      ii. working with parameters (@ * Param)
      iii. working with @Produces and @Consumes
      iv. working with the context and @Context
      v. other
   II. JAXB and JAX-RS
      i. MIME types and MIME providers providers

ii. marshalling and @Provider

III. Others JAX-RS implementations

i. Apache CXF

ii. JBoss RESTEasy

iii. Apache Wink

iv. others, np.: commercial JEE6 standards

IV. JBoss RESTEasy in details

i. JAX-RS compliance

ii. additional features

A. error handling

B. compression

C. cache

D. interceptors

E. asynchronous processing

F. security

G. AJAX support

a. integration with other frameworks and application servers

V. JAX-RS 2.0

9. REST and popular java frameworks

I. Spring

II. Struts

III. Seam

sages

10. REST clients implementation
    I. Jersey Client API
    II. JAX-RS 2.0 Client API
    III. Spring RestTemplate
    IV. AJAX
    V. generic http client