



S Y L L A B U S

Course name:	Java™¹ Portlet Development with JSR 286: Portlet 2.0 API & Liferay®² – JSP™, Spring MVC, JSF & AJAX Portlets
Course code:	IJ-07
Course domain:	Software Engineering
Number of modules:	1
Duration of each module:	40 study³ (30 astr.) hours
Overall duration of the course:	40 study (30 astr.) hours

Sofia, 2013

Copyright © 2003-2013 IPT – Intellectual Products & Technologies Ltd. All rights reserved.

1 Oracle®, Java™, JSP™, and Servlet™ are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

2 Liferay® is a registered trademark of Liferay, Inc.

3 Duration of a study hour is 45 minutes.



Java™ Portlet Development with JSR 286: Portlet 2.0 API & Liferay® – JSP™, Spring MVC, JSF & AJAX Portlets

(1 module)

STUDY PLAN

Module name	1. Java™ Portlet Development with JSR 286: Portlet 2.0 API & Liferay® – JSP™, Spring MVC, JSF & AJAX Portlets
Lectures (study hours)	20
Laboratory exercises (study hours)	20
Total (study hours)	40
Form of the exam	Test
Certificate	IPT

Lecturer:

Trayan Iliev

IPT – Intellectual Products & Technologies Ltd.

E-mail: tiliev@iproduct.org

Course Description:

In order to cope with the challenges of contemporary business environment including globalization, economical pressure for better efficiency, business process outsourcing, achievement of regulatory compliance, the enterprise needs improved support by technology. Recently the term Service Oriented Architecture (SOA) has become a pervasive buzz word together with *Cloud Computing*, *Software as a Service (SaaS)*, etc. The SOA standards like: *Simple Object Access Protocol (SOAP)*, *Web Services Description Language (WSDL)*, *Representational STate Transfer (REST)*, *Web Application Description Language (WADL)* have emerged and matured. But from the end user perspective all these technologies do not matter much.

In 2002 the Java™ community started a pioneering effort for standardization of the web-based presentation services as well. They were called portlets and were able to be easily combined and integrated into enterprise portals based on a common standard defining their interaction – [JSR 168: Java™ Portlet Specification](#). In 2005 it was followed by another more advanced specification – [JSR 286: Portlet Specification 2.0](#).



Portals offer many advantages over other software applications. First, they provide a single access point for all employees, partners, and customers. Second, portals provide access to business functionality transparently from any device in virtually any location. Third, portals are highly flexible; they can exist in the form of B2E intra-nets, B2B extra-nets, or B2C inter-nets. Fourth, portals can be combined to form a portal network that can span a company's ecosystem. Fifth, because they provide front end for different web services they can easily integrate existing heterogeneous software systems and are future proof.

The *Java™ Portlet technology (JSR 286: Portlet 2.0)* allows easy sharing and combination of web applications developed by different organizations and individuals in a personalized enterprise portals. New "portlet-based" web application development style has emerged. A brief description for portlet in SOA (Service Oriented Architecture) style is: "*presentation-oriented web service*". *Portlet apps (PA)* are more distributed, flexible and agile, compared to older style, monolithic web applications we know for years. *PA* can use asynchronous data requests and can be dynamically updated in response (and sometimes in anticipation) to user's needs. They are typically consisting of several different portlets that communicate using shared parameters or publish/subscribe events according to latest *Portlet 2.0 Specification (JSR-286)*. [Liferay®](#) is a state-of-the-art open-source Enterprise Portal Server licensed under LGPL, which was integrated by Sun in their portal server.

The IPT course *Java™ Portlet Development with JSR 286: Portlet 2.0 API & Liferay® – JSP™, Spring MVC, JSF & AJAX Portlets* introduces to practical development of individual portlets, themes and layout templates, their integration in portlet applications, using the specific services provided by the portlet container, such as users' personal data management, web content management, etc., as well as use of tools and templates for efficient development of portlets. Among the unique opportunities provided by this course is the possibility to share the experience gained by IPT in building new lightweight "JavaScript portlets" which execute entirely on the client side, and their integration with the portal container provided services.

In order portlet technology to become even more successful are needed developer frameworks and tools that support rapid portlet application development and reuse of components. As part of Java™ Enterprise Edition standardization process and as a recommended technology for web based presentation in the latest version of Java™ EE 6, Java™ Server Faces (JSF) framework became most promising candidate for such an enabling technology. Among the advantages of JSF are:

- ✓ easy construction of UI from a set of reusable components;



- ✓ clear separation between data and presentation using MVC design pattern;
- ✓ easy to use model for wiring client-generated events to server-side application code;
- ✓ UI components state managed automatically across client requests;
- ✓ separation of concerns between corporate developers and system programmers.

The latest version of JSF 2 provides additional advantages:

- ✓ more flexible and standard based presentation components using facelets;
- ✓ easy to use view/page templating;
- ✓ easy to create custom components without Java™ programming by composing existing components;
- ✓ seamless and unified integration with all different types of beans (ManagedBeans, POJO, EJB) using dependency injection annotations;
- ✓ new scopes (e.g. conversation scope, custom scopes);
- ✓ bookmarkable application states using view parameters;
- ✓ better *AJAX* support using standard tags – no need to manually write JavaScript code;
- ✓ partial view processing and rendering during *AJAX* requests;
- ✓ easier configuration, navigation and resource loading.

JSF technology provides many nice capabilities simplifying the development process, but there are some problems using JSF directly for portlet development. The most important one is the difference in lifecycles of portlets and JSF components – portlets separate action, event, resource and render requests, while standard JSF servlet™ handles them in a common request processing lifecycle. That is why a bridge between two technologies is needed in order to combine their advantages. Two new specification requests (targeted towards different versions of the portlet specification) were published to address this need – [JSR 301: Portlet 1.0 Bridge for JavaServer™ Faces 1.2](#), and [JSR 329: Portlet 2.0 Bridge for JavaServer™ Faces 1.2 Specification](#).



The course will present the **details of Java™ portlets development** using different web presentation technologies:

- [JavaServer Pages 2.2/Expression Language 2.2](#), [Standard Tag Library for JavaServer Pages \(JSTL\) 1.2](#), including [Asynchronous JavaScript + XML \(AJAX\)](#) portlet development;
- [Spring MVC](#) framework;
- [JSF 1.2](#) technology (including [JSR 301](#) and [JSR 329](#) open source implementations).

The course will also provide insight about emerging initiatives for development of [Portlet 2.0](#) to [JSF 2](#) bridge (no specification available yet) and practical development of portlets accessed through [Web Services for Remote Portlets \(WSRP\)](#) [OASIS](#) network protocol specification allows the portlets to be accessed remotely and combined easily by third party portals and consumers, supported by all of the portal market's major players, including Oracle®, IBM®, Microsoft®. The ultimate goal of WSRP is to bring the benefits of Service-Oriented Architecture to the end-user.

Practical examples using open source technologies will be demonstrated and developed by participants illustrating the concepts, and proving the value of the technologies. Participants will develop solutions to multiple problems and home work tasks with opportunity for individual consultation with the trainer. The implemented applications will be deployed on [Liferay® 6](#) state-of-the-art open-source portal server.

The course is led by Trayan Iliev – qualified trainer and university lecturer with 12-years pedagogical experience in Faculty of Mathematics and Informatics of Sofia University “ST. Kliment Ohridski” and IPT – Intellectual Products and Technologies. He has practical experience in technical development and management of multiple software projects (eGovernment, eLearning, WebTV, Web Based GPS Tracking, Web 2.0 Mashups, JavaScript Portlets).

Ideally, the course participants should have previous knowledge and practical experience in Java™ Web programming technologies (JSP™, Spring MVC, JSF, AJAX) that will be discussed during the course.



COURSE PROGRAM

A. Lectures.

Module 1: Java™ Portlet Development with JSR 286: Portlet 2.0 API & Liferay® – JSP™, JSF & AJAX Portlets

1. Enterprise web portals – advantages, features, architecture, components and services, examples. Open source portal development platforms – Liferay, JBoss GatIn, ExoPlatform, etc. Portlet Applications (PA). Portal installation – Liferay. Adding portal pages and portlets. Choosing page *layout* and *theme*. Deployment of custom portlets and themes. User and user group management. Content and document management. Liferay® portal configuration and administration specifics. (2 h.)
2. Overview of portlet specifications – *JSR 168: Java™ Portlet Specification*, and *JSR 286: Portlet Specification 2.0*. *JSR 286: Portlet 2.0 API* and practical portlet development. Portlet containers and portlets lifecycle. Different types of portlet requests – action request, event request, render request, resource request. Detailed review of *Portlet 2.0 API*: *init()*, *destroy()*, *processAction()*, *render()*, *doView()*, *doEdit()*, *doHelp()* etc. Portlet modes and portlet window states: MINIMIZED, NORMAL, MAXIMIZED. Portlet *deployment descriptor*. Portlet 2.0 Tag Library – main tags: *defineObjects*, *actionURL*, *renderURL*, *resourceURL*, *namespace*, *param*, *property*. Changing the default behavior using *escapeXml* tag attribute. *JSR 286: Portlet 2.0 API* advanced capabilities for inter-portlet interaction – *shared parameters* or *publish/subscribe events* and can dynamic queries for server resources. Examples of inter-portlet communication. (6 h.)
3. Introduction to *Asynchronous JavaScript + XML (AJAX)* technology. AJAX Portlets – asynchronous request processing in *Portlet 2.0 API* using resource requests. *Client-side* inter-portlet communication and interaction using JavaScript. Introduction to jQuery and using it for dynamic inter-portlet interaction in Liferay. Examples. Specifics of Liferay® portlet development – deployment descriptors, additional services provided by the portal not-defined in the specification and their use – access to current user data, etc. Liferay® custom tag libraries. Developing custom themes for Liferay® server. Single Sign On (SSO). Liferay® Portal and portlet security. (5 h.)
4. Using alternative view technologies for portlet development: Java™ Server Pages JSP, Spring MVC, Java™ Server Faces (JSF). Development of JSF 1.2 and 2.0 Portlets. Introduction to JSF. JSR 314: JavaServer Faces 2 new features. Facelets. Templating. Ajax support. Differences in portlet and servlet lifecycles. *JSR 301: Portlet 1.0 Bridge for JavaServer™ Faces 1.2* and *JSR 329: Portlet 2.0 Bridge for JavaServer™ Faces 1.2* specifications. Practical development of JSF portlets. Examples. Emerging initiatives for development of Portlet 2.0 to JSF 2 bridge. Portlet 2.0 to JSF 2 bridge implementation for Liferay®. Development of JSF 2 AJAX portlets. Examples. Web Services for Remote Portlets (WSRP) OASIS



network protocol specification. Development of WSRP portlets – detailed explanations. Examples . (6 h.)

5. Final test. (1 ч.)

B. Hands-on labs

The portlet development hands-on labs for this course use problem-oriented training methodology. The exploration of examples, and development of own Java™ portlet applications are done using latest versions of *Eclipse* (<http://www.eclipse.org/>) and/or NetBeans (<http://www.netbeans.org/>) IDEs, in combination with *Liferay® 6* state-of-the-art open-source portal server. They provide all necessary features to support the design, implementation, deployment, and testing of portlet applications.

The training is conducted in small groups – up to 6 participants. Towards each of them is applied individualized pedagogical approach. During the lessons a time is reserved for individual consultation, and discussion of additional questions interesting for participants.

RECOMMENDED LITERATURE AND INTERNET RESOURCES

1. Sarang, P., Practical Liferay® – Java™-based Portal Applications Development, Apress, 2009
2. Minter, D., Linwood, J., Building Portals with the Java™ Portlet API (Expert's Voice) (v. 2), Apress, 2004
3. Liferay® Enterprise Portal Server official webpage – <http://www.liferay.com/>
4. GateIn Portal Server official webpage – <http://www.jboss.org/gatein>
5. JSR 286: Portlet Specification 2.0 official webpage – <http://www.jcp.org/en/jsr/detail?id=286>
6. JSR 329: Portlet 2.0 Bridge for JavaServer™ Faces 1.2 – <http://jcp.org/en/jsr/detail?id=329>
7. Web Services for Remote Portlets (WSRP) OASIS official webpage – http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
8. Garrett. J., Ajax: A New Approach to Web Applications – <http://www.adaptivepath.com/publications/essays/archives/000385.php>