Java EE Fundamentals

Course notation

Introduction

The principles of developing web applications didn't change much since the dawn of providing dynamic content in Internet. There is usually a central (sometimes distributed) storage of user and application data. Then there is some business logic that works with that data. And finally there is the application view, which is used by the end user to communicate with the data.

These three tiers are not always set like that. In some solutions the business logic is developed directly in the database. While in other simpler ones the view part manipulates directly the data without going through an intermediate layer. The views themselves walked a long path from ugly representations of the underlying storage to appealing pieces of art that look much like desktop applications.

But the principles remain the same: there is data that should be processed and presented to the user. Throughout the years there were many approaches on how to put those principles to work. But quite a few of those approaches stood out. One of them is Java Enterprise Edition (or Java EE). It is a composition of a number of standards neatly weaved together to represent one whole.

This course will take you though the most prominent of these standard technologies and show you how you can use them to easily build a working web application. It is going to focus on the latest version of Java EE - Java EE 7.

Course overview

Each part of the course will start with an introduction of a certain technology from the Java EE stack in the form of slides. But we at VIDA Software know that the second best way of learning something is via practicing it (the best is via teaching it ;)). We also believe that true quality is achieved with down to earth examples, rather than with yet another Hello World or foobar nonsense. That is why this course is accompanied by a sample real world web application. During each installment the attendees will work on a few new features of this app, that are mostly connected with the covered material.

In order to achieve best understanding of the technology, the first one or two tasks after each lecture will be explained in a step-by-step manner in the course handout. Thus the attendees will be able to follow along and move on with the initial assignments. For the rest of the tasks we will leave a few hints on how to be implemented, but the participants will have to do them by their own. At the end they will have the chance to submit their work and get feedback by the trainer in the following days.

Course structure

The course is broken down into three profiles: *minimal*, *sensible* and *extended*. Each one of these comprises of a number of installments and develops on top of the previous one.

Here is a brief description of the course structure

Profile	Topic	Hours	Description
Minimal	JPA	3	Java Persistence API is the way to map relational data from the DB to the object oriented representation of the Java programming language
	Servlets	2	Although not much used directly in the contemporary Java EE applications, Java Servlets are the backbone of the web tier of any app. That is why understanding how they work is essential for getting the technologies built on top.
	CDI	6	If JPA lets the OO application speak to the database and the Servlet bridges the HTTP calls, then CDI is the technology that glues together all the rest (including the first two). Understanding CDI is key to writing proper Java EE applications.
	JAX-RS	3	JAX-RS lets you easily write RESTful APIs and thus opening your business logic to the non-Java world. If you plan to support UIs like JavaScript or mobile, that’s the way for them to connect to your backend.
	JSON-P	2	That is the API which marshals Java objects into JSON format and vice versa.
Sensible	EJB	3	The pre-CDI way to build the business layer, Enterprise JavaBeans (EJB) still have a lot of great capabilities. There are situations when using EJBs is simpler and faster. That is why it is crucial to know how to write proper EJBs.
	Bean Validation	2	Validating user input is one of the key features of robust application. Being able to configure the validation rules ones at the data layer (i.e. JPA) and let them propagate through the whole stack is the main selling point of Bean Validaton.
	JSF	6	JavaServer Faces is still the one and only Model-view-controller technology in Java EE. Besides controlling the basic request response cycle of the application, it has many other great features like automatic input validators and convertors, internatialization, AJAX support and many more.

Profile	Topic	Hours	Description
Extended	JMS	3	Communicating asynchronously and reliably with external systems
	JBatch	2	An enterprise application does not just show screens to a user for them to click links and buttons. A lot of work is left for the time when those users are offline. Executing batches of jobs is the main purpose of the JBatch specification, part of Java EE 7.
	WebSockets	2	The request-response nature of the HTTP protocol has its constraints. Web application developers have tried to overcome those constraints by techniques like long polling. WebSockets are a completely different paradigm of server pushing data to the client.
	Java EE Security	2	Not that security is not important topic, but it is so hard to do it by the book in Java EE 7. The last installment of this course looks at a few patterns to do application security. Hopefully Java EE 8 ships soon with the longly anticipated Security JSR.

Course requirements

In order to participate in this workshop, you need to bring your laptop. It may run on any operating system (Windows, Mac, Linux). Just make sure that you have the following software installed:

- Java SE 8
- Maven 3.3+
- An IDE (even Eclipse is good)
- WildFly 10 application server
- Git client (all IDEs have)
- Adobe Acrobat Reader

The course assumes fairly good knowledge of Java SE. Only basic understanding of Maven and Git is expected.