PV243 Pokročilé Java technologie: JBoss

Část 1. - Úvod, Java EE 6

Jiří Pechanec

březen 2012

Agenda

- Introduction
- Course information
- Tools
- Java EE 6

Motivation

- Bring latest information about Java EE technologies
- Provide information about JBoss projects
- Pilot run
- Feedback NEEDED

Organization

- 7 lessons
  - Presentation followed by practical training
    - Source code available in public
- Evaluation based on a homework project developed using Java EE 6 technologies
- Project is expected to be deployed on run on OpenShift PaaS

Topics

- Introduction to Java EE 6
- CDI, EJB 3.1
- Seam 3 (Apache DeltaSpike)
- Datagrids, Infinispan
- Java EE security
- Clustering and scalability in JBoss AS 7
- Management and monitoring

Tools

- JBoss AS 7.1
- JBoss Developer Studio 5.0.0 M5
- Maven
- Git
- VirtualBox image with preconfigured environment
- You can use any IDE you want
  - But only JBDS is supported by teachers

Java EE 6

- Profiles
- CDI 1.0
- Dependency Injection for Java 1.0
- EJB 3.1
- JAX-RS 1.1
- JPA 2.0
- Bean Validation 1.0
- Servlets 3.0
- JSF 2.0
- Stax 1.0

Profiles

- Simplification and reduction of size of Java EE
- Tailored to different applications
- Can represent sub-set of full Java EE or can contain additional technologies
- Defined through JCP process
- Introduces risks of fragmentation
- One new profile defined in Java EE 6
  - Web Profile

Web Profile

- Servlets
- JSP
- JSF
- CDI
- EJB (Lite)
- JPA
- Bean validation
- JTA

Pruning

- Get rid of obsolete technologies
  - No more EJB2
- Obsolete technologies can become optional in future releases
- No need for a container provider support them
- Based purely on community reactions
- Big companies can still deliver optional components

CDI (1/2)

- Context and Dependency Injection
- Key change in Java EE 6
- Based on Seam development ideas
  - But it is not Seam!
- One principle rules them all
  - All different beans in Java EE can be bind together
    - Java bean
    - EJB
    - JSF backing beans
- Heavily uses annotations
  - Application developer develops and provides its own annotations

CDI (2/2)

- Key features
  - Dependency injection
  - Lifecycle management
  - Event-based interaction
- More to come in separate lesson

EJB - Singletons

- Type of session bean
- A single instance of session bean per JVM
  - Not in cluster
- @Singleton annotation
- Initialized in @PostConstruct method
- Can be initialized during application startup @Startup
- By default fully synchronized
- Can use @Lock(READ) and @Lock write annotations for fine-grained concurrency control

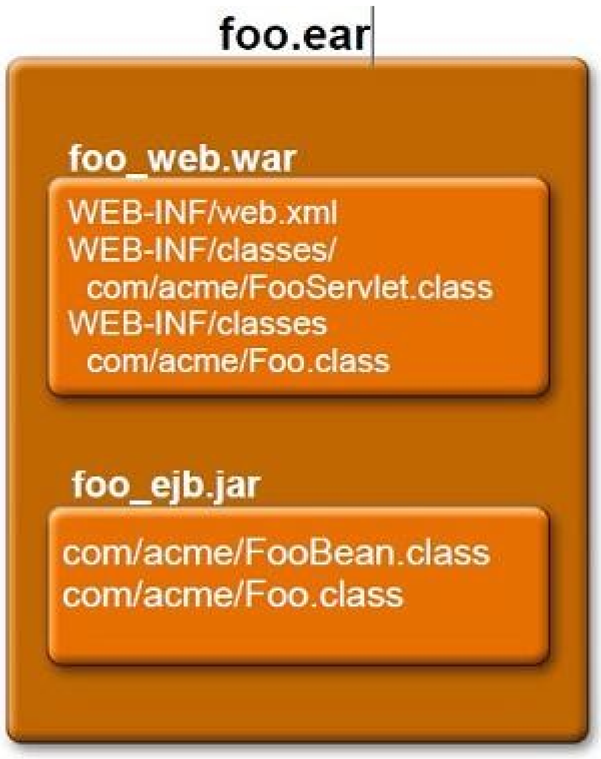
EJB – No-interface view

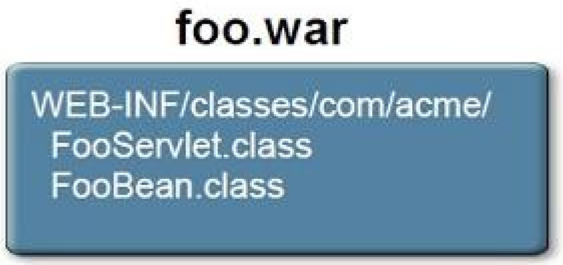
- Simplification of locally used EJBs
- No more necessary to define and implement interface
- Annotated with @Stateless, @Stateful or @Singleton but no interface implemented
- Contract are visible methods for clients
- Can be only injected – never use new keyword
- @LocalBean
  - To define no-interface view if any @Local or @Remote view exists

EJB – Asynchronous invocation

- @Asynchronous
- Method returns
  - void – one-way invocation
  - Future<V> - to return value
- Improves scalability

EJB – Simplified packaging





EJB – EJB Lite

- Component model for Web profile
- Session beans
- Local and no interface access
- Interceptors
- Transactions
- Security

JAX-RS

- Easy way to add REST support for Java EE app
- What is REST?
- Annotation-based configuration
- Works both for Java beans and EJBs

JAX-RS - Annotations

- @Path
  - Binds a method or class to URL pattern
- @GET, @POST, @PUT, @DELETE
  - Binds a method to an HTTP verb
- Method parameters can be injected
  - @PathParam, @QueryParam, @MatrixParam, @HeaderParam, ...

JAX-RS – Wire format

- A direct value return from method
  - byte[], String, char[], streams, ...
- A custom Java class annotated with JAXB annotations
  - Translated into XML, JSON, YAML, ...
- Method is annotated with @Produces, @Consumes to denote MIME type (and encoding) to be used over-the-wire

JPA – Overview

- Extended mapping features
  - Uni-/bi-directional relations
  - Collections of basic data types
- Standardized property names
- Shared cache API

JPA – new JPQL operators

- COALESCE - selects firs non-null value
  - SELECT Name, COALESCE(e.work_phone, e.home_phone) phone FROM Employee e
- NULLIF – substitute defined value with NULL

  SELECT AVG(NULLIF(e.salary, -99999)) FROM Employee e
- INDEX – works with the index of an item in the list
  - SELECT t FROM CreditCard c JOIN c.transactionHistory t WHERE INDEX(t) BETWEEN 0 AND 9
- TYPE – returns type of the entity
  - SELECT o FROM OrderEntity o WHERE TYPE(o) <> RecurringOrderEntity

JPA – Criteria API

- Dynamic construction of JPQL queries
- Typical use case – filter form
- Represents type-safe DSL
  - Needs a static or dynamic metamodel information
- Fallback
  - Use strings – loses type-safety

JPA - Locking

- Optimistic locking supported before
  - READ renamed to OPTIMISTIC
  - WRITE renamed to OPTIMISTIC_FORCE_INCREMENT
- Pessimistic locking supported now
  - PESSIMISTIC_READ
  - PESSIMISTIC_WRITE
  - PESSIMISTIC_FORCE_INCREMENT
- No locking allowed
  - NONE

Bean Validation - Overview

- All applications validate data
  - Presentation layer
  - Business layer
  - Database layer
- Bean validation centralizes validation information
  - Annotation-based on JPA entities
- Bean validation integrated with
  - JSF
  - JPA

Bean Validation - Effects

- JSF
  - JavaScript to validate before for submit
  - In validate phase of JSF life-cycle
- JPA
  - Validates during JPA operations
  - Generated SQL contains constraints to add validation at the database level

Bean Validation - Constraints

- Built-in
  - @Null, @NotNull
  - @AssertTrue, @AssertFalse
  - @Min, @Max, @Size
  - @Past, @Future
  - @Pattern
- Extensible
  - Define your own annotation
  - Compose existing annotations to create a new constraint

Bean Validation - API

- Use javax.validation.Validator
  - Validates the whole object
  - Validates selected properties
  - Provides the constraints associated with class
- ValidatorFactory
  - Validation.buildDefaultValidatorFactory()
- Validator
  - ValidatorFactory.getValidation()

Servlets - Annotations

- Used to configure metadata located in web.xml
- @WebServlet
  - Name, urlPattern
- @WebFilter
  - Name, parameters, urlPattern
- @WebInitParam
  - Definition of servlet initial paramter values
- @WebListener
  - Registration of lifecycle listener – context, servlet, ...

Servlets – Web fragments

- Enables modularization of web.xml DD
- It is possible to define parts of web.xml in separate files
  - Using large web-apps
  - Using 3rd party frameworks
- Order of processing can be set in
  - Main web.xml
  - In each web fragment

Servlets – Async processing

- Allows to re-use servlet threads for long-running operations
- Save resources – less servlets needed
- Improve scalability
- Good for AJAX
- @WebServlet(asyncSupported=true)
- AsyncContext
  - request.startAsync(request, response)
- Continue processing in separate thread and call AsyncContex.complete()
- AsyncListener
  - Informed about completion or timeout

JSF - Facelets

- Designed by community
- Plain XHTML file
- Faster parsing
- No more JSP/JSF mismatch
  - Frequent JSP recompilation during JSF development
  - JSF/JSF lifecycle incompatibility
    - Can lead to different rendering

JSF - Templating

- Define template files with regions
  - <ui:insert name=”...”>
- Define implementation pages
  - <ui:composition template="...">
    - Refers to template file
  - <ui:define name=”...”>
    - Refers to variable region

JSF - AJAX

- What is AJAX
- Partial view render and traversal supported
- <f:ajax execute="..." render="..."/>
- <f:ajax listener="..." render="..."/>
  - Re-renders part of the page based on the event generated
- JavaScript API provided for manual JavaScript development

OpenShift

- Platform-as-a-Service
- IaaS, SaaS, PaaS
- http://openshift.redhat.com
- PHP hosting on steroids
- Controlled from JBDS or Git CLI

Setup IDE

- Download JBDS 5 M5 from
  - http://devstudio.jboss.com/earlyaccess/
- Download JBoss AS 7 from
  - http://www.jboss.org/as7
- Unzip JBoss AS 7
- Install JBDS
  - java -jar ...
  - Direct JBDS to the directory with AS 7 during installation
- And you are done :-)

Questions?

jiri.pechanec@redhat.com | www.jboss.org