



PV243 Pokročilé Java technologie: JBoss

Část 1. - Úvod, Java EE 7

Jiří Pechanec

únor 2017

Agenda

- Introduction
- Course information
- Tools
- Java EE 7

Motivation

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

Organization

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

Topics

- Introduction to Java EE 7
- CDI 1.1, EJB 3.2
- HTML5 – WebSockets, JSON, REST
- Datagrids, Infinispan
- Java EE security
- Clustering and scalability in WildFly 10
- (Management and monitoring)

Tools

- WildFly 10 Final
- JBoss Developer Studio 10.3
- Maven
- Git
- You can use any IDE you want
 - But only JBDS is supported by teachers

Java EE 7 (1/2)

- New/major upgrade specifications for web 2.0 apps
 - WebSocket 1.0
 - JSON-P 1.0
 - JAX-RS 2.0
- New/major upgrade specifications for backend
 - Batch Applications 1.0
 - Concurrency 1.0
 - JMS 2.0

Java EE 7 (2/2)

- Minor updates (not comprehensive)
 - CDI 1.1
 - JSF 2.2
 - Bean Validation 1.1
 - JPA 1.1
 - JTA 1.2
 - Servlet 3.1
- Other updates
 - EL 3.0

Concurrency Utilities 1.0 (1/2)

- Application managed multithreading
- Based on `java.util.concurrent` infrastructure
 - `ExecutorService`
 - `ScheduleExecutorService`
 - `ThreadFactory`

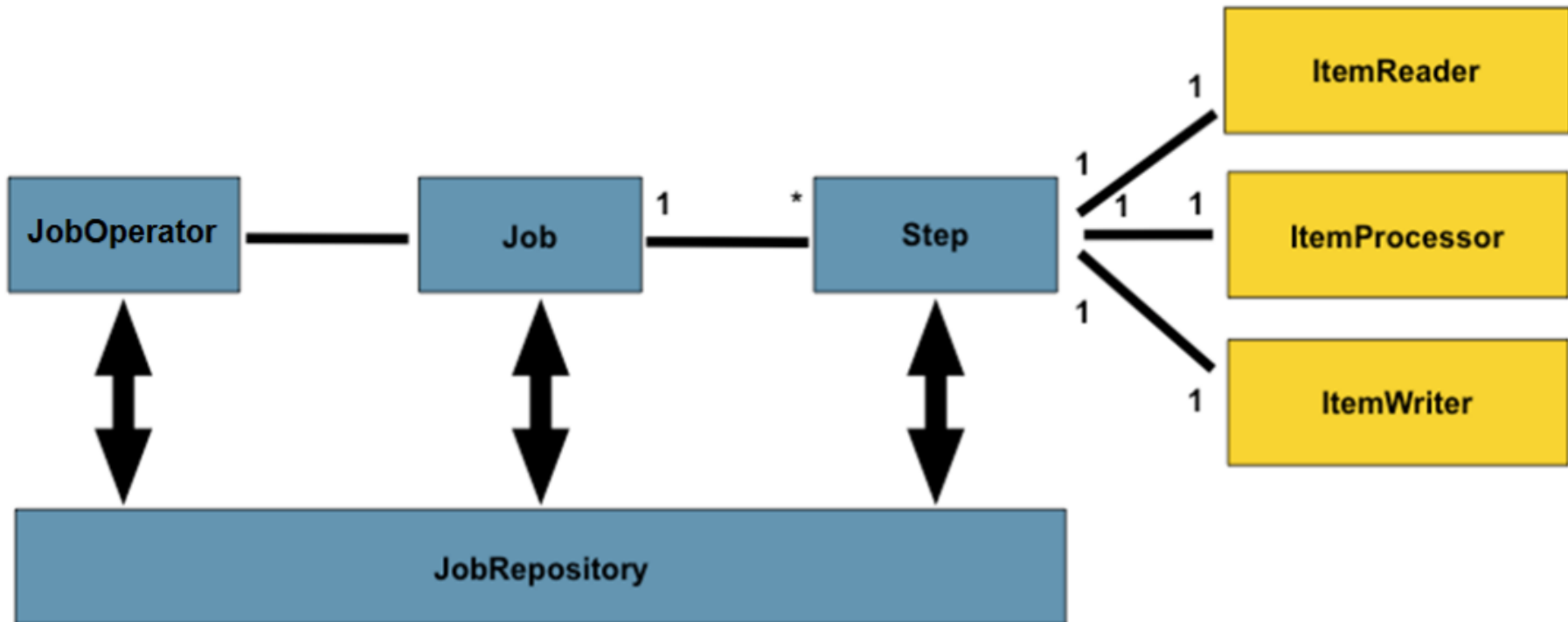
Concurrency Utilities 1.0 (2/2)

- Java EE 7 counterparts
 - ManagedExecutorService
 - ManagedScheduledExecutorService
 - ManagedThreadFactory
- Configured in container
 - @Resource
- ContextService
 - Allows execution of task in a context of current thread
 - e.g. security
- Tasks/threads run in its own transaction

Batch Applications 1.0 (1/8)

- For non-interactive long-running tasks
- Usually very big, running in the night
- Typical uses cases
 - Billing
 - Clearing (banks)
 - Bulk data processing
 - Mass customer operations

Batch Applications 1.0 (2/8)



Batch Applications 1.0 (3/8)

- Job is a sequence of Steps
- Step is
 - A simple Batchlet
 - Or a combination of reader/processor/writer
- JobOperator manages jobs (starts/stops) and is an interface to JobRepository
- JobRepository stores information about running and past Jobs
- Id of classes
 - CDI name (@Named)
 - XML file

Batch Applications 1.0 (4/8)

- Packaging
 - Class ids in
 - META-INF/batch.xml
 - Job definitions
 - META-INF/batch-jobs/<jobname>.xml
- API entry point
 - `BatchRuntime.getJobOperator()`

Batch Applications 1.0 (5/8)

- Batchlet processes everything in one phase
- Reader/processor/writer supports chunks
 - Processing and aggregation of subset of records in a single transaction
 - Checkpoints
 - Allows to restart a job from last processed chunk
 - Defined by item count, time or custom
 - Errors/exceptions
 - Could be ignored (skipped) or retried

Batch Applications 1.0 (6/8)

- Batchlet
 - Batchlet, AbstractBatchlet
- Reader
 - ItemReader, AbstractItemReader
- Processor
 - ItemProcessor
- Writer
 - ItemWriter, AbstractItemWriter
- Job/step can have property - defined in XML
- @BatchProperty

Batch Applications 1.0 (7/8)

- Injectable
 - @BatchProperty
 - JobContext
 - StepContext

Batch Applications 1.0 (8/8)

- Chunks can be executed in parallel – partition
- Jobs can contain complex processes
 - Flow
 - Split
 - Decision
 - Transition
 - Fail/End/Stop

JMS 2.0 (1/2)

- Aligned with Java 7 and CDI
- Much easier used and less verbose code
- Auto-closeable objects
 - Connection, Session, MessageProducer/Consumer
 - Significantly improves resource handling
- Admin object definitions
 - Simpler deployment – destinations and CF configurable in code
 - @JMSTDestinationDefinition(s)
 - @JMSConnectionFactoryDefinition(s)

JMS 2.0 (2/2)

- New construct – JMSContext
 - @Inject
 - Replaces Connection, Session, MessageProducer/Consumer
 - Fully managed by container – no resource handling
- Further configurable with
 - @JMSConnectionFactory
 - @JMSPasswordCredential
 - @JMSSESSIONMode

EL 3.0 (1/2)

- Standalone API
 - Can use EL for application purpose
 - Works in Java SE
- New operators
 - += (string concatenation)
 - ; (expression chaining – like , in C)
- Can access static fields and methods
- Lambdas
 - Ported from Java 8
- Stream API

EL 3.0 (2/2)

- Lambdas
 - Ported from Java 8
 - Functions can be assigned to variables
 - FP-like features
- Stream API
 - Collection can be converted to stream, FP-like functions supported
 - filter, map, distinct, sorted, forEach
 - reduce, max, min, count, sum
 - anyMatch, allMatch, findFirst

Servlet 3.1

- Support for NIO in async execution
 - Reads and writes are event based
 - ReadListener
 - WriteListener
- HTTP Protocol Upgrade Support
 - e.g. for HTTP → WebSocket

JTA 1.2

- Allows CDI bean methods to run in transaction context
 - @Transactional - required, requires new, etc.
 - Handled via CDI interceptors
- New CDI scope - @TransactionScoped
 - The bean's lifecycle is tied to a running transaction

Bean Validation 1.1

- CDI Integration
 - Can @Inject Validator and ValidatorFactory
- Design by contract
 - Constructor and method params can be validated
- Support for EL in error messages
 - Incl. locales
- Group conversion
 - @ConvertGroup
 - Maps requested validation group to the group actually used

JPA 2.1 (1/2)

- Criteria API – bulk updates
 - createCriteriaUpdate
 - createCriteriaDelete
 - Portable way for big database changes
- Schema generation
 - Standardized across JPA providers

JPA 2.1 (2/2)

- Stored procedures
 - create(Named)StoredProcedureQuery
 - @NamedStoredProcedureQuery/ies
- Native FUNCTION call
 - Used in JPQL query
 - `FUNCTION(f_name {,args}*)`
 - Can execute any SQL native function
 - Potentially non-portable

OpenShift

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

WildFly Swarm

- <http://wildfly-swarm.io/>
- Microservices bandwagon
- Container as a fatjar
- Just a Maven plug-in
- Strip down WildFly to services only used
- Custom main()
- 3rd party integration
 - NetflixOSS, Spring
- Additional services
 - Logging - logstash
 - Management - Jolokia
 - Clustering - Consul

Setup IDE

- Download JBDS 10.3 from
 - <https://developers.redhat.com/>
- Download WildFly 10 from
 - <http://www.wildfly.org/download/>
- Unzip WildFly 10
- Install JBDS
 - `java -jar ...`
 - Direct JBDS to the directory with WildFly 10 during installation
- And you are done :-)



Questions?

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