software pilots

TRIFORK.

Spring User Group
Spring 3.0

Agenda

- Where would we like to go with the Spring User group?
- The members expectations & wishes.
- Exchange experience & knowledge with Spring projects.
- Talks about acknowledged technologies.
- How many meet ups during the year?

About

- Joakim Recht, MSc, CS

- Software Pilot @ Trifork for 4 years
  - Mostly Open Source projects in the public sector
  - All things Java: J(2)EE, Spring, WS, REST, Security, etc
  - Training: Spring, patterns, UML, JEE, etc

What's new in Spring 3.0

- Spring 3.0M4 released 2009-08-10
- Java-based configuration
- REST support
- Spring Expression Language
- Portlet 2.0 support
- Declarative model validation
- A lot of smaller changes

Agenda

- General changes
- Spring Beans
- Threads and Scheduling
- REST Support / @MVC
- JDBC/DAO/ORM
- Spring EL
- Model validation

General Changes

- Deprecated JDK1.4 support
  - Junit 3 (use Junit 4 or TestNG instead)
  - MVC Controller hierarchy (use @MVC instead)
  - Generics and varargs introduced where possible
- A lot of dependencies upgraded
- In many cases just a drop-in replacement

Spring Beans

- Autowire scalar values from PropertyPlaceholders:
  - @Value("${property"}
- More annotations:
  - @Lazy, @DependsOn, @Primary
- New type conversion infrastructure
  - Supercedes standard PropertyEditors

Configuring beans in Java

- Not a XML replacement
  - Still requires context:annotation-config or context:component-scan
- Define beans in a type-safe Java-class
- Annotate classes with @Configuration
- Annotate each method with @Bean
- Bean methods can refer to each other

Threads and Scheduling

- Much better support for java.util.concurrent
- Classes and methods can be marked as asynchronous with @Async
  - Must be void or return java.util.concurrent.Future
  - AOP will execute the methods in an executor
- Schedule methods with @Scheduled
- New task namespace
  - <task:annotation-driven />

REST and @MVC

- New RestTemplate for accessing REST resources
  - Makes it possible to get a resource from a url and convert it into a types object
  - Common operations: GET, POST, PUT, DELETE, etc
- New package with low-level HTTP support: org.springframework.http

REST Services

- REST Services are @MVC controllers
- A number of new annotations
  - @PathVariable binds a part of the url to a method parameter
  - @RequestBody binds the body to a method parameter
  - @RequestHeader binds a HTTP header to a method parameter
  - @CookieValue binds a cookie value to a method parameter

REST Services

- @ExceptionHandler an be added to methods to handle certain exceptions in different views
- New view types for feeds:
  - AbstractAtomFeedView
  - AbstractRssFeedView
- servletContext and servletConfig now exists as beans in the context

JDBC/DAO/ORM

- Removed legacy support code
- Generified callbacks and templates
- Built-in support for embedded databases
  - <jdbc:embedded-database>
  - Load DDL automatically from script

Spring Expression Language

- New EL, somewhat like JSTL, OGNL, and others – just a little different
- Used everywhere in Spring where expressions are possible
- Expressions are placed in #{…}

SpEL Features and Examples

- Property access: #{bean.messages}
- Create instance: #{new java.io.File(bean.path)}
- Static method: #{T(String).format('%s', bean.message)}
- Properties access: #{props['jdbc.url']}
- Safe navigation: #{bean?.inner?.prop}
- Collection select: #{users.?[active == true]}

Model Validation

- Declarative model validation using annotations
- Support for JSR-303 Bean Validation
- @Constraint, @NotNull, @Max, …
- Not in M4 yet

Roadmap

- 3.0 RC1 in September 2009
  - Feature-complete and documented
- 3.0 GA in October
  - Of cource, this depends on feedback on RC1
- 3.1
  - Early 2010
  - Full support for JEE6