HOWTO - Spring with OJB

by Jason McKerr

Table of contents

1. Spring with OJB

This Howto refers to external articles describe the usage of Spring with OJB's PB-api.

Included are three articles that are in-depth How-to's for using Spring Framework's *declarative transaction engine* with OJB's *Persistence Broker* implementation. For each article, I have included a complete sample application with an embedded database that you can run.
They are simple to setup and require minimal configuration.

1.1. Spring, OJB, and Struts, getting started

The first article covers a basic configuration and setup for integrating Spring with OJB. It covers:

- Connecting to the database using Spring's local datasources (*LocalOjbConfigurer*)
- Register the appropriate transactionManager for use with OJB (PersistenceBrokerTransactionManager)
- Creating beans and interfaces for use with the database
- Having Spring handle transactions for the beans in declarative fashion

The article link is: Spring, OJB, and Struts, getting started

1.2. Spring, OJB, and Struts. Version 2 with DAO.

The second article covers more advanced Spring usage, separating out the database implementation code from your Data Access Objects. This is an important abstraction layer for controlling data access ubiquitously.

The article link is: Spring, OJB, and Struts. Version 2 with DAO.

1.3. Spring, OJB, and Struts. Version 3: JNDI Datasources and Caching

The Third article covers using OJB's datasourcing and caching strategies.
The first two articles require local datasources and Per Broker caching because of the way that the *OjbConfigurer* works.
Now that we've completed basics of integrating OJB in Spring, this last entry shows how to use JNDI datasourcing with OJB and Spring, which in turn allows users to get back to other OJB caching strategies.

The article link is: Spring, OJB, and Struts. Version 3: JNDI Datasources and Caching