Coherence Spring Reference Documentation

Gunnar Hillert

Version 3.0.0-SNAPSHOT

Legal

Oracle licenses the Oracle Coherence Spring project under the The Universal Permissive License (UPL), Version 1.0.

The Universal Permissive License (UPL), Version 1.0

Subject to the condition set forth below, permission is hereby granted to any person obtaining a copy of this software, associated documentation and/or data (collectively the "Software"), free of charge and under any and all copyright rights in the Software, and any and all patent rights owned or freely licensable by each licensor hereunder covering either (i) the unmodified Software as contributed to or provided by such licensor, or (ii) the Larger Works (as defined below), to deal in both

(a) the Software, and (b) any piece of software and/or hardware listed in the lrgrwrks.txt file if one is included with the Software (each a "Larger Work" to which the Software is contributed by such licensors),

without restriction, including without limitation the rights to copy, create derivative works of, display, perform, and distribute the Software and make, use, sell, offer for sale, import, export, have made, and have sold the Software and the Larger Work(s), and to sublicense the foregoing rights on either these or other terms.

This license is subject to the following condition: The above copyright notice and either this complete permission notice or at a minimum a reference to the UPL must be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 1. Coherence Spring Documentation

Welcome to the reference documentation of Coherence Spring, a collection of several libraries that will help you to integrate Coherence with the wider Spring ecosystem.

This section provides a brief overview of the Coherence Spring reference documentation.

1.1. About the Documentation

The Coherence Spring reference guide is available as:

- Multi-page HTML
- Single page HTML
- PDF

1.2. Getting Help

If you have trouble with Spring Coherence, we would like to help.

- Try the How-to documents. They provide solutions to the most common questions.
- Learn the Coherence basics. All Spring Coherence module directly support Oracle Coherence. Check the Coherence CE web-site for general Coherence targeted reference documentation.
- Learn the Spring basics. Spring Coherence builds on several other Spring projects. Check the spring.io web-site for general reference documentation. If you are starting out with Spring, try one of the guides.
- Ask a question. Chat with us directly on Slack. We also monitor stackoverflow.com for questions tagged with oracle-coherence.
- Report bugs with Spring Coherence via GitHub Issues.

All of Coherence Spring is open source, including the documentation. If you find problems with the docs or if you want to improve them, please get involved.

1.3. What is new?

In order to see what changes were made from earlier versions of Coherence Spring, see the Change History as well as the GitHub Releases page.

1.4. First Steps

If you are getting started with Spring Boot or 'Spring' in general, start with the following topics:

- **From scratch:** Overview | Requirements | Installation
- **Tutorial:** Part 1 | Part 2

- **Running your example:** Part 1 | Part 2

Chapter 2. Quickstart

We provide various demos to illustrate basic usage of Oracle Coherence when using it with Spring:

- Coherence Spring Demo

This repository contains essentially 2 versions of the same app:

- `coherence-spring-demo-classic` Provides a demo using Spring Framework without Spring Boot
- `coherence-spring-demo-boot` Provide a demo using Spring Boot

Chapter 3. Coherence Spring Core

This section dives into the Coherence Spring Core module. Coherence Spring Core provides the basic support support for the Spring Framework.

3.1. Getting Started

TBD

Chapter 4. Coherence Spring Cache

This section dives into the Coherence Spring Cache module. It explains how to use Coherence's support for the Spring Framework's Cache Abstraction.

4.1. Introduction

Spring provides its own cache abstraction, allowing you to add caching to Java methods. Coherence Spring provides an implementation of this abstraction for Oracle Coherence.

Spring's Cache abstraction also supports JSR-107 which is also supported by Oracle Coherence. As such you have another alternative for setting up caching.

4.2. Configuring Coherence Cache for Spring

As a start, please familiarize yourself with Spring's Cache Abstraction by reading the relevant section of Spring's reference documentation.

Properties

```
example.property.alpha=a
```

Yaml

```
example:
  property:
    alpha: a
```

Properties

```
spring.devtools.restart.exclude=static/**,public/**
```

Yaml

```
spring:
  devtools:
    restart:
      exclude: "static/**,public/**"
```

Example 1. Creating a CoherenceInstance

Java

```
@Configuration
@EnableCaching
public class CacheConfiguration {

    @Bean
    public CoherenceInstance coherenceInstance() {
      return new CoherenceInstance();
    }

    @Bean
    public CacheManager cacheManager(CoherenceInstance coherenceInstance) {
      return new CoherenceCacheManager(coherenceInstance);
    }
}
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cache="http://www.springframework.org/schema/cache"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/cache
https://www.springframework.org/schema/cache/spring-cache.xsd">

  <cache:annotation-driven/>

  <bean id="coherenceInstance"
class="com.oracle.coherence.spring.CoherenceInstance"/>

  <bean id="cacheManager"
class="com.oracle.coherence.spring.cache.CoherenceCacheManager">
    <constructor-arg ref="coherenceInstance"/>
  </bean>
</beans>
```

Chapter 5. Coherence Spring Session

This section dives into the Coherence Spring Session module. It explains how to use Coherence's support for Spring Session.

5.1. Getting Started

TBD

Chapter 6. Coherence Spring Data

This section dives into the Coherence Spring Data module. It explains how to use Coherence's support for Spring Data repositories.

6.1. Getting Started

TBD

Chapter 7. Coherence Spring Boot

This section dives into the Coherence Spring Boot module. It explains how to use Coherence's dedicated support for Spring Boot, e.g. Autoconfiguration.

7.1. Getting Started

Maven

```
<dependencies>
    <dependency>
        <groupId>com.oracle.coherence.spring</groupId>
        <artifactId>coherence-spring-boot-starter</artifactId>
        <version>3.0.0-SNAPSHOT</version>
    </dependency>
</dependencies>
```

Gradle

```
dependencies {
    compile("com.oracle.coherence.spring:coherence-spring-boot-starter:3.0.0-
SNAPSHOT")
}
```

7.2. Using Coherence with Spring Boot

7.3. Using Coherence as Spring Caching Provider

Coherence provides dedicated support for Spring Boot in order to further simplify configuration options for Spring.

If caching is enabled via `@EnableCaching`, Coherence Autoconfiguration will it automatically provide a `CacheManager` to the ApplicationContext, however only if no `CacheManager` was configured explicitly beforehand.

Caches can be created on startup by setting the property. If a custom `ConfigurationBuilder` bean is defined, it is used to customize the caches.

The support of Infinispan in Spring Boot is restricted to the embedded mode and is quite basic. If you want more options, you should use the official Infinispan Spring Boot starter instead. See Infinispan's documentation for more details.

Chapter 8. Appendices