

Spring extension - Integration modes

Introduction

During the development of the 1.0 version of the API, several users attempted to integrate Restlet with Spring. They were especially trying to use the XML-based bean wiring feature of Spring. This resulted in several examples available on the [community Wiki](#)¹.

In order to facilitate this integration, two dedicated Spring extension were added to the Restlet. It allows us to provide several integration modes.

Restlet as main container

In the first mode, the goal is to leverage the concept of Restlet Application and all the associated services, as well as the transparent deployment to either a Servlet container (using the adapter `ServerServlet` extension class) or using a standalone HTTP server connector. For this, you can leverage the `SpringContext` class which is a Spring's `GenericApplicationContext` subclass. You can associate a list of XML or property configuration URIs (`file:///` or `war:///` URIs) in order to have Spring auto-instantiate and wire your Restlet beans.

Spring as main container

In the second mode, the goal is to leverage the concept of Spring Web Application as an alternative to the Restlet Application. This is sometimes required when the Restlet code is part of a larger Spring-based Web application, with dependencies on the Servlet API for example.

Initially, it was hard to achieve this integration because the Servlet extension, and especially the `ServerServlet` adapter class was assuming the usage of a Restlet Application. Later we added a lighter adapter based on the `ServletConverter` class that lets you directly instantiate Restlet Routers, Finders and Resources from your existing Servlet-based Spring code. You can check the Javadocs for details.

Finally, there is also a `SpringFinder` class available in the Spring extension. It hasn't any specific dependency to Spring, but the addition of a parameter-less `createResource()` method allows the usage of the Spring's "lookup-method" mechanism.

In Restlet 1.1, the Spring extensions received several contributions, increasing the number of ways to integrate Restlet with Spring. There is now a `RestletFrameworkServlet`, a `SpringServerServlet`, `SpringBeanFinder` and `SpringBeanRouter`. Please check the Javadocs of the **`org.restlet.ext.spring`** and **`com.noelios.restlet.ext.spring`** modules for more details.

1. [../../../../../../../../](#)