

# SpringIntegration

## Spring Integration Component

The **spring-integration** component provides a bridge for Camel components to talk to [spring integration endpoints](#).

Maven users will need to add the following dependency to their `pom.xml` for this component:

```
xml <dependency> <groupId>org.apache.camel</groupId> <artifactId>camel-spring-integration</artifactId> <version>x.x.x</version> <!-- use the same version as your Camel core version --> </dependency>
```

### URI format

`spring-integration:defaultChannelName[?options]`

Where **defaultChannelName** represents the default channel name which is used by the Spring Integration Spring context. It will equal to the `inputChannel` name for the Spring Integration consumer and the `outputChannel` name for the Spring Integration provider.

You can append query options to the URI in the following format, `?option=value&option=value&...`

### Options

confluenceTableSmall

Name	Type	Description
<code>inputChannel</code>	String	The Spring integration input channel name that this endpoint wants to consume from, where the specified channel name is defined in the Spring context.
<code>outputChannel</code>	String	The Spring integration output channel name that is used to send messages to the Spring integration context.
<code>inOut</code>	String	The exchange pattern that the Spring integration endpoint should use. If <code>inOut=true</code> then a reply channel is expected, either from the Spring Integration Message header or configured on the endpoint.

### Usage

The Spring integration component is a bridge that connects Camel endpoints with Spring integration endpoints through the Spring integration's input channels and output channels. Using this component, we can send Camel messages to Spring Integration endpoints or receive messages from Spring integration endpoints in a Camel routing context.

### Examples

#### Using the Spring integration endpoint

You can set up a Spring integration endpoint using a URI, as follows:

```
{snippet:id=example|lang=xml|url=camel/trunk/components/camel-spring-integration/src/test/resources/org/apache/camel/component/spring/integration/producer.xml} {snippet:id=example|lang=xml|url=camel/trunk/components/camel-spring-integration/src/test/resources/org/apache/camel/component/spring/integration/twoWayConsumer.xml}
```

Or directly using a Spring integration channel name:

```
{snippet:id=example|lang=xml|url=camel/trunk/components/camel-spring-integration/src/test/resources/org/apache/camel/component/spring/integration/springChannelConverter.xml}
```

#### The Source and Target adapter

Spring integration also provides the Spring integration's source and target adapters, which can route messages from a Spring integration channel to a Camel endpoint or from a Camel endpoint to a Spring integration channel.

This example uses the following namespaces:

```
{snippet:id=header|lang=xml|url=camel/trunk/components/camel-spring-integration/src/test/resources/org/apache/camel/component/spring/integration/adapter/CamelTarget.xml}
```

You can bind your source or target to a Camel endpoint as follows:

```
{snippet:id=example|lang=xml|url=camel/trunk/components/camel-spring-integration/src/test/resources/org/apache/camel/component/spring/integration/adapter/CamelTarget.xml} {snippet:id=example|lang=xml|url=camel/trunk/components/camel-spring-integration/src/test/resources/org/apache/camel/component/spring/integration/adapter/CamelSource.xml} Endpoint See Also
```