



Spring Integration 5.2.2



Extends the Spring programming model to support the well-known **Enterprise Integration Patterns**. Spring Integration enables lightweight messaging within Spring-based applications and supports integration with external systems via declarative adapters. Those adapters provide a higher-level of abstraction over Spring's support for remoting, messaging, and scheduling. Spring Integration's primary goal is to provide a simple model for building enterprise integration solutions while maintaining the separation of concerns that is essential for producing maintainable, testable code.

Introduction

Using the Spring Framework encourages developers to code using interfaces and use dependency injection (DI) to provide a Plain Old Java Object (POJO) with the dependencies it needs to perform its tasks. Spring Integration takes this concept one step further, where POJOs are wired together using a messaging paradigm and individual components may not be aware of other components in the application. Such an application is built by assembling fine-grained reusable components to form a higher level of functionality. With careful design, these flows can be modularized and also reused at an even higher level.

In addition to wiring together fine-grained components, Spring Integration provides a wide selection of channel adapters and gateways to communicate with external systems. Channel Adapters are used for one-way integration (send or receive); gateways are used for request/reply scenarios (inbound or outbound). For a full list of adapters and gateways, refer to the reference documentation.

The Spring Cloud Stream project builds on Spring Integration, where Spring Integration is used as an engine for message-driven microservices.

Features

- Implementation of most of the Enterprise Integration Patterns
- Endpoint
- Channel (Point-to-point and Publish/Subscribe)
- Aggregator
- Filter
- Transformer
- Control Bus
- ...
- Integration with External Systems
- ReST/HTTP
- FTP/SFTP
- Twitter
- WebServices (SOAP and ReST)
- TCP/UDP
- JMS
- RabbitMQ