10. Integrating Spring

-Just as we need to connect to the I to be productive, many apps must connect to external systems to perform their work.

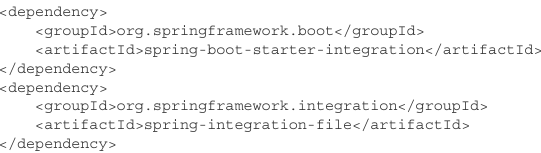
-Spring Integration an implementation of integration patterns that are catalogued in Enterprise Integration Patterns. Each pattern is implemented as a component through which messages ferry data in a pipeline. Using Spring configuration, you can assemble these components into a pipeline through which data flows.

# 10.1 Declaring a simple integration flow

-Spring Integration enables the creation of integration flows through which an app can receive or send data to some resource external to the app itself.

-Create an integration flow that writes data to filesystem

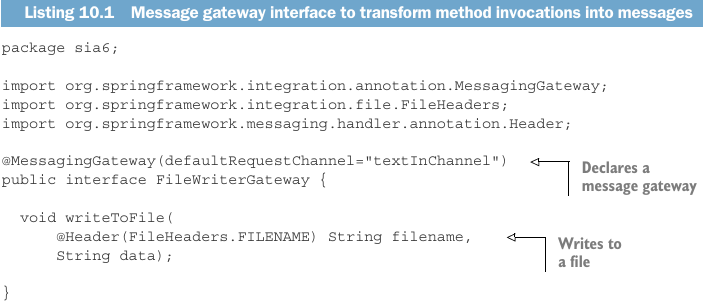
-Add Spring Integration:



+1st dependency: essential to develop a Spring Integration flow, regardless of what the flow may integrate with.

+2nd dependency: one of over 2 dozen endpoint modules used to integrate with external systems. It offers the ability to ingest files from filesystem into an integration flow and/or to write data from a flow to filesystem

-Next, create a way for the app to send data into an integration flow so that it can be written to a file. You will create a gateway interface:



## 10.1.1 Defining integration flows with XML

## 10.1.2 Configuring integration flows in Java

## 10.1.3 Using Spring Integration’s DSL configuration

# 10.2 Surveying the Spring Integration landscape

## 10.2.1 Message channels

## 10.2.2 Filters

## 10.2.3 Transformers

## 10.2.4 Routers

## 10.2.5 Splitters

## 10.2.6 Service activators

## 10.2.7 Gateways

## 10.2.8 Channel adapters

## 10.2.9 Endpoint modules

# 10.3 Creating an email integration flow

# -Summary