

Introducción al middleware

Spring integration



Instituto de
Computación



Facultad de
Ingeniería



Universidad de la
República de Uruguay

Spring integration

- ❑ Es un framework de integración desarrollado en Java que extiende el framework Spring
- ❑ Implementa los patrones de mensajería EIP
- ❑ Provee conectores para integrar diferentes sistemas externos
 - JMS, File, TCP, HTTP, etc



Spring integration

- ❑ Permite implementar “flujos de integración” entre sistemas utilizando los patrones EIP
- ❑ <https://spring.io/projects/spring-integration>

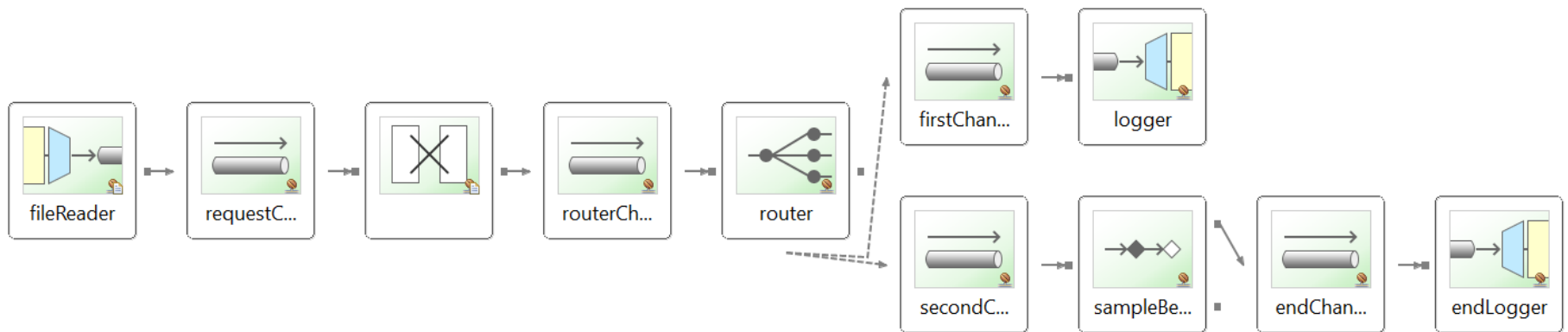


Spring integration

- ❑ Provee dos formas para definir flujos de integración:
 - Lenguaje XML
 - Lenguaje DSL (Domain Specific Language) en Java
- ❑ En el curso recomendamos utilizar la opción XML por su simplicidad.
- ❑ Si ya se conoce el Framework de Spring, puede optarse el uso del DSL



Ejemplo



Ejemplo XML

```
<int-file:inbound-channel-adapter id="fileReader"
    directory="c:/temp/files" filename-pattern="*.dat" channel="requestChannel">
    <int:poller fixed-delay="5000" />
</int-file:inbound-channel-adapter>

<int:channel id="requestChannel"/>

<int-file:file-to-string-transformer input-channel="requestChannel" output-channel="routerChannel"
    delete-files="true"/>

<int:channel id="routerChannel"/>

<int:recipient-list-router id="router" input-channel="routerChannel">
    <int:recipient channel="firstChannel" />
    <int:recipient channel="secondChannel" />
</int:recipient-list-router>

<int:channel id="firstChannel"/>
<int:logging-channel-adapter id="logger" channel="firstChannel" expression="'See your headers: '+headers"/>

<int:channel id="secondChannel"/>
<int:service-activator id="sampleBeanInvocation"
    input-channel="secondChannel" output-channel="endChannel"
    ref="sampleBean" method="process"/>

<int:channel id="endChannel"/>
<int:logging-channel-adapter id="endLogger" channel="endChannel"
    expression="'This is the end, see your message: '+payload"/>

<bean id="sampleBean" class="edu.fing.middleware.ejemplo.SampleBean"/>
```



Ejemplo XML

- ❑ El flujo de integración lee archivos de una carpeta y genera una copia que será registrada en consola por dos tipos de logger
 - Logger Spring Integration
 - Logger personalizado



Spring Beans

- ❑ Los Beans son conceptos clave en el uso del Framework Spring.
- ❑ *In Spring, the objects that form the backbone of your application and that are managed by the Spring IoC container are called beans. A bean is an object that is instantiated, assembled, and otherwise managed by a Spring IoC container.*



Spring Beans

- ❑ Sin embargo, en el curso se utilizarán en su forma más básica, permitiendo agregar lógica personalizada a un “flujo de integración”

- ❑ Advertencia:
 - No tomar el uso que se le da en el curso a los Beans como una buena práctica
 - El curso está enfocado a comprender patrones de mensajería EIPs



Ejemplo

```
package edu.fing.middleware.ejemplo;

import org.springframework.messaging.Message;
import org.springframework.messaging.support.MessageBuilder;

public class SampleBean {

    public Message<String> process(Message<String> message){
        String payload = message.getPayload().toUpperCase();
        return MessageBuilder.withPayload(payload).build();
    }
}
```



Ejemplo

- ❑ Los Beans deben tener un método para procesar el mensajes
- ❑ Este método puede tener el parámetro Message de Spring Integration para tener acceso a todo el mensaje
- ❑ El parámetro retornado por el método será enviado al siguiente paso del flujo de integración.
 - Se recomienda retornar un Message
 - Se debe usar MessageBuilder para crear el Mensaje



Ejemplo XML

```

<int-file:inbound-channel-adapter id="fileReader"
    directory="c:/temp/files" filename-pattern="*.dat" channel="requestChannel">
    <int:poller fixed-delay="5000" />
</int-file:inbound-channel-adapter>

<int:channel id="requestChannel"/>

<int-file:file-to-string-transformer input-channel="requestChannel" output-channel="routerChannel"
    delete-files="true"/>

<int:channel id="routerChannel"/>

<int:recipient-list-router id="router" input-channel="routerChannel">
    <int:recipient channel="firstChannel" />
    <int:recipient channel="secondChannel" />
</int:recipient-list-router>

<int:channel id="firstChannel"/>
<int:logging-channel-adapter id="logger" channel="firstChannel" expression="'See your headers: '+headers"/>

<int:channel id="secondChannel"/>
<int:service-activator id="sampleBeanInvocation"
    input-channel="secondChannel" output-channel="endChannel"
    ref="sampleBean" method="process"/>

<int:channel id="endChannel"/>
<int:logging-channel-adapter id="endLogger" channel="endChannel"
    expression="'This is the end, see your message: '+payload"/>

<bean id="sampleBean" class="edu.fing.middleware.ejemplo.SampleBean"/>

```

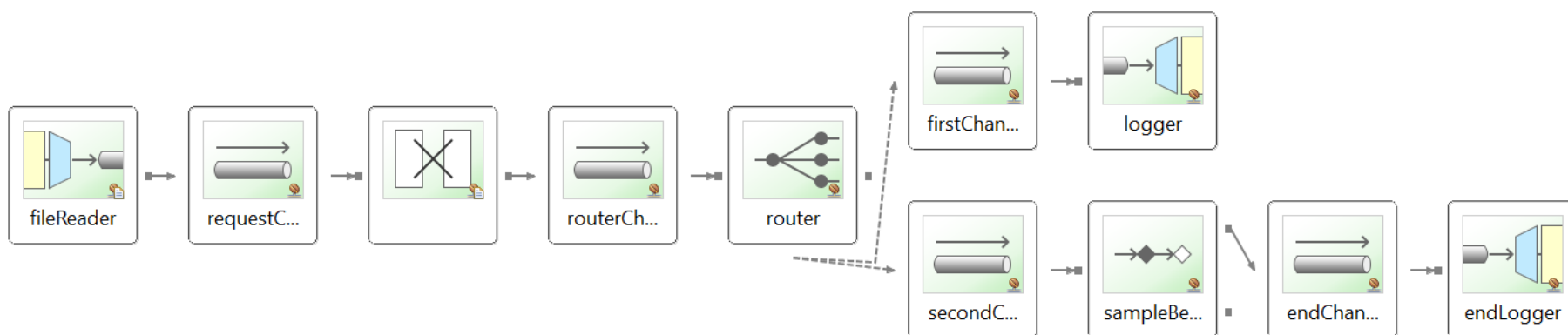


Spring Tool Suite

- ❑ Entorno de desarrollo para aplicaciones Spring y en particular, flujos de integración con Spring Integration
- ❑ Permite visualizar gráficamente en un grafo los flujos de integración
 - Permite reducir la curva de aprendizaje
 - Grafo complejo en integraciones grandes
- ❑ Ayuda en el desarrollo del flujo de integración XML
 - Utilizar la version 3.9.x ya que version 4.x no provee asistencia



Ejemplo



Correr una aplicación SI

```
import org.springframework.context.support.AbstractApplicationContext;

/*
 * This Java source file was generated by the Gradle 'init' task.
 */
public class App {

    public static void main(String[] args) {

        AbstractApplicationContext context = new ClassPathXmlApplicationContext
            ("/META-INF/spring/integration/integration.xml", App.class);
    }
}
```



Conclusiones

- ❑ Spring integration es un framework utilizado en el curso para el desarrollo de flujo de integración entre sistemas

- ❑ Se presentó un ejemplo de uso
 - Publicado en el EVA

- ❑ Se presentó el STS como ambiente de desarrollo.

