

Enrichers

Express Spring Integration



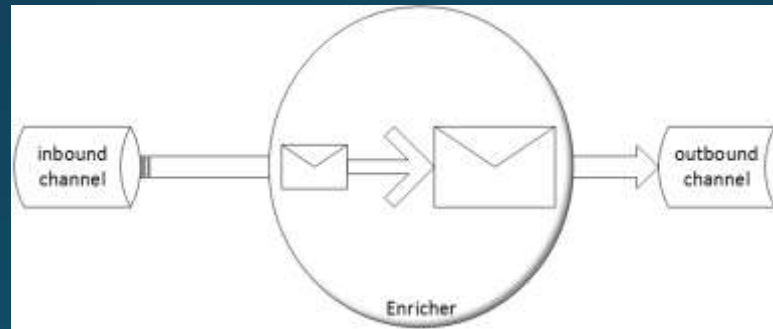
SI Routers Review

- Routers act as message distribution components.
 - They take messages from one channel and distribute the message to one or more other channels.
 - Some routers must inspect a message to determine where to send the message.
 - Other routers simply spray the message to all receiving channels.
- Content routers examine the incoming message content and use the payload type or header value to determine which channel receives the message.
- Recipient list routers don't examine the messages. They deliver the messages to all listed recipient channels.

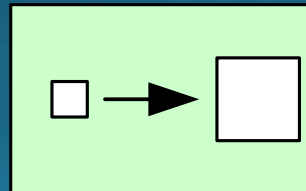


SI Enrichers

- Enrichers add information or content to an SI message.
 - Enrichers pull messages from a channel, add data or information to the message's header or payload, and post the message to another channel.



- Enrichers are considered variants of transformers.
- Spring provides a number of enrichers out of the box.
- You can also create your own enrichment customization by using additional SI components.
- The EIP icon for an enricher is:



Example Header Enricher

- Below is the configuration for a simple header enricher

```
<int:header-enricher input-channel="in" output-channel="out">  
  <int:priority value="HIGHEST"/>  
  <int:header name="paymentAccountId" value="234562"/>  
</int:header-enricher>
```

- Note there are some standard headers, like priority, that are built into configuration.
- Other generic or custom headers can be added with <header> elements (like paymentAccountId above).



Example Payload Enricher

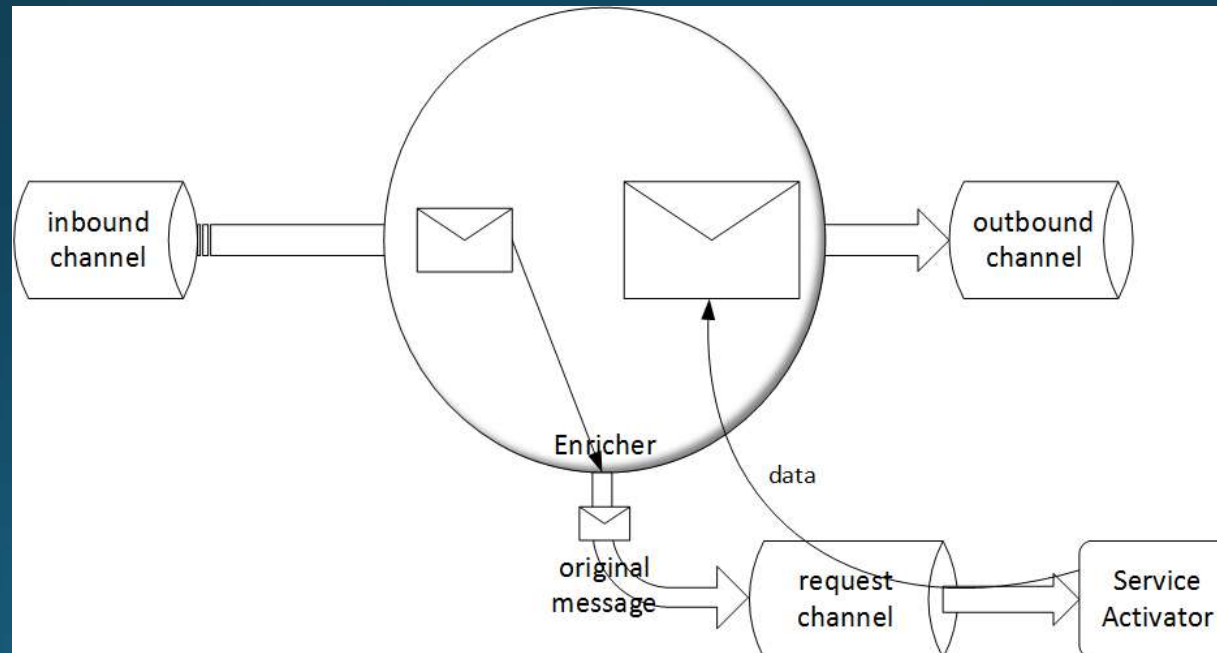
- Payload enrichers, like that below, add data to the message payload, which in this example is about setting a object property to a new Date

```
<int:enricher id="orderEnricher" input-channel="inChannel" output-channel="outChannel">  
  <int:property name="shipDate" expression="new java.util.Date()"/>  
</int:enricher>
```



Another Payload Enricher

- Customize the enricher with a service activator bean.
 - Use a request channel to call on the service activator to provide the augmenting data.



You are ready to tackle Lab 6

- In Lab 6, you revisit the XML-to-object shipment order message transformation work from lab 4.
- Your SI app uses a simple payload enricher to update a boolean property.
- The enricher is improved to call on a service activator component (through a request channel) to calculate the total order cost and add the total cost to the payload.

