Camel Training

split & aggregate

Datum: 23.10.2020
Autor: Johannes Riegler

Splitter EIP

New Order → Splitter → Order Item 1, Order Item 2, Order Item 3
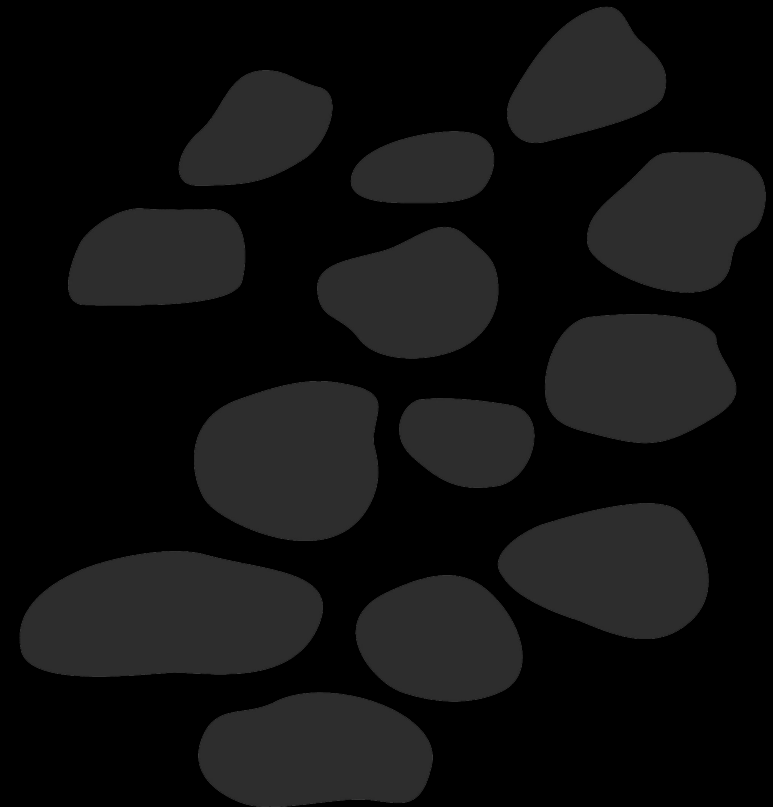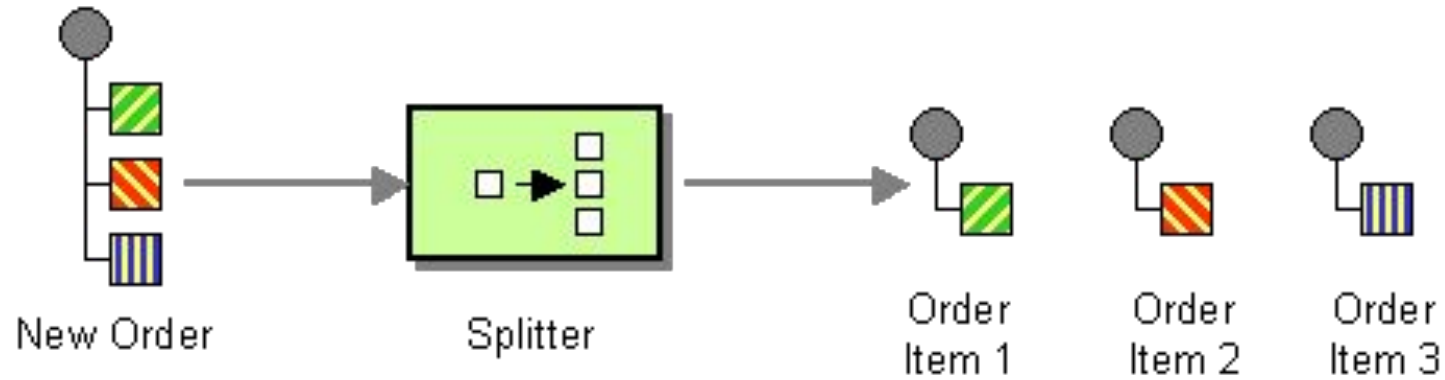
```
from("direct:splitUsingBody" )
    .split(body())
        .to("mock:result");
```

```
from("direct:splitWithResult")
        .split(body()).streaming()
                .to(splitEndpoint)
        .end()
        .to(sumEndpoint);
```

doesn't wait until all submessages are read

sends splitted exchanges

returns (per default) the original exchange

Options

- streaming   splits the messages on demand

- stopOnException   stops further processing on exception

- parallelProcessing   processing submessages in parallel

- timeout

```
.split(body())
  .streaming()
  .parallelProcessing()
  .timeout(5000)
```

Exchange properties

- CamelSplitIndex   counter that increases for each Exchange being split

- CamelSplitSize   total count of submessages

- CamelSplitComplete   true if all submessages split

```
.log( "${header.CamelSplitIndex}/${header.CamelSplitSize}/${header.CamelSplitComplete}")
```

What is splittable ?

- Out of the box

  - `.split(body(String.class).tokenize("\n"))`

  - XPath, XQuery, SQL or one of the Scripting Languages `.split(xpath("//foo/bar"))`

  - Collection (e.g body or headers)

- Write custom Splitter

```
from("direct:message")
  .split().method("mySplitterBean", "spltMessage")
    .to("mock:result");
```
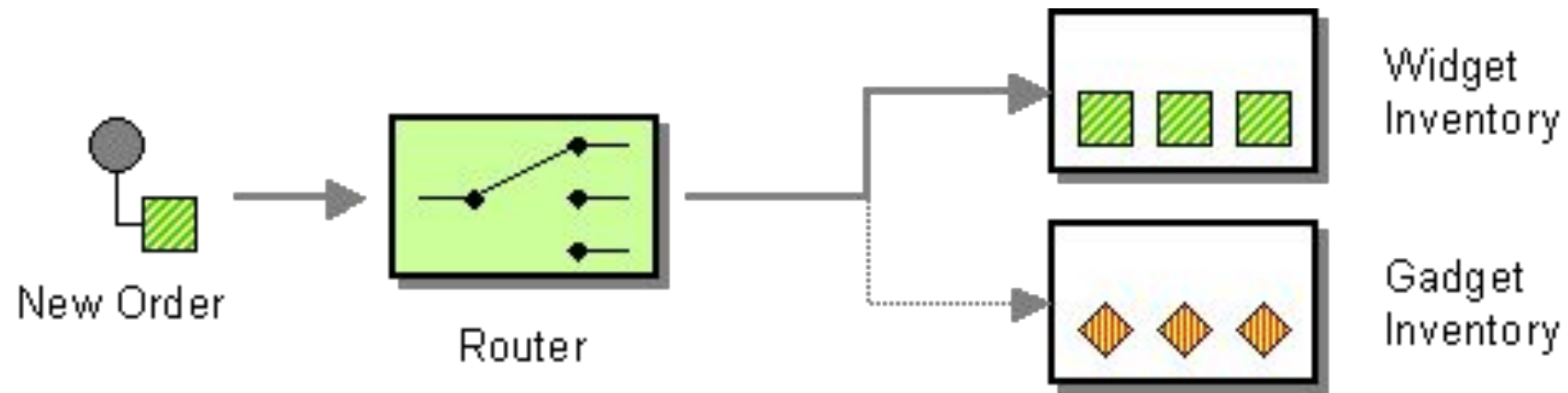
```
from("direct:message")
  .split().method(new OrderSplitter())
    .to("mock:result");
```

```java
public class OrderSplitter {

  public static List<OrderToProducer> splitOrder(Order order) {
    List<OrderToProducer> result = new ArrayList<>();
      for (OrderItem item : order.getItems()) {
        result.add(new OrderToProducer(item, order.getPartnerId()));
      }
      return result;
  }
}
```

Example: best_08_splitter

Choice

when also known as "if"

"Simple" language

expects Predicate object

```java
public void configure() {
 from("direct:a")
    .choice()
     .when(simple("${header.foo} == 'bar'"))
       .to("direct:b")
    .when(simple("${header.foo} == 'cheese'"))
       .to("direct:c")
    .otherwise()
       .to("direct:d")
    .end();
}
```
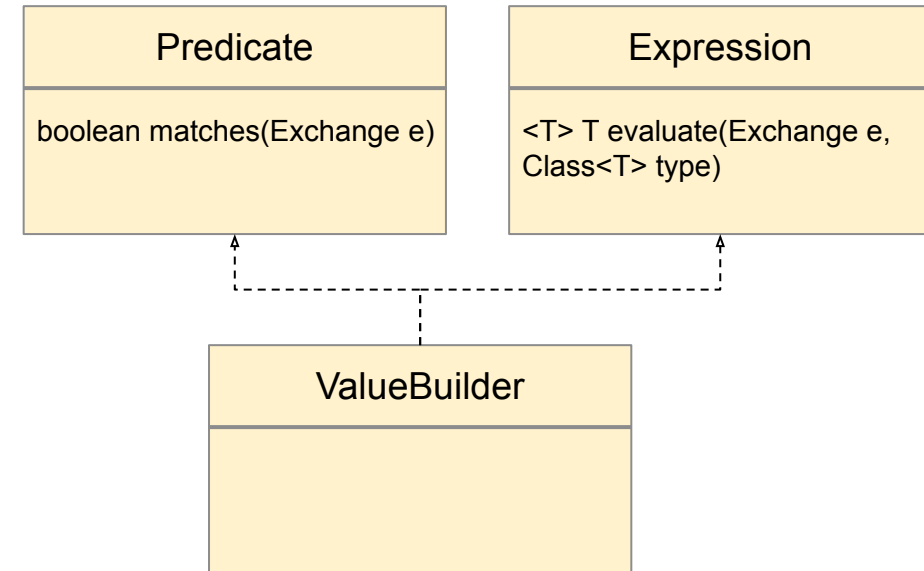
Predicate
boolean matches(Exchange e)

Simple Language

- One of currently 26 [Expression Languages](#)

- Used in other components, like log `.log("Start with body: ${body}")`

- Access it with `simple()` in RouteBuilders

`public ValueBuilder simple(String value)`

Predicate
boolean matches(Exchange e)

| Expression |
|---|
| <T> T evaluate(Exchange e, Class<T> type) |

ValueBuilder

9

Simple Language Examples

"Sending ${body} to file"	Writes message body as string
""CamelSplitIndex: ${header.CamelSplitIndex}"	Writes exchange header with name "CamelSplitIndex"
"Body is ${body.class}"	Writes the class of the Body
simple("${body.code} == 1")	Invokes getCode() on the body-object of an exchange and returns true if it equals 1.
"Code is ${body.getCode()}"	Invokes getCode() on the body and writes the value
simple("${bean:idGenerator.nextId}")	Looks up a registered bean with name "idGenerator" and invokes the method nextId().

See Variables in Simple-Language and Apache OGNL

Example: best_09_choice

Aggregate

New Order   Splitter   Order Item 1   Order Item 2   Order Item 3   Aggregator   Inventory Order
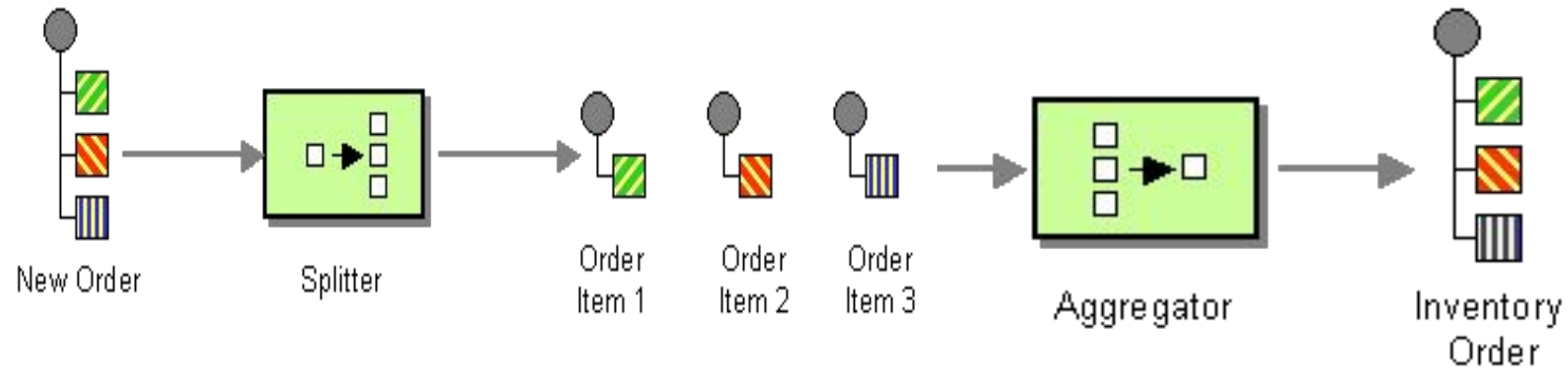
my AggregationStrategy

```java
class StringAggregationStrategy implements AggregationStrategy {

    public Exchange aggregate(Exchange oldExchange, Exchange newExchange) {
        if (oldExchange == null) {
            return newExchange;
        }

        String oldBody = oldExchange.getIn().getBody(String.class);
        String newBody = newExchange.getIn().getBody(String.class);
        oldExchange.getIn().setBody(oldBody + "+" + newBody);
        return oldExchange;
    }
}
```

Split with Aggregation

```
.split(body(), AggregationStrategies.groupedBody())
    .process(exchange -> {
    final ClientEntitySet camelTrainingSet =
exchange.getProperty("CamelTraining",ClientEntitySet.class);
    exchange.getIn().setBody(
        new ExpandedCamelTrainingSet(exchange.getIn().getBody(List.class), camelTrainingSet));
  })
.end()
.to("direct:camelTrainingAggregated");
```