gepardec

Validation and Filters

Why?

// Use an expression or predicates to validate the content of a message.

// Ensure that messages are valid before attempting to process them

How?

// You can use the validate DSL with all kind of Predicates

// Validate evaluates the Predicate and if it is false a `PredicateValidationException` is thrown.

// If it is true message processing continues.

Code/Syntax

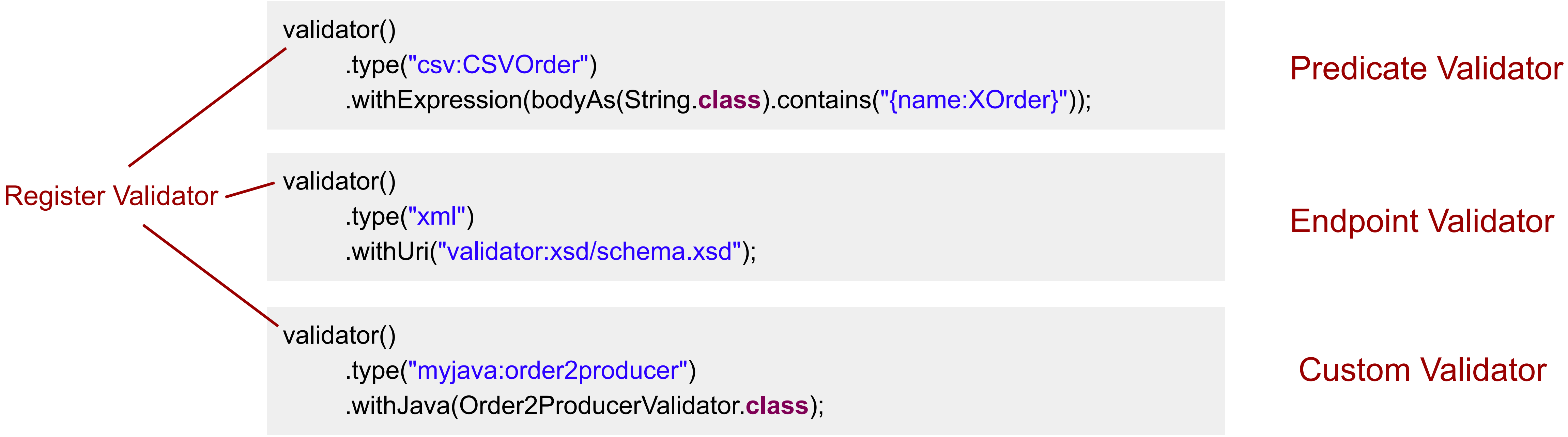
```
from("...")
   .validate(bodyAs(String.class).regex("^\\w{10}\\,\\d{2}\\,\\w{24}$"))
   .to("...");
```

```
from("...")
   .validate(header("bar").isGreaterThan(100))
   .to("...");
```

```
from("...")
   .validate(simple("${in.header.bar} == 100"))
   .to("...");
```

Validator

// Declarative validation of the message according to the declared Input Type and/or Output Type



Register Validator

```
validator()
        .type("csv:CSVOrder")
        .withExpression(bodyAs(String.class).contains("{name:XOrder}"));
```

Predicate Validator

```
validator()
        .type("xml")
        .withUri("validator:xsd/schema.xsd");
```

Endpoint Validator

```
validator()
        .type("myjava:order2producer")
        .withJava(Order2ProducerValidator.class);
```

Custom Validator

```
 from("direct:validate")
   .inputTypeWithValidate("myjava:order2producer")
```

Exception Handling

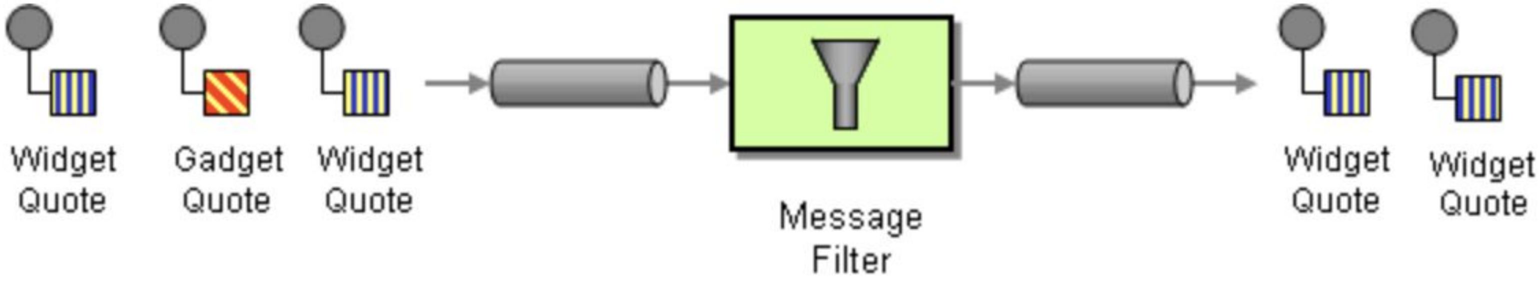
// Try-Catch-Finally

```
from("direct:start")
    .doTry()
        .process(new ProcessorFail())
        .to("mock:result")
    .doCatch(IOException.class, IllegalStateException.class)
        .to("mock:catch")
    .doFinally()
        .to("mock:finally")
    .end();
```

// Use onException inside a RouteBuilder

```
onException(FileNotFoundException.class)
    .handled(true)
    .to("log:nofile");
```

Filter



// Filter evaluates a Predicate and if it is false it ignores the message

```
from("...")
  .filter(header("bar").isGreaterThan(100))
  .to("...");
```

```
from("...")
  .filter(simple("${in.header.bar} == 100"))
  .to("...");
```

Exercise

// best_13_1_filter