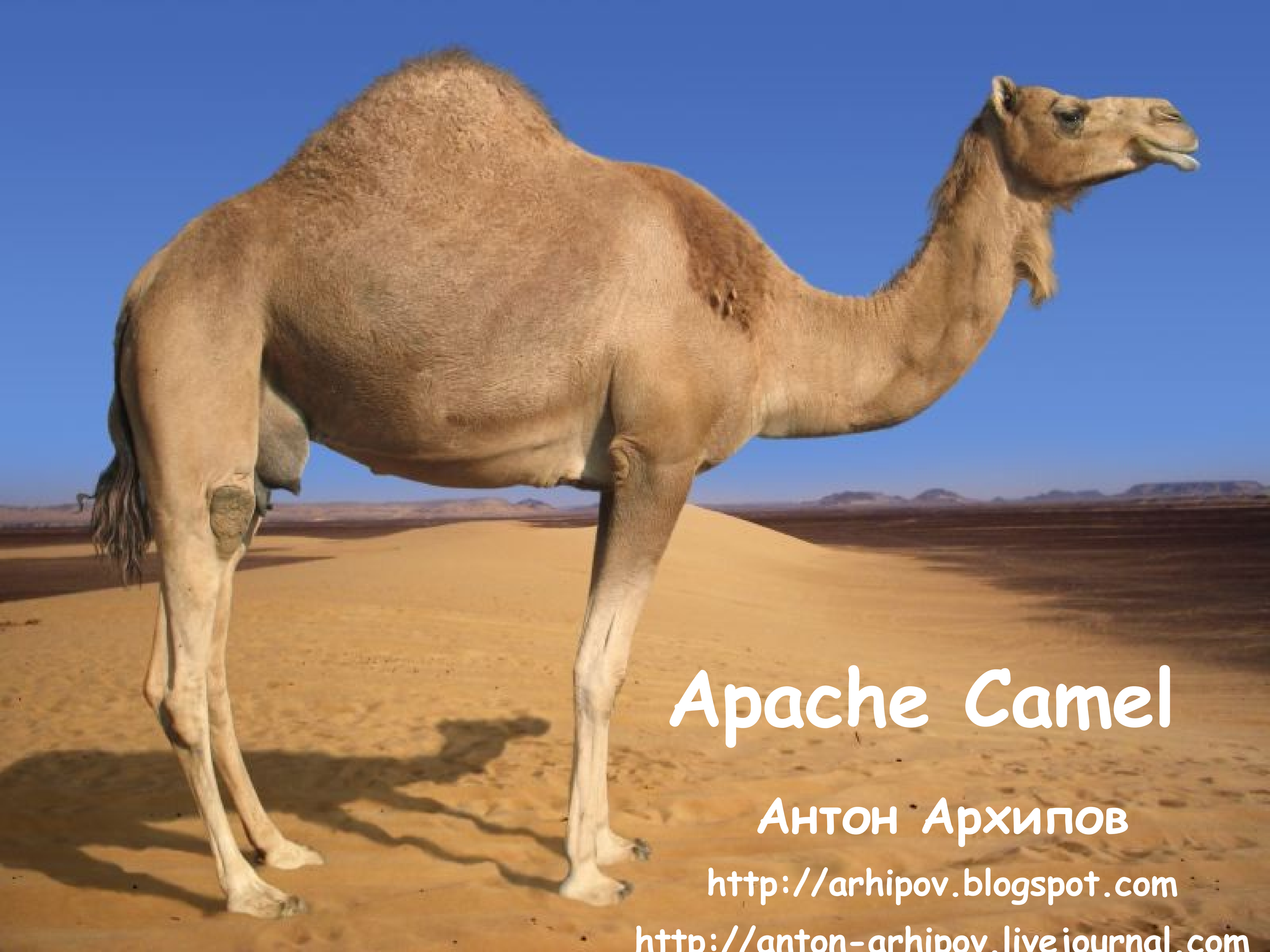
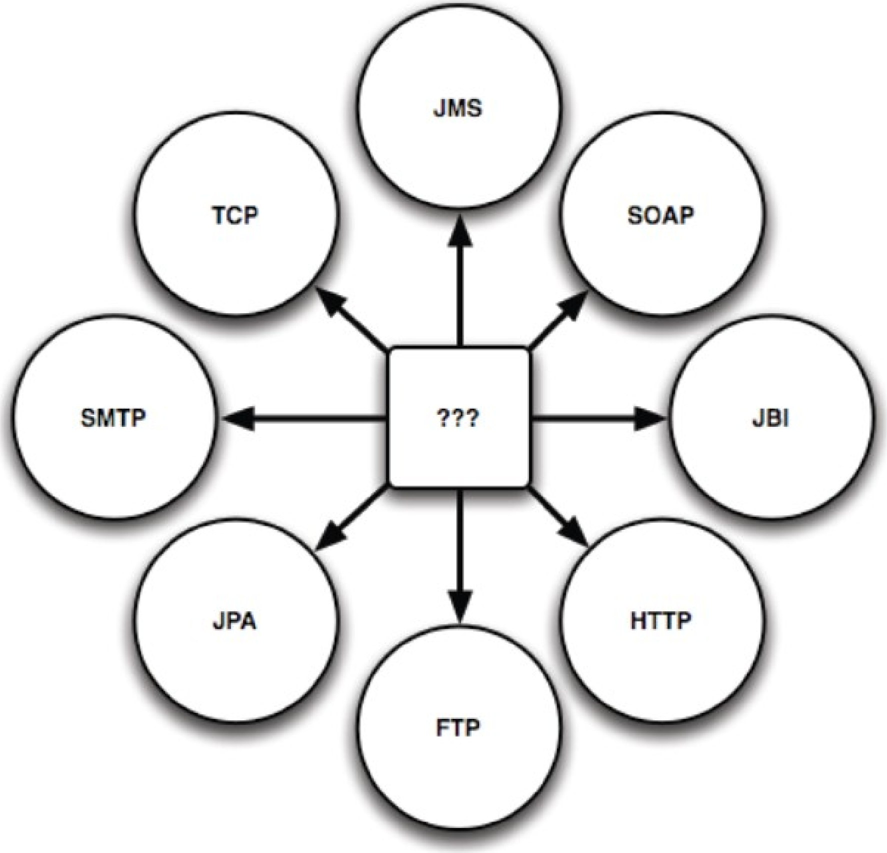


Apache Camel

Антон Архипов

http://arhipov.blogspot.com

http://anton-arhipov.livejournal.com



???

JMS

SOAP

JBI

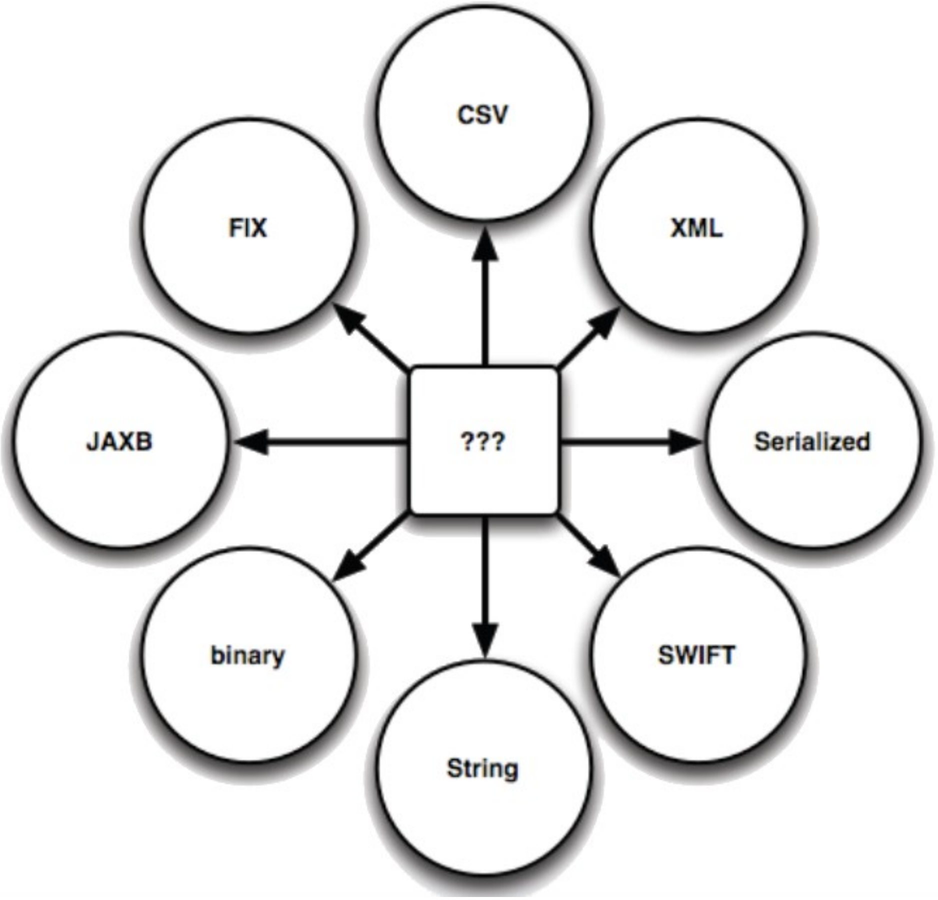
HTTP

FTP

JPA

SMTP

TCP



???

CSV

XML

Serialized

SWIFT

String

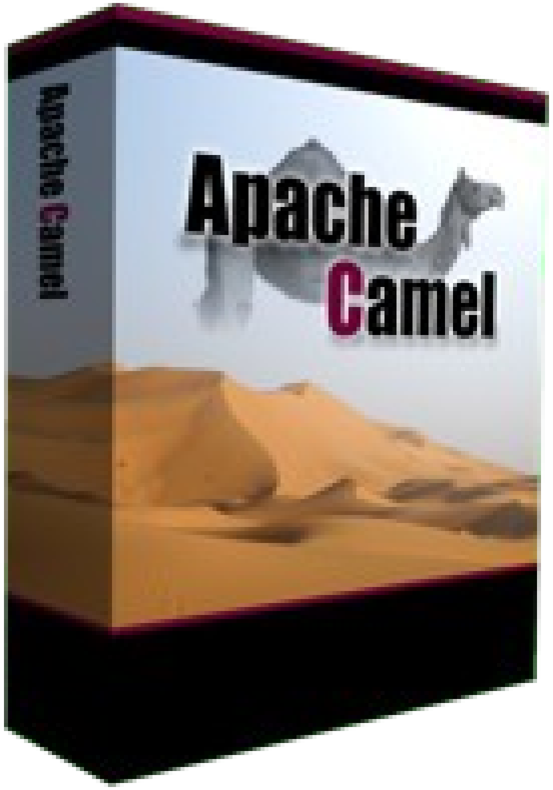
binary

JAXB

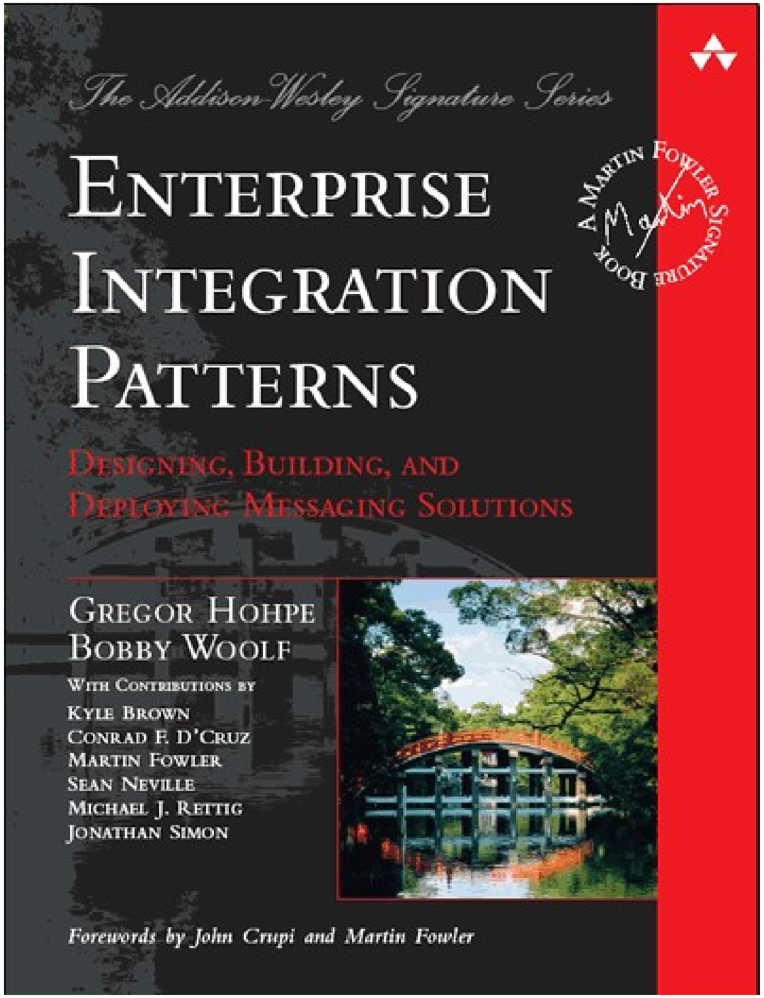
FIX

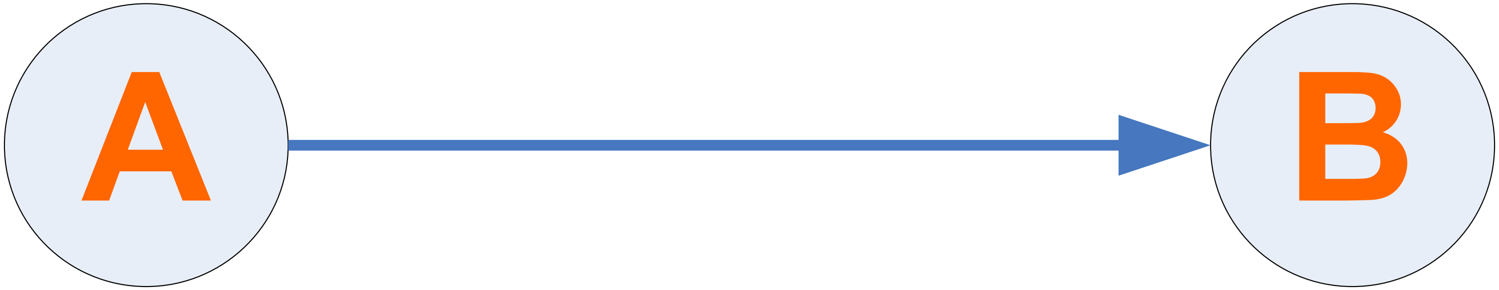
SOA: Spaghetti-Oriented Architecture

What is Apache Camel?

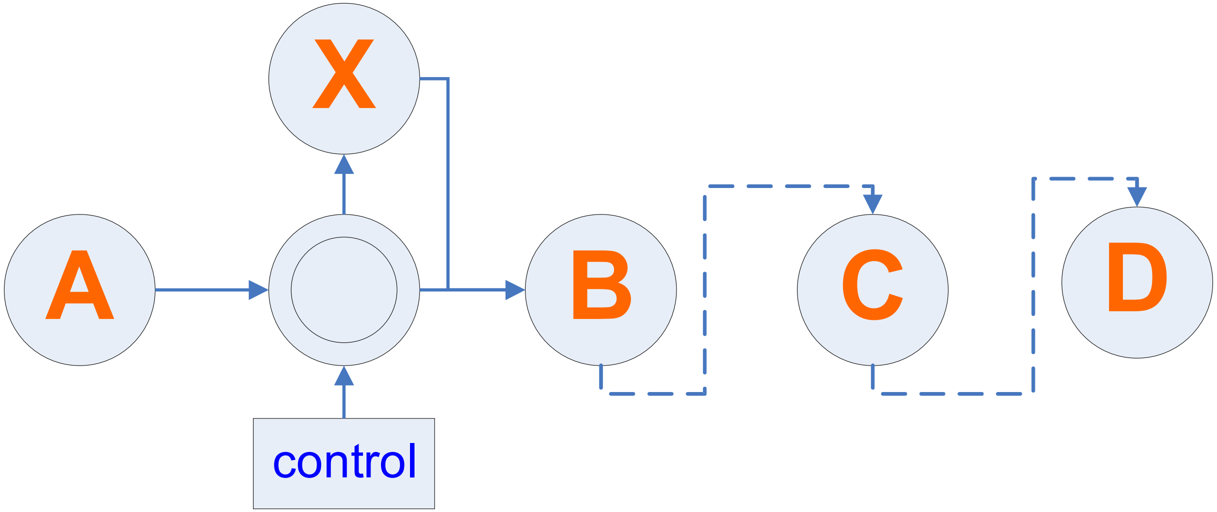


http://camel.apache.org


The Addison-Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
A Martin Fowler Signature Book
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler

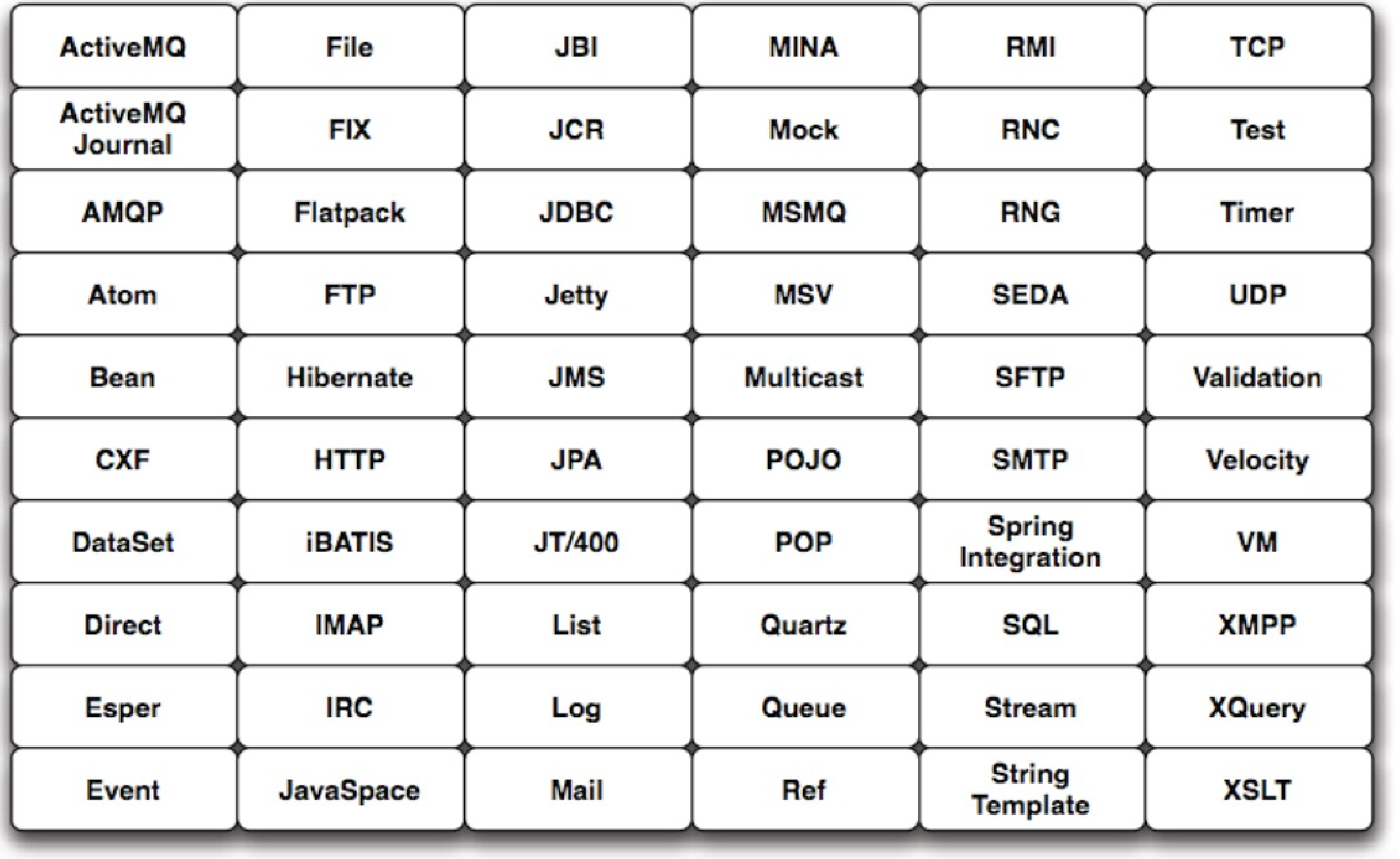


```
from(“a”).to(“b”)
```



```
from("a").choice().when().
   method("controlBean","isOK").
   to("x").end().
pipeline("b", "c", "d");
```

Components



ActiveMQ	File	JBI	MINA	RMI	TCP
ActiveMQ Journal	FIX	JCR	Mock	RNC	Test
AMQP	Flatpack	JDBC	MSMQ	RNG	Timer
Atom	FTP	Jetty	MSV	SEDA	UDP
Bean	Hibernate	JMS	Multicast	SFTP	Validation
CXF	HTTP	JPA	POJO	SMTP	Velocity
DataSet	iBATIS	JT/400	POP	Spring Integration	VM
Direct	IMAP	List	Quartz	SQL	XMPP
Esper	IRC	Log	Queue	Stream	XQuery
Event	JavaSpace	Mail	Ref	String Template	XSLT

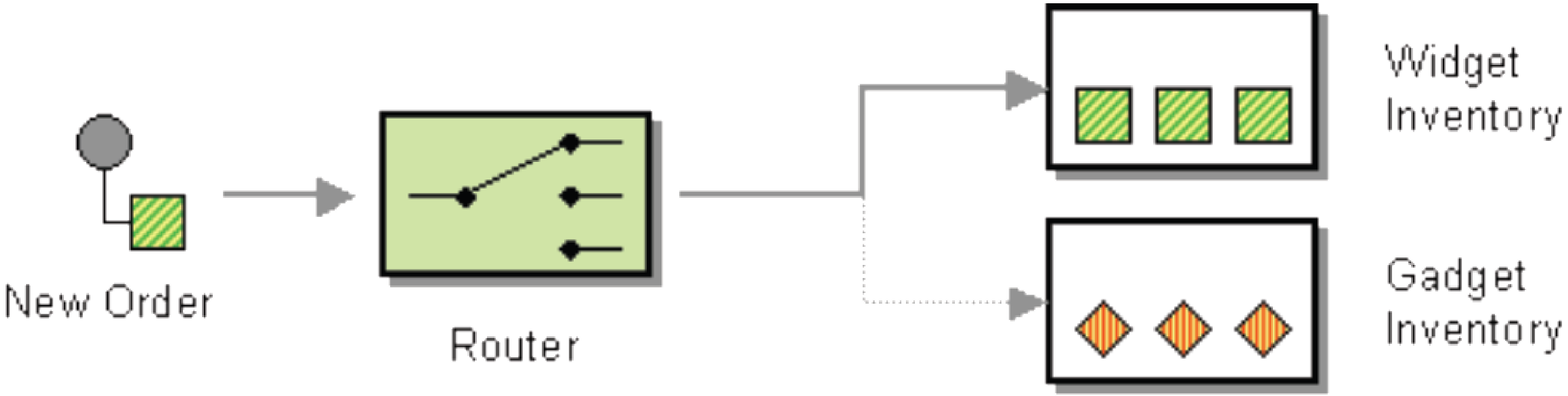
Endpoint URI

```
<component-name>:<component-spec>[?params]
```

```
from(“file:/tmp?consumer.delay=1000”)

from(“bean:myBean?methodName=getOrders”)

from(“quickfix-server:server.cfg?log=#loggingFactoryBean”)
```

Pattern Examples

Content Based Router

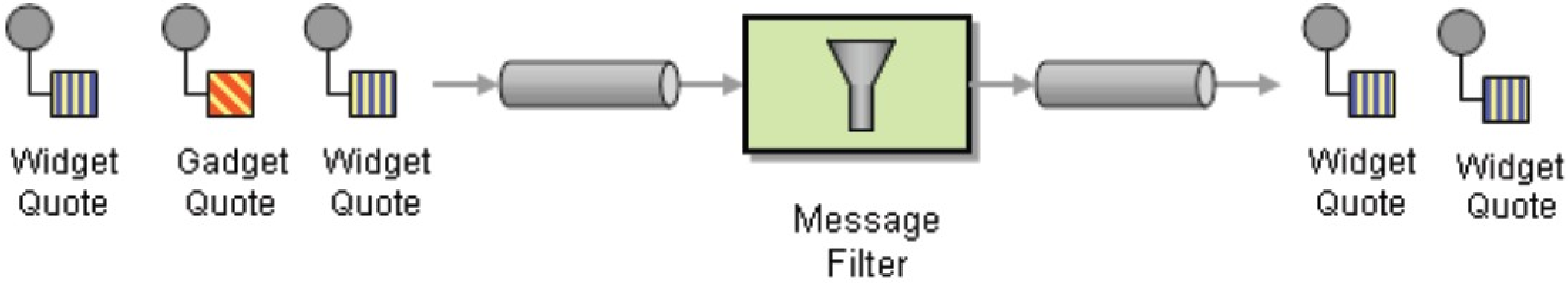


```
  RouteBuilder builder = new RouteBuilder() {
public void configure() {
  from("seda:a").choice().when(header("foo")
    .isEqualTo("bar")).to("seda:b")
      .when(header("foo").isEqualTo("cheese"))
        .to("seda:c").otherwise().to("seda:d");
      }
  };
```

Content Based Router

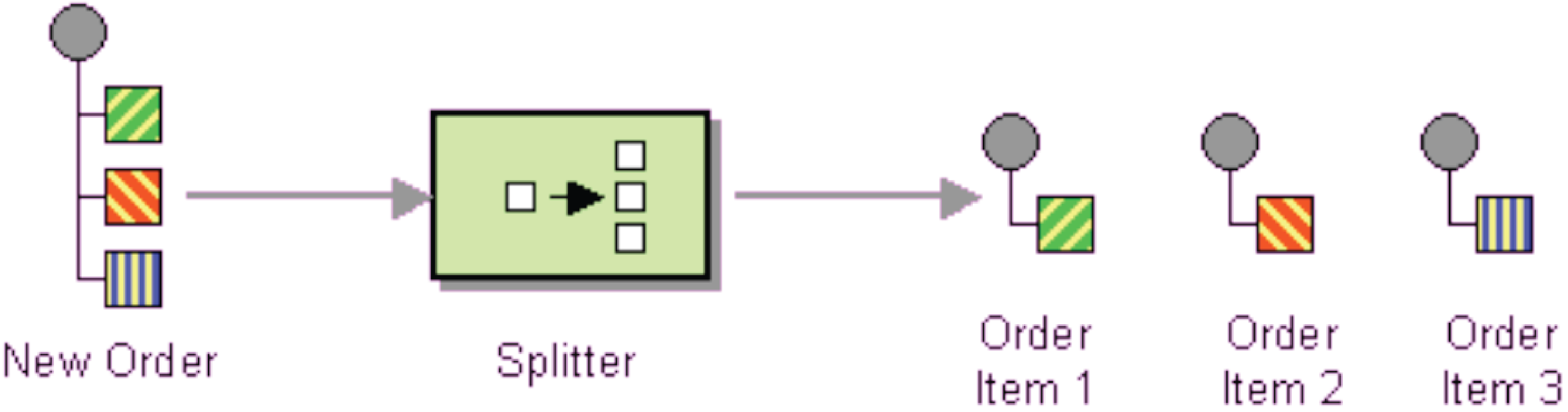
```xml
<camelContext
  xmlns="http://activemq.apache.org/camel/schema/spring">
  <route>
    <from uri="activemq:NewOrders"/>
    <choice>
      <when>
        <xpath>/order/product = 'widget'</xpath>
        <to uri="activemq:Orders.Widgets"/>
      </when>
      <when>
        <xpath>/order/product = 'gadget'</xpath>
        <to uri="activemq:Orders.Gadgets"/>
      </when>
      <otherwise>
        <to uri="activemq:Orders.Bad"/>
      </otherwise>
    </choice>
  </route>
</camelContext>
```

Message Filter



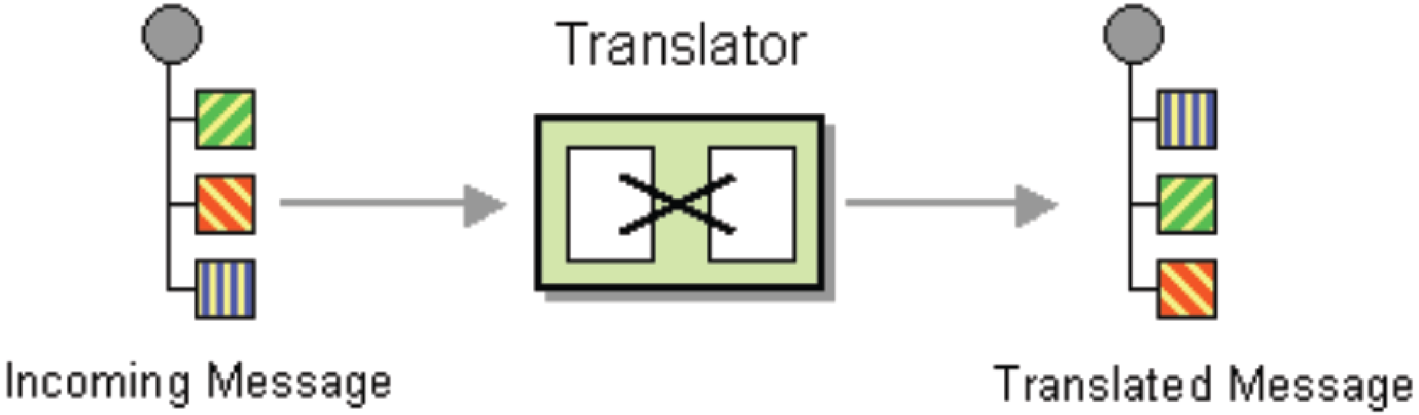
```
public class MyRouteBuilder extends RouteBuilder {
    public void configure() {
        from("activemq:topic:Quotes).
            filter().xpath("/quote/product = ‘widget’").
                to("mqseries:WidgetQuotes");
    }
}
```

Splitter



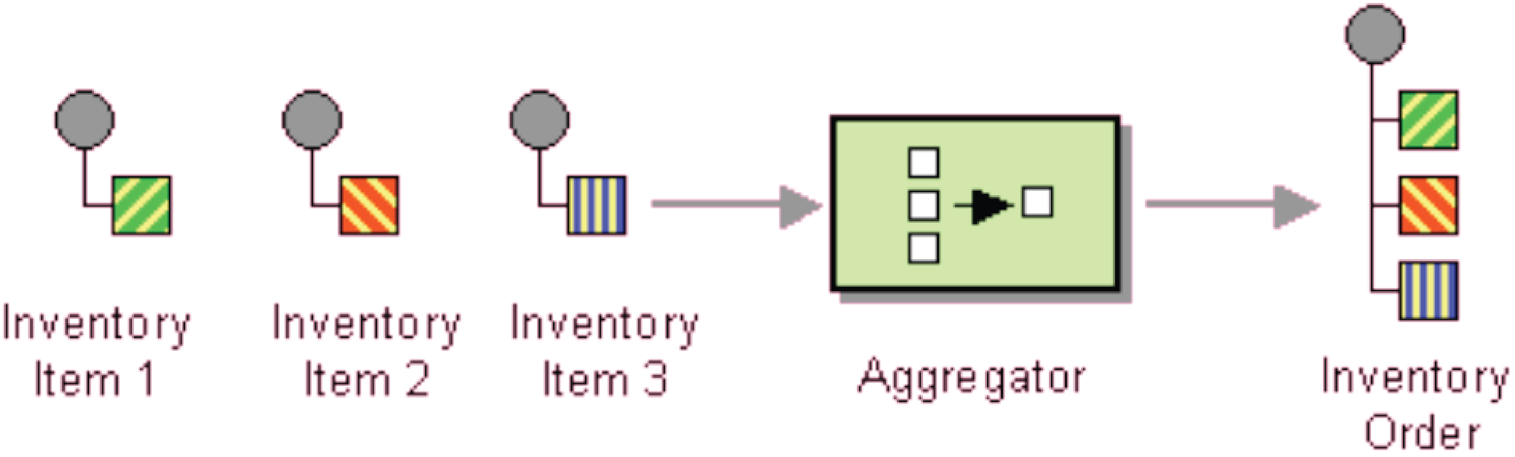
```
public class MyRouteBuilder extends RouteBuilder {
    public void configure() {
      from("file://orders").
        splitter(body().tokenize("\n")).
          to("activemq:Order.Items");
    }
  }
```

Translator



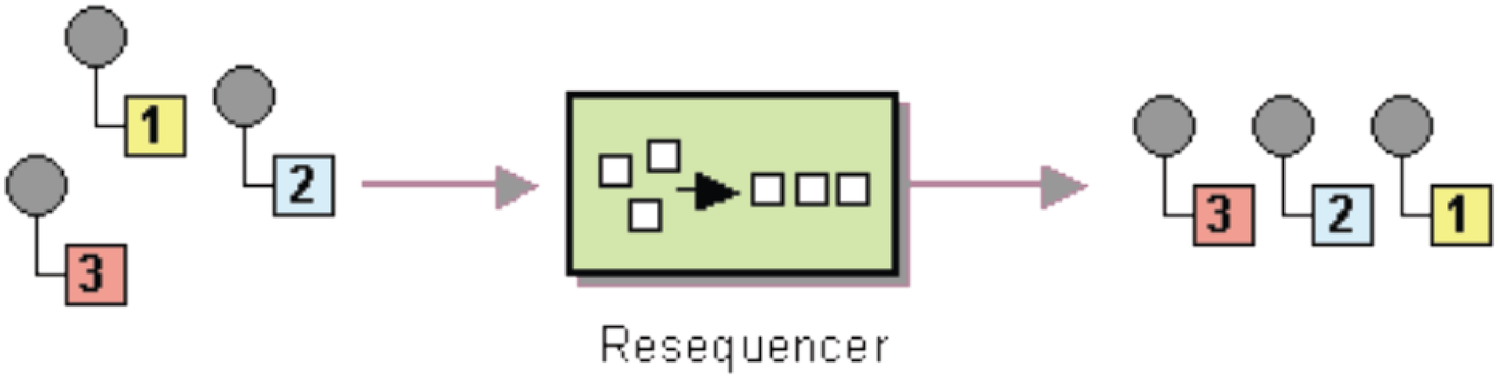
```
public class MyRouteBuilder extends RouteBuilder {
    public void configure() {
      from("file://incoming”).
        to("xslt:com/acme/mytransform.xsl").
          to("http://outgoing.com/foo");
  }
}
```

Aggregator



```
public class MyRouteBuilder extends RouteBuilder {
    public void configure() {
      from("activemq:Inventory.Items").
      aggregator(header("symbol").isEqualTo("IBM").
        to("activemq:Inventory.Order");
  }
}
```

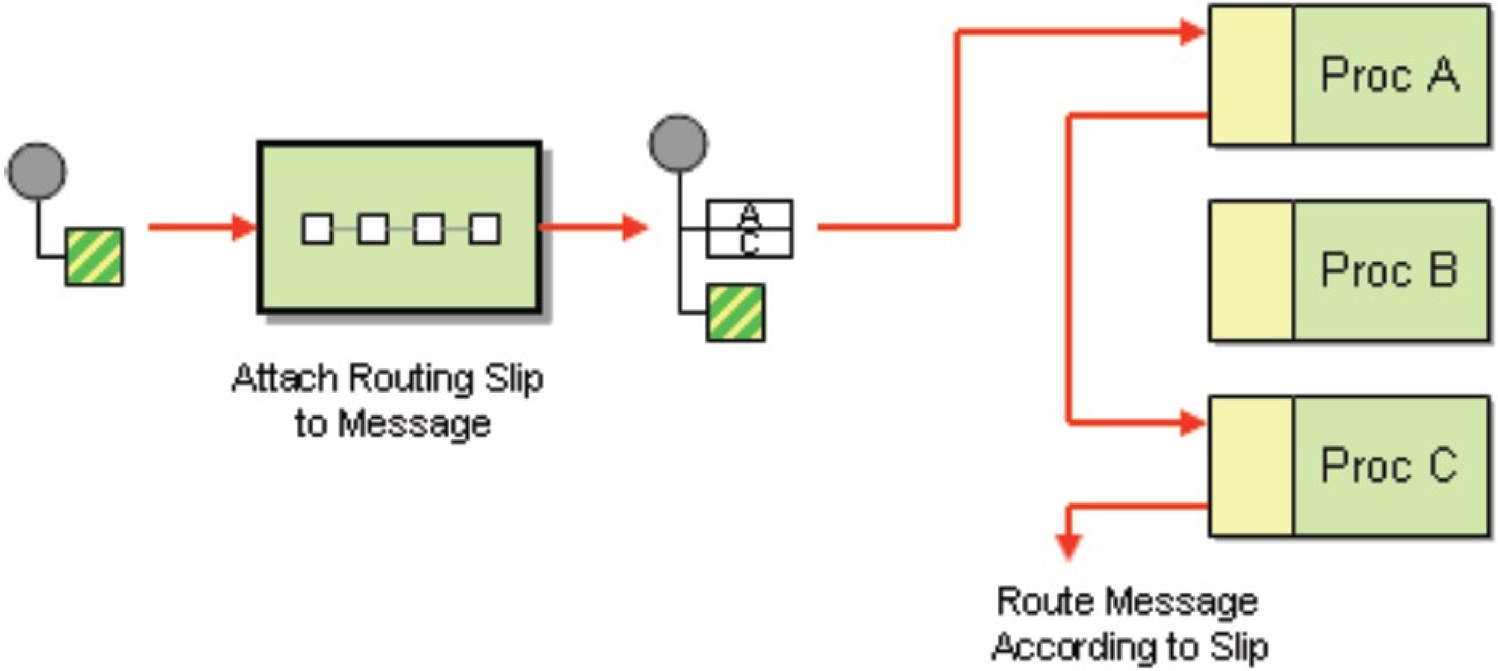
Resequencer



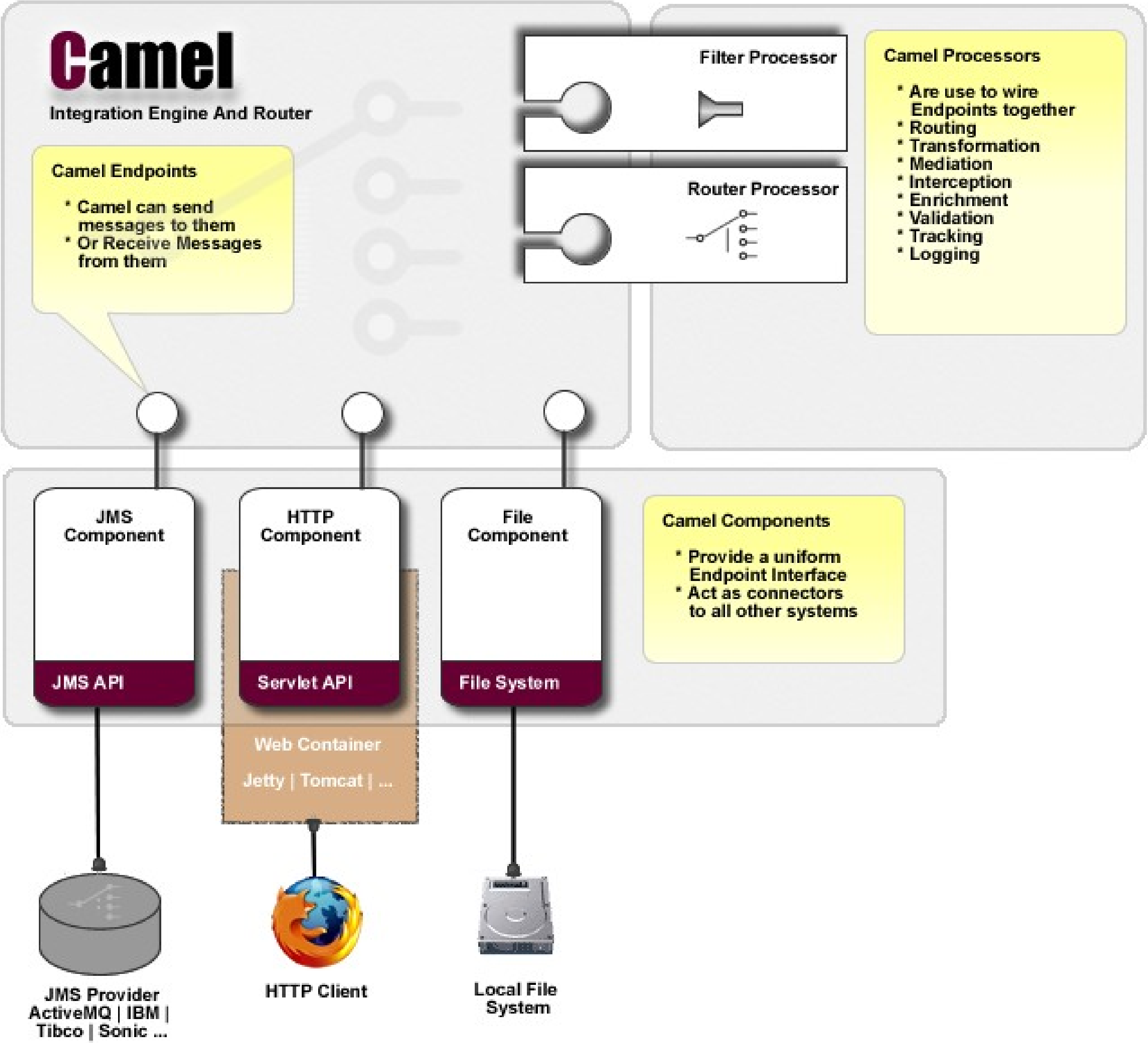
```
public class MyRouteBuilder extends RouteBuilder {
    public void configure() {
      from("direct:a”).
        resequencer(header("customerRank")).
          to("seda:b");
  }
}
```

reliable delivery!

Routing Slip



```
public class MyRouteBuilder extends RouteBuilder {
  public void configure() {
    from("direct:a”).routingSlip();
  }
}
```



Camel Endpoints

- Camel can send messages to them
- Or Receive Messages from them

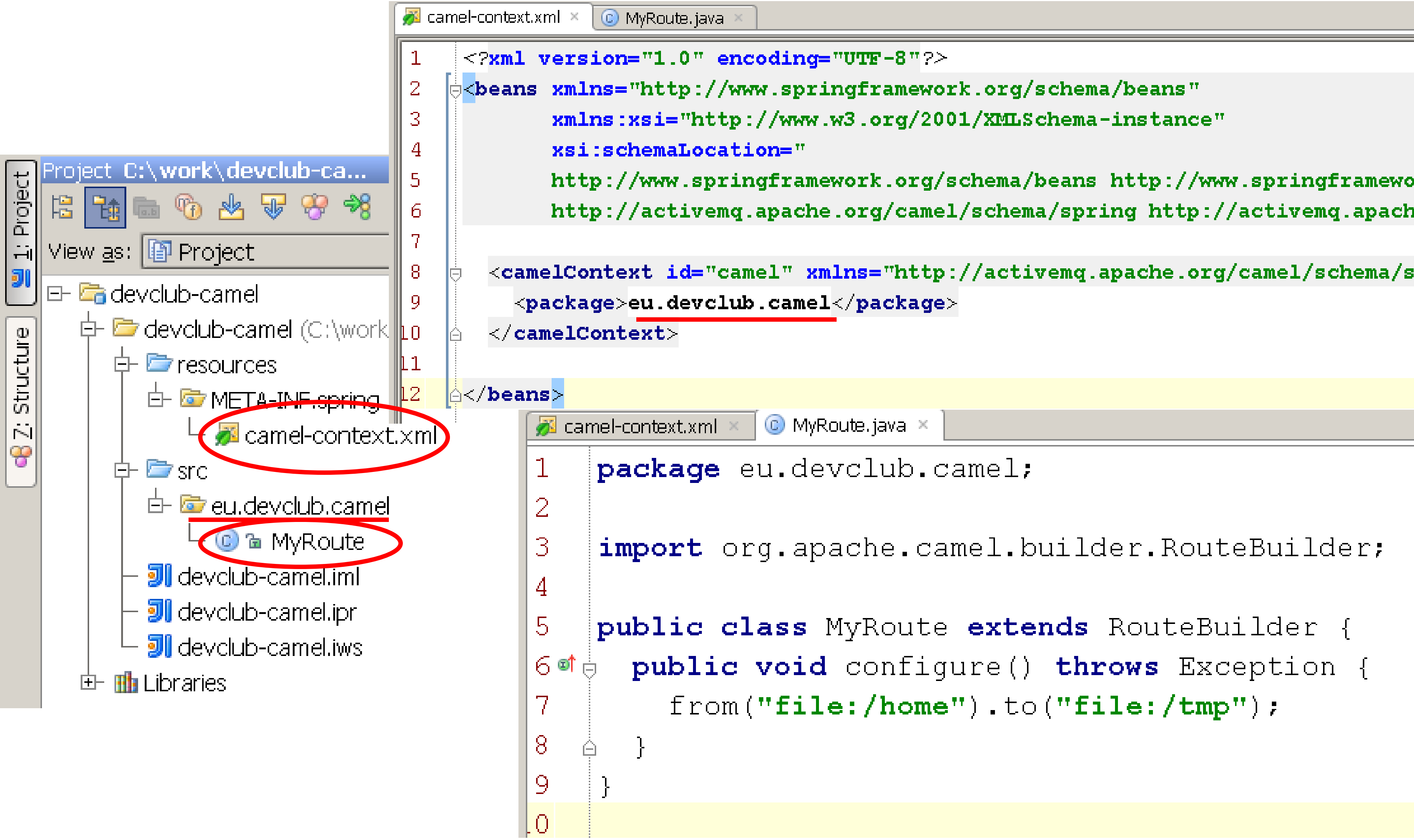
Camel Processors

- Are use to wire Endpoints together
- Routing
- Transformation
- Mediation
- Interception
- Enrichment
- Validation
- Tracking
- Logging

Camel Components

- Provide a uniform Endpoint Interface
- Act as connectors to all other systems

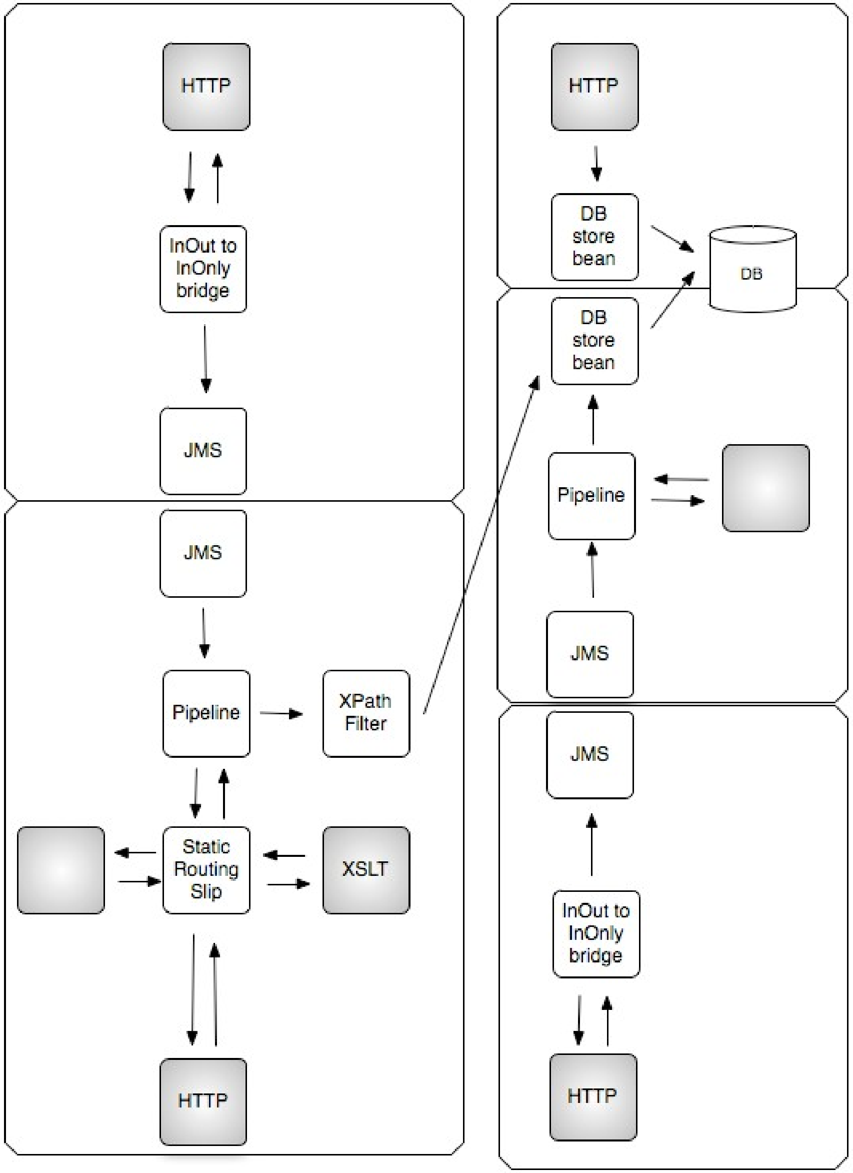
Quick Start



```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
       http://www.springframework.org/schema/beans http://www.springframewo
       http://activemq.apache.org/camel/schema/spring http://activemq.apach

  <camelContext id="camel" xmlns="http://activemq.apache.org/camel/schema/s
    <package>eu.devclub.camel</package>
  </camelContext>

</beans>
```

```
package eu.devclub.camel;

import org.apache.camel.builder.RouteBuilder;

public class MyRoute extends RouteBuilder {
  public void configure() throws Exception {
    from("file:/home").to("file:/tmp");
  }
}
```

java -cp … org.apache.camel.spring.Main



```
from(“http://localhost:8080/requests/”).
    tryBlock().
        to(“activemq:queue:requests”).
        setOutBody(constant(“<ack/>”)).
    handle(Throwable.class).
        setFaultBody(constant(“<nack/>”));

from((“activemq:queue:requests?transacted=true”).
    process(requestTransformer).
    to(“http://host:8080/Request”).
    filter(xpath(“//nack”)).
    process(nackTransformer).
    to(“jdbc:store”);

from(“http://localhost:8080/responses/”).
    tryBlock().
        to(“activemq:queue:responses”).
        setOutBody(constant(“<ack/>”)).
    handle(Throwable.class).
        setFaultBody(constant(“<nack/>”));

from(“activemq:queue:responses?transacted=true”).
    process(responseTransformer).
    to(“jdbc:store”);

from(“http://localhost:8080/pull/”).
    to(“jdbc:load”);
```

Random Features

Error Handling

```
RouteBuilder builder = new RouteBuilder() {
    public void configure() {
        // use logging error handler
        errorHandler(loggingErrorHandler("com.mycompany.foo"));

        // here is our regular route
        from("seda:a").to("seda:b");
    }
};
```

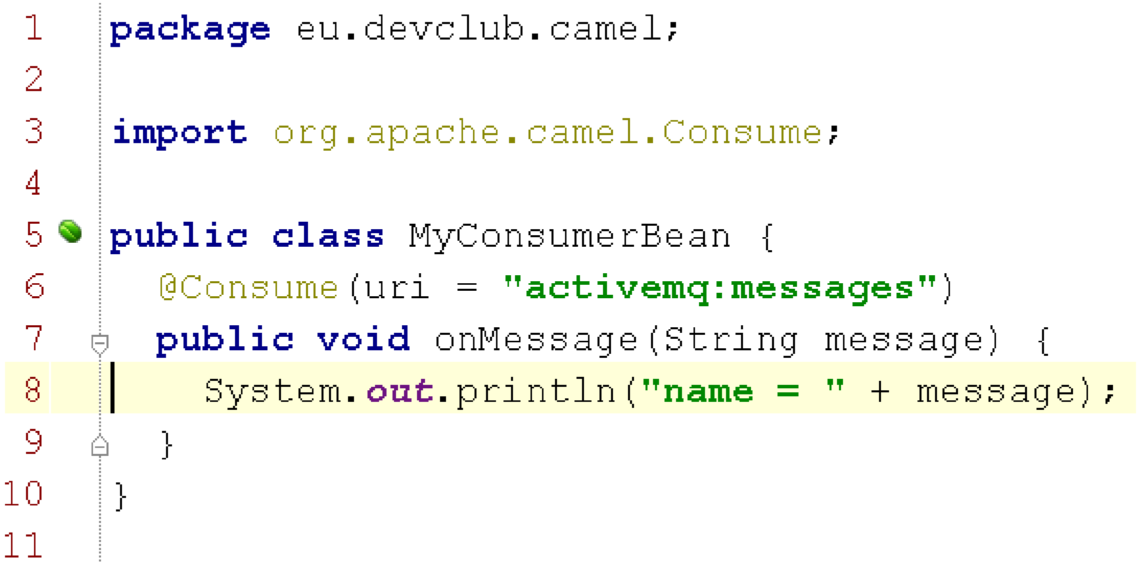
```
RouteBuilder builder = new RouteBuilder() {
    public void configure() {
        // this route is using a nested logging error handler
        from("seda:a")
            // here we configure the logging error handler
            .errorHandler(loggingErrorHandler("com.mycompany.foo"))
            // and we continue with the routing here
            .to("seda:b");

        // this route will use the default error handler (DeadLetterChannel)
        from("seda:b").to("seda:c");
    }
};
```

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();
```

Message-Driven POJO

```xml
<camelContext id="camel" xmlns="http://camel.apache.org/schema/spring">
  <package>eu.devclub.camel</package>
</camelContext>

<bean id="myBean" class="eu.devclub.camel.MyConsumerBean"/>

<bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
  <property name="brokerURL" value="tcp://localhost:61616"/>
</bean>
```

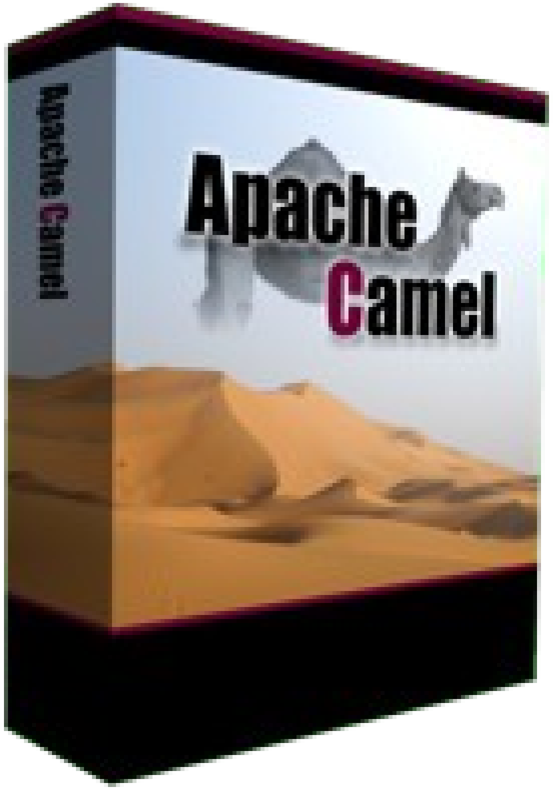


```java
package eu.devclub.camel;

import org.apache.camel.Consume;

public class MyConsumerBean {
  @Consume(uri = "activemq:messages")
  public void onMessage(String message) {
    System.out.println("name = " + message);
  }
}

```

Camel & ActiveMQ

${ACTIVEMQ_HOME}/conf/activemq.xml

```
<camelContext id=“camel”
                xmlns=“http://camel.apache.org/schema/spring”>
   <route>
       <!-- you can add routes here -->
   </route>
</camelContext>
```



http://camel.apache.org