



Take a ride on Apache Camel

James Strachan

FuseSource

A Progress Software Company

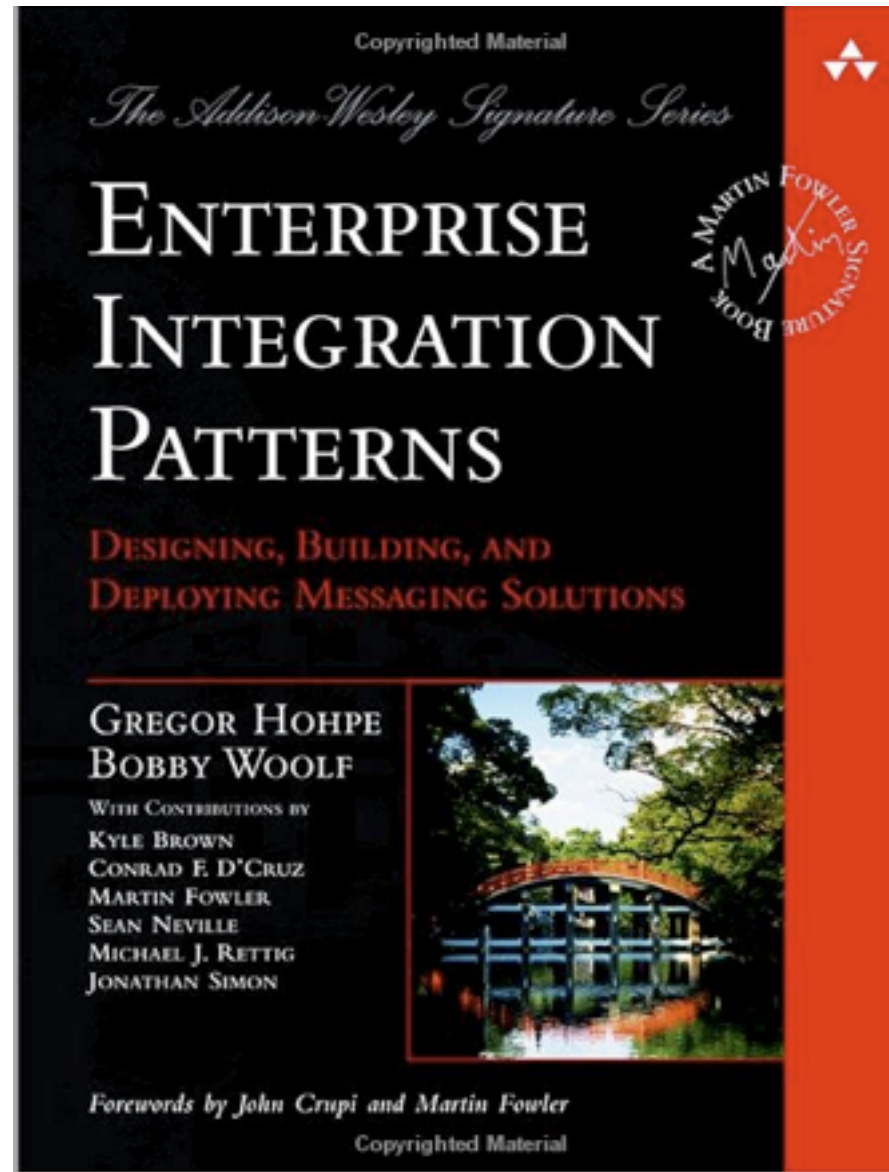
Your speaker today

- James Strachan
 - james@fusesource.com
 - twitter: @jstrachan
 - blog: <http://macstrac.blogspot.com/>
- Software Fellow at FuseSource
 - leaders in open source integration & messaging
 - we provide training, consulting, support, distributions & tools for open source integration software
- Open Source hacker
 - created the Groovy programming language
 - co-founder of
 - Apache ActiveMQ, Camel, ServiceMix
 - Scalate
 - lets not mention Jelly :)

What are Enterprise Integration Patterns?

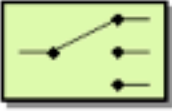







Book by Gregor & Bobby!



A selection of some of the patterns...

Message Routing

	Content Based Router	How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?
	Message Filter	How can a component avoid receiving uninteresting messages?
	Recipient List	How do we route a message to a list of dynamically specified recipients?
	Splitter	How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?
	Aggregator	How do we combine the results of individual, but related messages so that they can be processed as a whole?
	Resequencer	How can we get a stream of related but out-of-sequence messages back into the correct order?
	Throttler	How can I throttle messages to ensure that a specific endpoint does not get overloaded, or we don't exceed an agreed SLA with some external service?
	Delayer	How can I delay the sending of a message?

What is Apache Camel?

<http://camel.apache.org/>

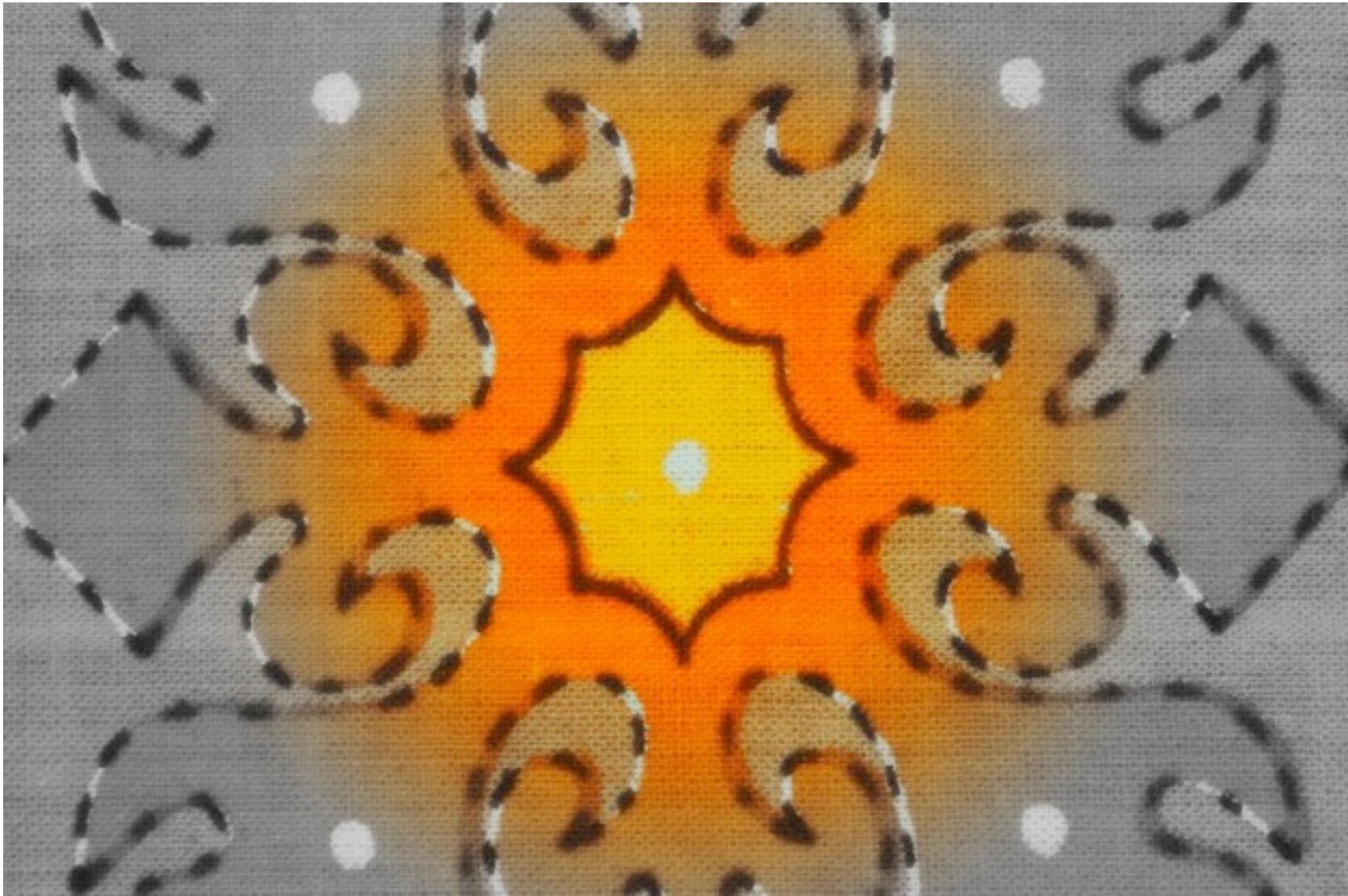


What is Apache Camel?

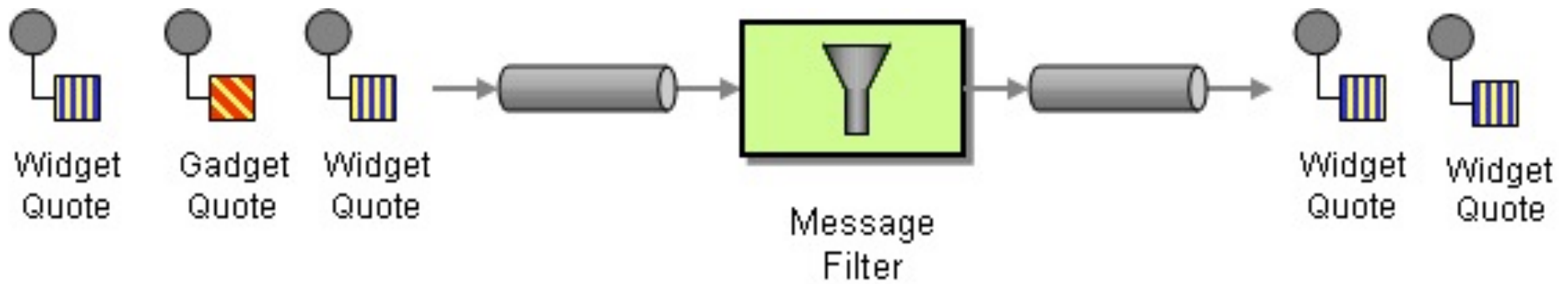
Apache Camel is a Powerful Open Source
Integration Framework
based on known
Enterprise Integration Patterns

<http://camel.apache.org/enterprise-integration-patterns.html>

Lets look at a pattern!



Message Filter



Message Filter : XML

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="activemq:topic:Quotes"/>
    <filter>
      <xpath>/quote/product = 'widget'</xpath>
      <to uri="mqseries:WidgetQuotes"/>
    </filter>
  </route>
</camelContext>
```

Message Filter : Spring XML

```
<?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.springframework.org/schema/beans/spring-beans-2.0.xsd
http://camel.apache.org/schema/spring
http://camel.apache.org/schema/spring/camel-spring.xsd">

  <camelContext xmlns="http://camel.apache.org/schema/spring">
    <route>
      <from uri="activemq:topic:Quotes"/>
      <filter>
        <xpath>/quote/product = 'widget'</xpath>
        <to uri="mqseries:WidgetQuotes"/>
      </filter>
    </route>
  </camelContext>

</beans>
```

Message Filter : XML

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="activemq:topic:Quotes"/>
    <filter>
      <xpath>/quote/product = 'widget'</xpath>
      <to uri="mqseries:WidgetQuotes"/>
    </filter>
  </route>
</camelContext>
```


Expressions & Predicates

15 Expression Languages

BeanShell	Python
EL	Ruby
Groovy	Simple
JavaScript	SpEL
JSR 223	SQL
OGNL	XPath
MVEL	XQuery
PHP	

URIs, Endpoints and Components (80+)

<http://camel.apache.org/components.html>

activemq	cxfrs	flatpack	jasypt
activemq-journal	cxfrs	freemarker	javaspace
amqp	dataset	ftp/ftps/sftp	jbi
atom	db4o	gae	jcr
bean	direct	hdfs	jdbc
bean validation	ejb	hibernate	jetty
browse	esper	hl7	jms
cache	event	http	jmx
cometd	exec	ibatis	jpa
crypto	file	irc	jt/400

80+ components...

language	properties	seda	stream
ldap	quartz	servlet	string-template
mail/imap/pop3	quickfix	sip	test
mina	ref	smooks	timer
mock	restlet	smpp	validation
msv	rmi	snmp	velocity
nagios	rnc	spring-integration	vm
netty	rng	spring-security	xmpp
nmr	rss	spring-ws	xquery
printer	scalate	sql	xslt

Message Filter : XML

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="activemq:topic:Quotes"/>
    <filter>
      <xpath>/quote/product = 'widget'</xpath>
      <to uri="mqseries:WidgetQuotes"/>
    </filter>
  </route>
</camelContext>
```


Message Filter : Java

```
from("activemq:topic:Quotes").  
  filter().xpath("/quote/product = 'widget'").  
  to("mqseries:WidgetQuotes");
```

Message Filter : Java Complete

```
package com.acme.quotes;

import org.apache.camel.builder.RouteBuilder;

public class MyRouteBuilder extends RouteBuilder {

    public void configure() {

        // forward widget quotes to MQSeries
        from("activemq:topic:Quotes").
            filter().xpath("/quote/product = 'widget'").
            to("mqseries:WidgetQuotes");
    }
}
```

Message Filter : Scala

```
"direct:a" when(_.in == "<hello/>") {  
    to("mock:a")  
}
```

IDE support

```
public class FilterRoute extends RouteBuilder {  
    public void configure() throws Exception {  
        Endpoint A = endpoint("activemq:queue: quote");  
        Endpoint B = endpoint("mq:quote");  
        Predicate isWidget = xpath("/quote/product = 'widget'");  
        from(A).fi  
    }  
}
```

		<code>filter(Predicate predicate)</code>	<code>FilterDefinition</code>
		<code>filter()</code>	<code>ExpressionClause<FilterDefinition></code>
		<code>filter(ExpressionDefinition expression)</code>	<code>FilterDefinition</code>
		<code>filter(String language, String expression)</code>	<code>FilterDefinition</code>

Press ^\ to view all accessible classes

Documentation for filter(Predicate)

n: org.apache.camel:camel-core:2.3-SNAPSHOT

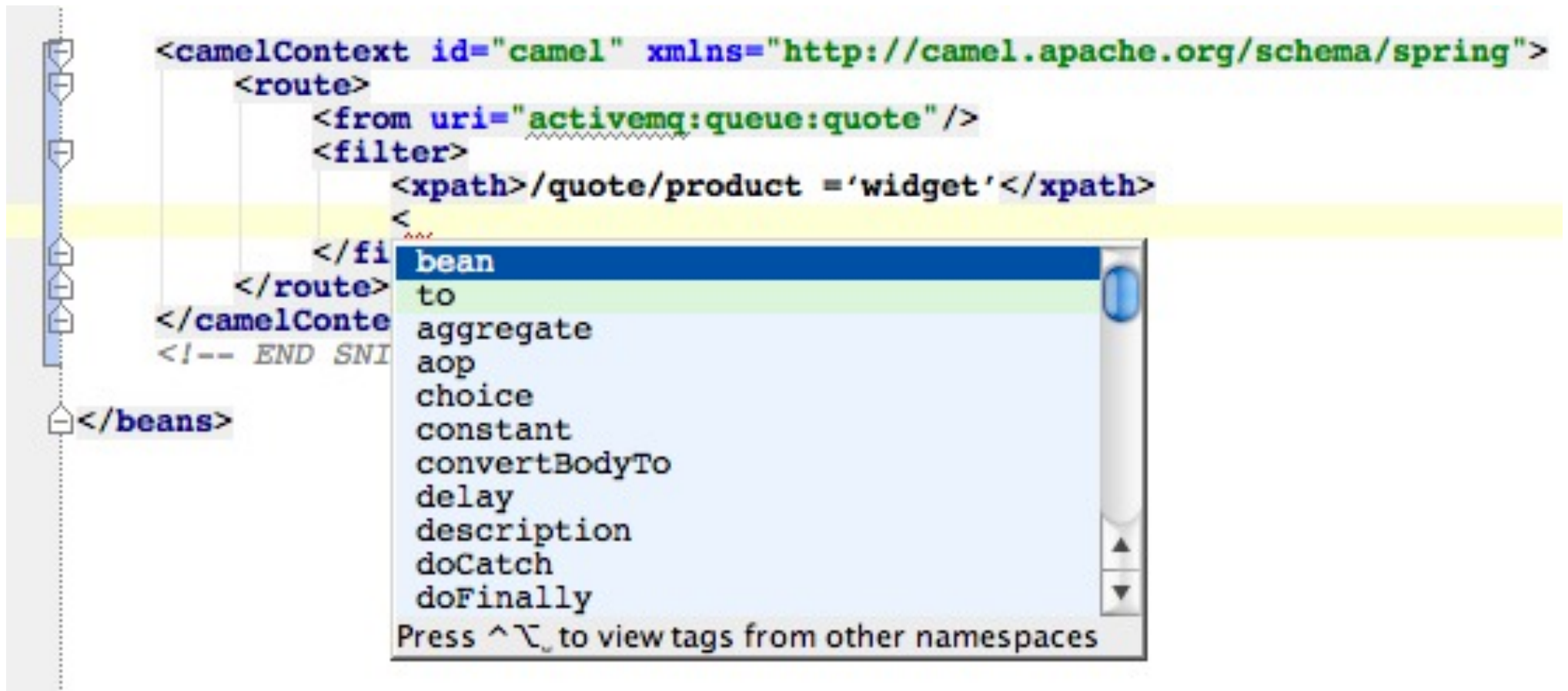
[org.apache.camel.model.ProcessorDefinition](#)

public [org.apache.camel.model.FilterDefinition](#) filter(
[Message Filter EIP](#): Creates a predicate which is applied and only if it is true then the exchange is forwarded to the destination

Parameters:
predicate - predicate to use

Returns:
the builder

IDE support (XML)



Recap - core concepts of Camel

- Enterprise Integration Patterns
- Routing
- Domain Specific Language (DSL)
- Endpoints & URIs
- Predicates & Expressions
- Components (lots of 'em!)
- Test Kit

and much more ...



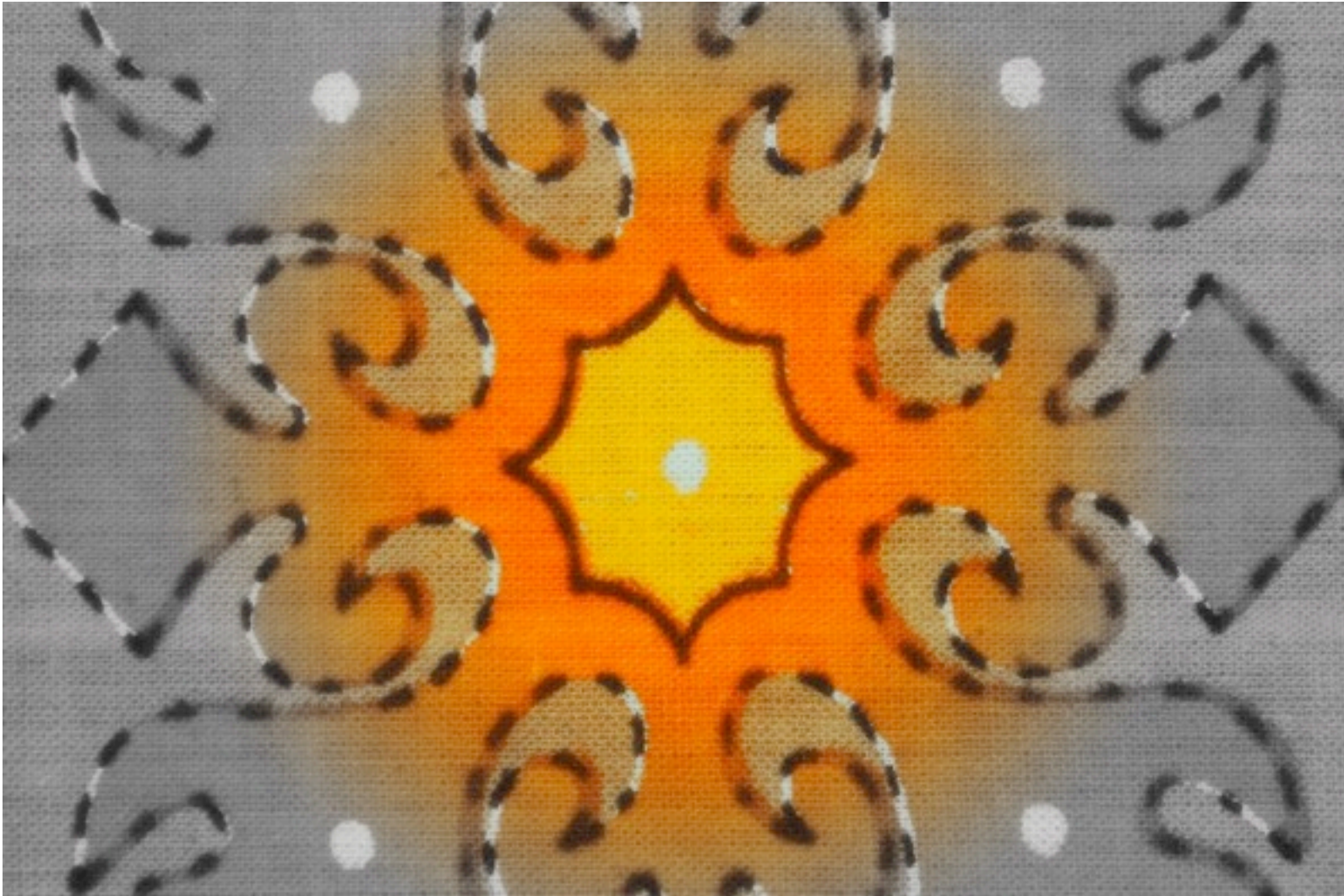
Create CamelContext in Java

```
CamelContext context = new DefaultCamelContext();  
context.addRoutes(new MyRouteBuilder());  
context.start();
```

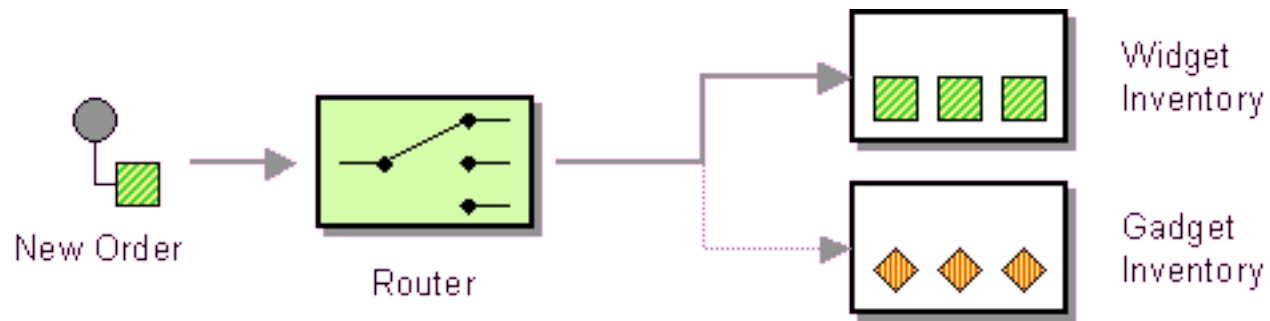
Create CamelContext in Spring

```
<camelContext xmlns="http://camel.apache.org/schema/spring">  
  <package>com.acme.quotes</package>  
</camelContext>
```


More Patterns!



Content Based Router



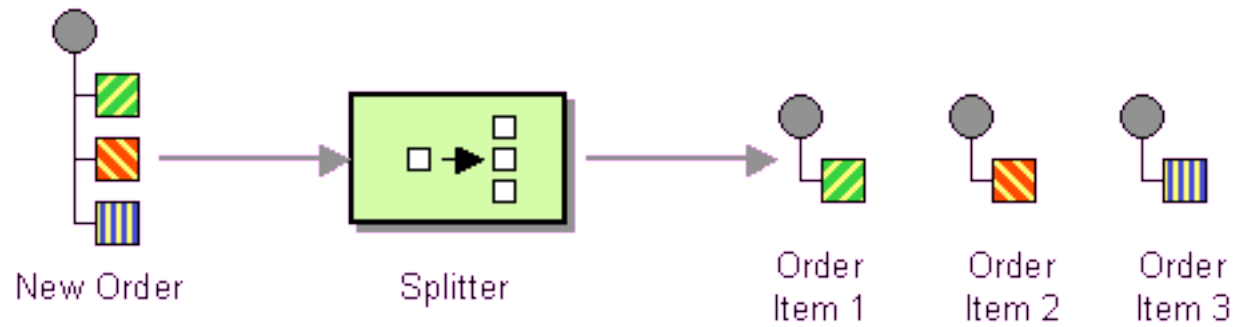
Content Based Router : Scala

```
"activemq:Orders" ==> {  
  when(_.in == "<foo/>") {  
    to ("activemq:BigOrders")  
  }  
  when(_.in == "<bar/>") {  
    to ("activemq:SmallOrders")  
  }  
  otherwise {  
    to ("mqseries:BadStuff")  
  }  
  to ("activemq:AuditQueue")  
}
```

Content Based Router

```
<camelContext xmlns="http://camel.apache.org/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>
```

Splitter



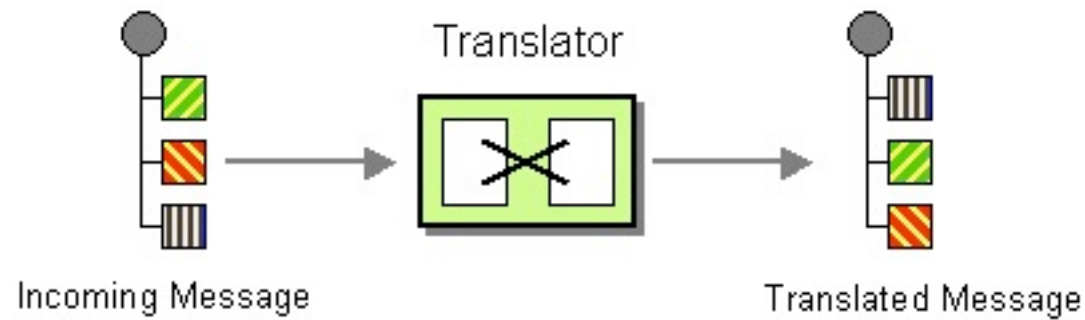
Splitter

```
from("file://orders").  
  splitter(body().tokenize("\n")).  
  to("activemq:Order.Items");
```


Splitter using XQuery

```
from("file://orders").  
  splitter().xquery("/order/items").  
  to("activemq:Order.Items");
```

Message Translator



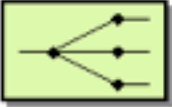


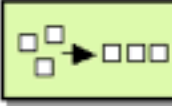


Message Translator

```
from("file://incoming").  
  to("xslt:com/acme/mytransform.xsl").  
  to("http://outgoing.com/foo");
```

Quick recap

Message Routing

	Content Based Router	How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?
	Message Filter	How can a component avoid receiving uninteresting messages?
	Recipient List	How do we route a message to a list of dynamically specified recipients?
	Splitter	How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?
	Aggregator	How do we combine the results of individual, but related messages so that they can be processed as a whole?
	Resequencer	How can we get a stream of related but out-of-sequence messages back into the correct order?
	Throttler	How can I throttle messages to ensure that a specific endpoint does not get overloaded, or we don't exceed an agreed SLA with some external service?
	Delayer	How can I delay the sending of a message?

Beans



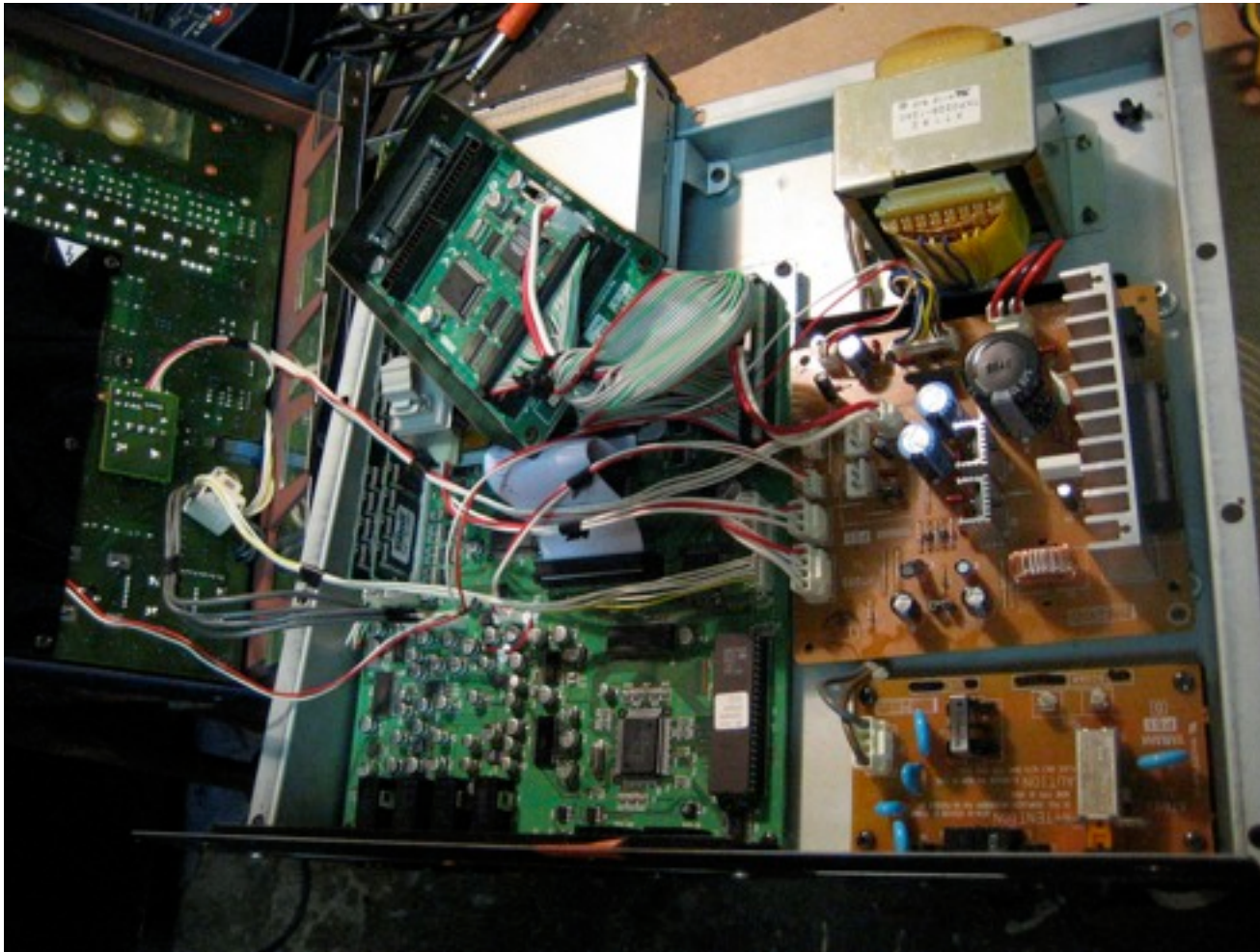
Bean as a Message Translator

```
from("activemq:Incoming").  
    beanRef("myBeanName", "someMethod").  
    to("activemq:Outgoing");
```


Bean

```
public class Foo {  
  
    public String someMethod(String name) {  
        return "Hello " + name;  
    }  
}
```

Type Conversion



Type Conversion

```
package com.acme.foo.converters;

import org.apache.camel.Converter;
import java.io.*;

@Converter
public class IOConverter {

    @Converter
    public static InputStream toInputStream(File file) throws FileNotFoundException {
        return new BufferedInputStream(new FileInputStream(file));
    }
}
```

```
# META-INF/services/org/apache/camel/TypeConverter
com.acme.foo.converters
```

Binding Beans to Camel Endpoints

```
public class Foo {  
  
    @Consume(uri="activemq:cheese")  
    public Object onCheese(String name) {  
        ...  
    }  
}
```

Binding Method Arguments

```
public class Foo {  
  
    public Object onCheese(  
        @XPath( "/foo/bar" ) String name,  
        @Header( "JMSCorrelationID" ) String id) {  
        ...  
    }  
}
```

for more annotations see

<http://camel.apache.org/bean-integration.html>

Sending messages to endpoints

```
public class Foo {  
    @Produce(uri="activemq:foo.bar")  
    ProducerTemplate producer;  
  
    public void doSomething() {  
        if (whatever) {  
            producer.sendBody("<hello>world!</hello>");  
        }  
    }  
}
```


Sending messages to endpoints

```
public interface MyListener {
    String sayHello(String name);
}

public class MyBean {
    @Produce(uri = "activemq:foo")
    protected MyListener producer;

    public void doSomething() {
        // lets send a message
        String response = producer.sayHello("James");
    }
}
```

Dependency Injection

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
  ...
</camelContext>

<bean id="activemq" class="org.apache.camel.component.jms.JmsComponent">
  <property name="connectionFactory">
    <bean class="org.apache.activemq.ActiveMQConnectionFactory">
      <property name="brokerURL" value="vm://localhost?broker.persistent=false"/>
    </bean>
  </property>
</bean>
```

19 Data Formats

bindy	protobuf
castor	serialization
csv	soap
crypto	syslog
dozer	tidy markup
flatpack	xml beans
gzip	xml security
hl7	xstream
jaxb	zip
json	

Data Format

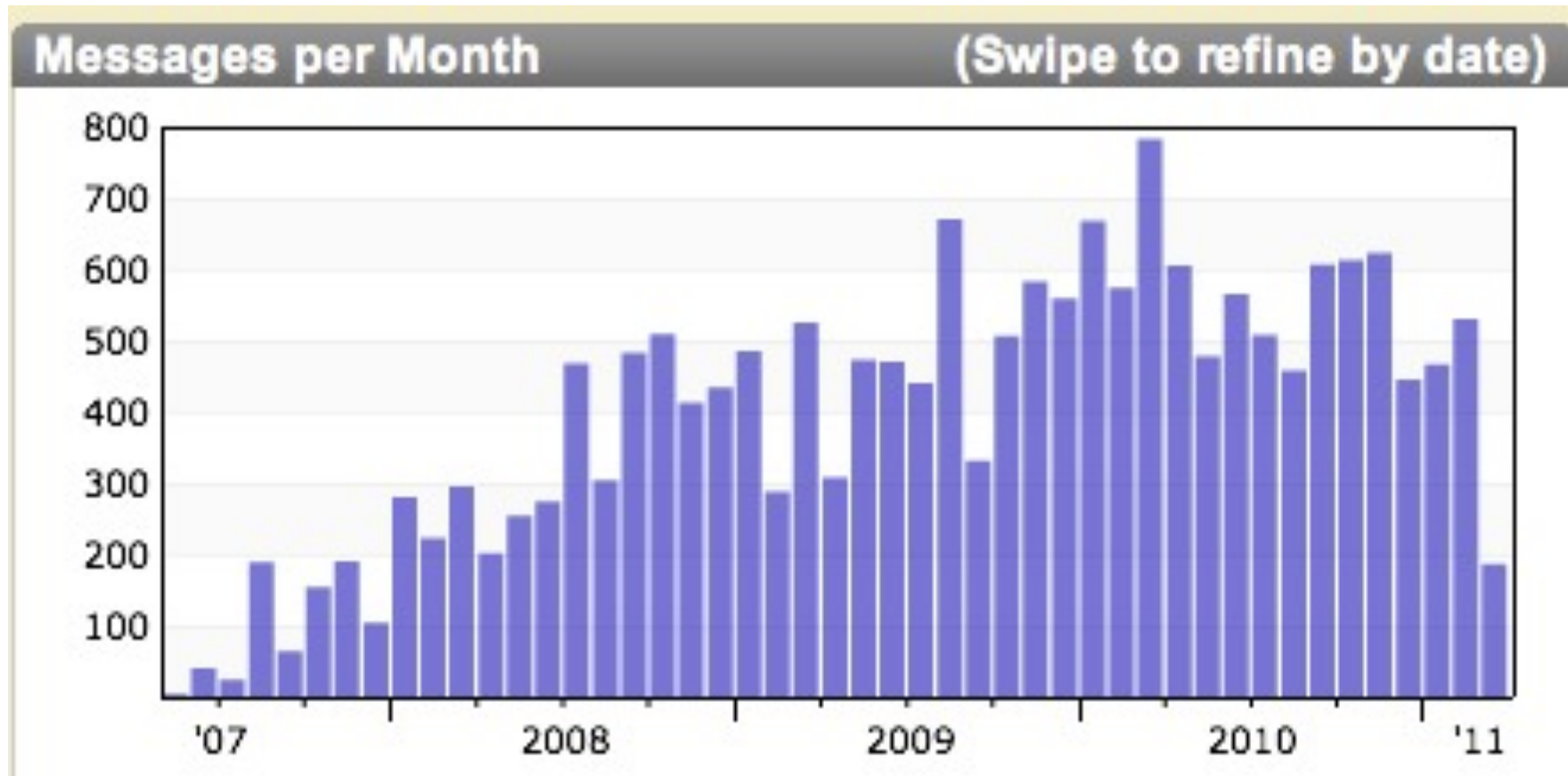
```
from("activemq:QueueWithJavaObjects").  
    marshal().jaxb().  
    to("mqseries:QueueWithXMLMessages");
```

Camel Community



The Camel Community

Camel User Mailing List



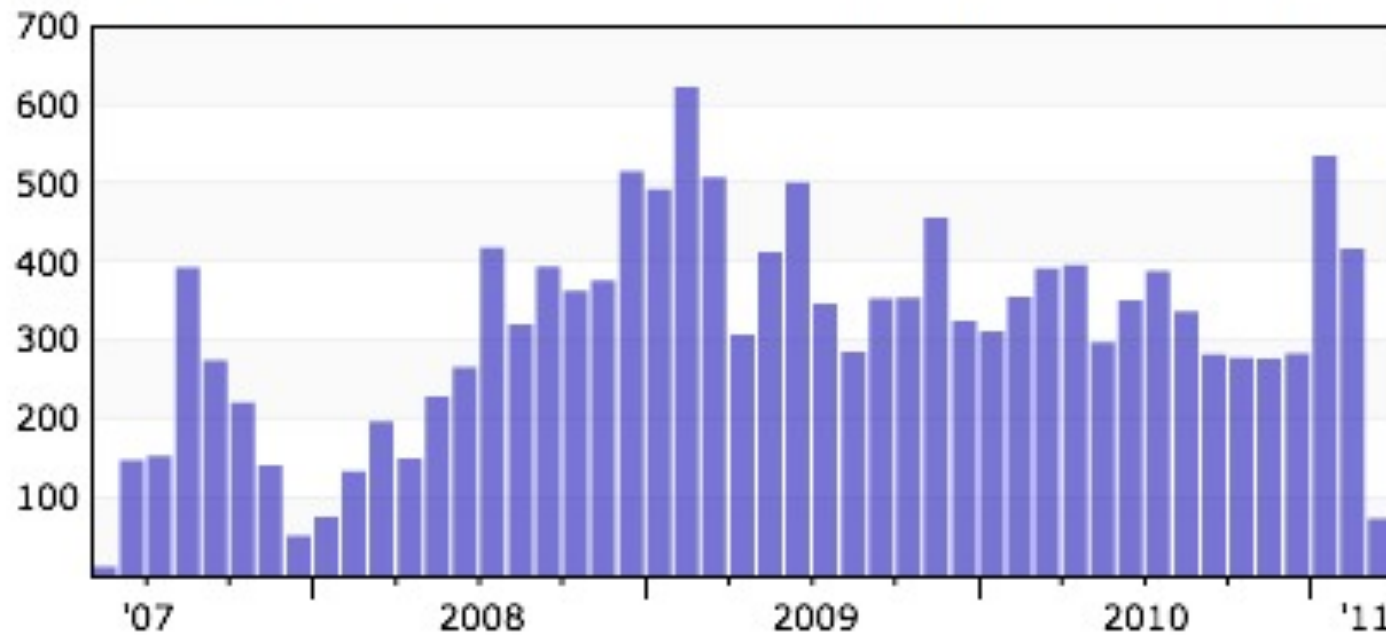
Development Activity

Camel Team

- 25 committers
 - of which 12 has done 50+ commits
- Commits mailing list

Messages per Month

(Swipe to refine by date)



<http://markmail.org/>

Camel Community

- Issue tracker

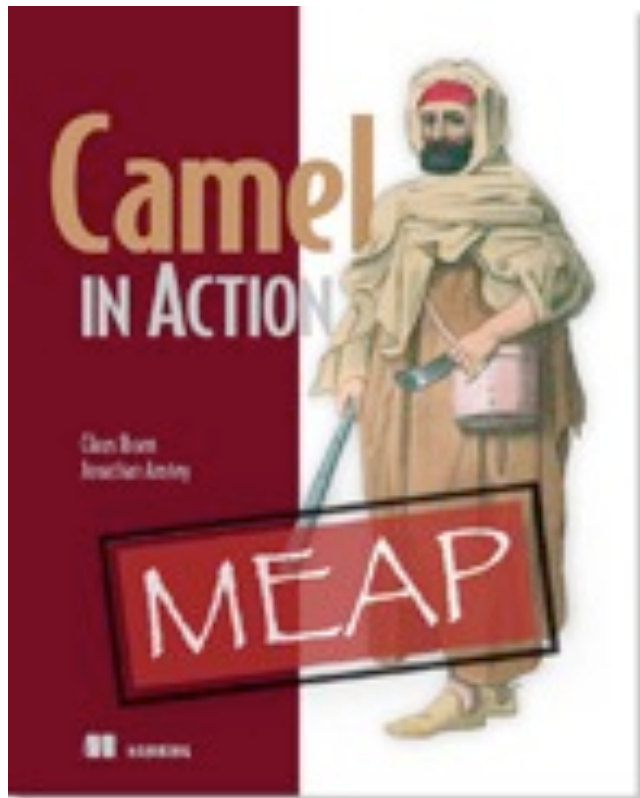
Total	3767
Resolved	3574
Open	193
Bugs	5 (3% of open)

Camel Releases

Release	Date	Tickets
Camel 2.0	Aug 2009	760
Camel 2.1	Dec 2009	303
Camel 2.2	Feb 2010	180
Camel 2.3	May 2010	273
Camel 2.4	July 2010	182
Camel 2.5	Oct 2010	300
Camel 2.6	Jan 2011	297
Camel 2.7	Mar 2011	163

Try the book - its great! :)

- Manning top-15 year to date (2010)



#10



#12

Riding the camel



What should I do next?

- Download Fuse IDE
 - <http://fusesource.com>
- Ride that Camel
 - <http://camel.apache.org/>
- don't get the hump!



Any Questions?



twitter: @jstrachan #fusenews

<http://fusesource.com>