**@Consume**

To consume a message you use the [@Consume](http://camel.apache.org/maven/current/camel-core/apidocs/org/apache/camel/Consume.html) annotation to mark a particular method of a bean as being a **consumer** method. The uri of the annotation defines the Camel [Endpoint](http://camel.apache.org/endpoint.html) to consume from.

e.g. lets invoke the onCheese() method with the String body of the inbound JMS message from [ActiveMQ](http://camel.apache.org/activemq.html) on the cheese queue; this will use the [Type Converter](http://camel.apache.org/type-converter.html) to convert the JMS ObjectMessage or BytesMessage to a String - or just use a TextMessage from JMS

|  |
| --- |
| public class Foo {      @Consume(uri="activemq:cheese")    public void onCheese(String name) {      ...    }  } |

The [Bean Binding](http://camel.apache.org/bean-binding.html) is then used to convert the inbound [Message](http://camel.apache.org/message.html) to the parameter list used to invoke the method .

What this does is basically create a route that looks kinda like this

|  |
| --- |
| from(uri).bean(theBean, "methodName"); |

**Using an explicit route**

If you want to invoke a bean method from many different endpoints or within different complex routes in different circumstances you can just use the normal routing [DSL](http://camel.apache.org/dsl.html) or the [Spring](http://camel.apache.org/spring.html) XML configuration file.

For example

|  |
| --- |
| from(uri).beanRef("myBean", "methodName"); |

which will then look up in the [Registry](http://camel.apache.org/registry.html) and find the bean and invoke the given bean name. (You can omit the method name and have Camel figure out the right method based on the method annotations and body type).

**Use the Bean endpoint**

You can always use the bean endpoint

|  |
| --- |
| from(uri).to("bean:myBean?method=methodName"); |

<http://camel.apache.org/pojo-consuming.html>