Apache Camel

An Intro to Content Based Routing and Wire Tap

# Pre-requisites

Some basic knowledge of what Camel is and installing/using camel and EIP’s

# Content Based Routing

This component routes message based on the content of the message.

Multiple match criteria can be given but only the first match is processed.

Java DSL

import org.apache.camel.builder.RouteBuilder;

public class ContentBasedRouterRouteBuilder extends RouteBuilder {

    @Override

    public void configure() throws Exception {

        from("direct:end")

            .choice()

                .when().simple("${body} contains 'end'")

                    .to("mock:end")

                    .log("end ${body}")

                .otherwise()

                    .to("mock:camel")

                    .log("Camel ${body}")

            .end()

            .log("Message ${body}");

    }

}

Above, I’ve used a ‘direct’ component to pass a message through the choice statement – If the message contains the text ‘end’ – It will pass through to the mock:end route else the message would be passed to the mock:camel route.

Driver Code

public class ContentBasedDriver {

    public static void main(String[] args) throws Exception {

        SimpleRegistry registry = new SimpleRegistry();

        CamelContext context = new DefaultCamelContext(registry);

        context.addComponent("direct", new DirectComponent());

        context.addRoutes(new ContentBasedRouterRouteBuilder());

        context.start();

        ProducerTemplate template = context.createProducerTemplate();

        template.sendBody("direct:start", "This is a test message");

        context.stop();

    }

}

# Wire Tap

This component can help an incoming message get copied over another route, without affecting the main route of the message.

Java DSL

**public** **class** WireTapRouteBuilder **extends** RouteBuilder {

@Override

**public** **void** configure() {

**try** {

from("direct:start")

.wireTap("mock:tapped")

.to("mock:out");

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

As before a direct and a mock component are used and the logs will show a copy of the incoming message being routed. The driver for this code can be adapted from the earlier and is left as an exercise.