Camel Testing Framework using CamelSpringTestSupport

This document describes how to create JUnit camel tests using the class CamelSpringTestSupport

Step-by-step guide

1. Add the camel-test-spring to the pom.xml file

Pom file

```
<dependency>
    <groupId>org.apache.camel</groupId>
    <artifactId>camel-test-spring</artifactId>
    <version>${camel.version}</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
</dependency>

<camel.version>2.15.1</camel.version>
<junit.version>4.11</junit.version>
```

2. Define class which extends CamelSpringTestSupport

Using CamelSpringTestSuport

```
public class TestSupportMockTest extends CamelSpringTestSupport {

 @Override
 protected AbstractApplicationContext createApplicationContext() {
     return new
ClassPathXmlApplicationContext("META-INF/spring/test-support-mock-context.
xml");
 }
}
```

3. Define the route builder in camel context. Camel context file **test-support-mock-context.xml** should be placed in **resource/META-INF/spring** directory

Camel context with route builder reference

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

    <bean id="testSupportMockRouteBuilder"
class="com.myproject.example.routebuilder.TestSupportMockRouteBuilder"/>

    <camelContext id="testSupportMockRouteContext"
xmlns="http://camel.apache.org/schema/spring">
        <routeBuilder ref="testSupportMockRouteBuilder" />
    </camelContext>

</beans>
```

4. Define the route builder

Example route builder file

```
public class TestSupportMockRouteBuilder extends SpringRouteBuilder{

    @Override
    public void configure() throws Exception {

        from("direct:start").routeId("startRouteId")
            .log("Command name = '${header.myCommand}'")
            .choice()
                .when(header("myCommand").isEqualTo("firstCommand"))
                    .log("I am firstCommand")
                    .to("direct:first")
                .when(header("myCommand").isEqualTo("secondCommand"))
                    .log("I am secondCommand")
                    .to("direct:second")
            .end()
            .log("===")
            .to("mock:finish");

        from("direct:first").routeId("firstRouteId")
            .transform(constant("First Value"))
            .log("My body in first = '${body}'")
            .to("mock:first");

        from("direct:second").routeId("secondRouteId")
            .transform(constant("Second Value"))
            .log("My body in second = '${body}'")
            .to("mock:second");
    }
}
```

5. Example tests

Example junit tests with mocks

```
@Test
public void testMockEndpoints() throws Exception {
    //given
    MockEndpoint mockFinish = getMockEndpoint("mock:finish");
    MockEndpoint mockFirst = getMockEndpoint("mock:first");

    mockFinish.expectedMessageCount(3);
    mockFinish.expectedBodiesReceived("Funny World", "First Value", "Second
Value");

    mockFirst.expectedMessageCount(1);
    mockFirst.expectedBodiesReceived("First Value");

    //when
    template.sendBody("direct:start", "Funny World");
    template.sendBodyAndHeader("direct:start", "Hello World", "myCommand",
"firstCommand");
    template.sendBodyAndHeader("direct:start", "Hate World", "myCommand",
"secondCommand");

    //then
    mockFirst.assertIsSatisfied(); //for given mock endpoint
    mockFinish.assertIsSatisfied(); //for given mock endpoint
}
```

6. Mocking all endpoints:

Mocking all endpoints

```
@Test
public void testMockAllEndpoints() throws Exception {
    context.getRouteDefinitions().get(0).adviceWith(context, new
AdviceWithRouteBuilder() {
        @Override
        public void configure() throws Exception {
            // mock all endpoints
            mockEndpoints();
        }
    });

    MockEndpoint mockFinish = getMockEndpoint("mock:finish");
    MockEndpoint mockFirst = getMockEndpoint("mock:first");
    MockEndpoint mockDirectFirst = getMockEndpoint("mock:direct:first");

    mockFinish.expectedMessageCount(3);
    mockFinish.expectedBodiesReceived("Funny World", "First Value", "Second
Value");
    mockFirst.expectedMessageCount(1);
    mockFirst.expectedBodiesReceived("First Value");
    mockDirectFirst.expectedBodiesReceived("Hello World");
    mockDirectFirst.expectedMessageCount(1);

    template.sendBody("direct:start", "Funny World");
    template.sendBodyAndHeader("direct:start", "Hello World", "myCommand",
"firstCommand");
    template.sendBodyAndHeader("direct:start", "Hate World", "myCommand",
"secondCommand");

    assertMockEndpointsSatisfied(); //use this for all mock endpoints

    // additional test to ensure correct endpoints in registry
    assertNotNull(context.hasEndpoint("direct:start"));
    assertNotNull(context.hasEndpoint("direct:first"));
    assertNotNull(context.hasEndpoint("direct:second"));
    assertNotNull(context.hasEndpoint("mock:first"));
    assertNotNull(context.hasEndpoint("mock:second"));
    assertNotNull(context.hasEndpoint("mock:finish"));
    // all the endpoints was mocked
    assertNotNull(context.hasEndpoint("mock:direct:start"));
    assertNotNull(context.hasEndpoint("mock:direct:first"));
    assertNotNull(context.hasEndpoint("mock:direct:second"));
}
```

7. Try these piece of code

Piece of code

```
//given
mockFinish.expectedBodiesReceivedInAnyOrder("First Value", "Second Value",
"Funny World");

//given
context.getRouteDefinitions().get(0).adviceWith(context, new
AdviceWithRouteBuilder() {
    @Override
    public void configure() throws Exception {
        // mock only direct:first
        mockEndpoints("direct:first");
    }
});

//given
mockDirectFirst.message(0).body().contains("World");
mockDirectFirst.message(0).body().startsWith("Hello");
//other API
mockDirectFirst.message(0).body().xxx(...)
...
mockDirectFirst.allMessages().body().contains("World");


//then
assertMockEndpointsSatisfied(); //use this for all mock endpoints


//then
List<Exchange> exchangeList = mockDirectFirst.getExchanges();
assertEquals(3, exchangeList.size());
assertTrue(exchangeList.get(0).getIn().getBody(String.class).contains("Hel
lo World"));
....
assertTrue(exchangeList.get(0).getIn().getHeaders().containsValue("firstCo
mmand"));
```

Github resource with example code

https://github.com/pawmac24/camel-spring-testing

Apache Camel resource:

http://camel.apache.org/testing.html

http://camel.apache.org/spring-testing.html

http://camel.apache.org/mock.html

http://camel.apache.org/advicewith.html