Custom DataFormat

Custom DataFormat

You can use your custom Data Format implementation with Camel. All you have to do is to implement the `DataFormat` interface. For example in the following we will implement a reverse data format as shown below:

Error formatting macro: snippet: java.lang.IndexOutOfBoundsException: Index: 20, Size: 20

And to use it in Java DSL:

Error formatting macro: snippet: java.lang.IndexOutOfBoundsException: Index: 20, Size: 20

Notice we use custom to refer to the Data Format in the Registry. In Java DSL you can also provide the instance directly as shown:

```
from("direct:a")
    .marshal(new MyReverseDataFormat())
    .to("mock:a");
```

And likewise to use it in XML DSL:

Error rendering macro 'code': Invalid value specified for parameter 'java.lang.NullPointerException'

```
<!-- this is our custom data format implementation -->
<bean id="reverse" class="org.apache.camel.impl.RefDataFormatTest$MyReverseDataFormat"/>

<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="direct:a"/>
    <marshal>
      <!-- refer to my custom data format -->
      <custom ref="reverse"/>
    </marshal>
    <to uri="mock:a"/>
  </route>

  <route>
    <from uri="direct:b"/>
    <unmarshal>
      <!-- refer to my custom data format -->
      <custom ref="reverse"/>
    </unmarshal>
    <to uri="mock:b"/>
  </route>
</camelContext>
```

Notice in the XML DSL example above we use <custom> to refer to a custom data format. This requires **Camel 2.8** or better. In older releases you would have to use the `ref` attribute as shown below. Notice the `ref` attribute has been @deprecated and you should prefer to use the <custom> way:

```
<marshal ref="reverse"/>
```

See Also

- Configuring Camel
- Component
- Endpoint
- Getting Started