

Base64

Base64 Data Format

Available as of Camel 2.11

The Base64 [Data Format](#) is a data format for [base64 encoding and decoding](#).

Options

Option	Default	Description
lineLength	76	To specific a maximum line length for the encoded data.
lineSeparator	\n	The line separators to use.
urlSafe	false	Instead of emitting '+' and '/' we emit '-' and '_' respectively. urlSafe is only applied to encode operations. Decoding seamlessly handles both modes.

In Spring DSL, you configure the data format using this tag:

```
<camelContext>
  <dataFormats>
    <!-- for a newline character (\n), use the HTML entity notation coupled with the ASCII code. -->
    <base64 lineSeparator="&#10;" id="base64withNewLine" />
    <base64 lineLength="64" id="base64withLineLength64" />
  </dataFormats>
  ...
</camelContext>
```

Then you can use it later by its reference:

```
<route>
  <from uri="direct:startEncode" />
  <marshal ref="base64withLineLength64" />
  <to uri="mock:result" />
</route>
```

Most of the time, you won't need to declare the data format if you use the default options. In that case, you can declare the data format inline as shown below.

Marshal

In this example we marshal the file content to base64 object.

```
from("file://data.bin").marshal().base64().to("jms://myqueue");
```

In Spring DSL:

```
<from uri="file://data.bin">
<marshal>
  <base64/>
</marshal>
<to uri="jms://myqueue" />
```

Unmarshal

In this example we unmarshal the payload from the JMS queue to a byte[] object, before its processed by the newOrder processor.

```
from("jms://queue/order").unmarshal().base64().processRef("newOrder");
```

In Spring DSL:

```
<from uri="jms://queue/order">
<marshal>
  <base64/>
</marshal>
<to uri="bean:newOrder"/>
```

Dependencies

To use Base64 in your Camel routes you need to add a dependency on **camel-base64** which implements this data format.

If you use Maven you can just add the following to your pom.xml:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-base64</artifactId>
  <version>x.x.x</version> <!-- use the same version as your Camel core version -->
</dependency>
```