Jing

Jing Component

The Jing component uses the Jing Library to perform XML validation of the message body using either

- RelaxNG XML Syntax
- RelaxNG Compact Syntax

Maven users will need to add the following dependency to their `pom.xml` for this component:

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

Note that the MSV component can also support RelaxNG XML syntax.

URI format Camel 2.15 or older

```
rng:someLocalOrRemoteResource
rnc:someLocalOrRemoteResource
```

Where **rng** means use the RelaxNG XML Syntax whereas **rnc** means use RelaxNG Compact Syntax. The following examples show possible URI values

Example	Description
rng:foo/bar.rng	References the XML file **foo/bar.rng** on the classpath
rnc: http://foo.com/bar.rnc	References the RelaxNG Compact Syntax file from the URL, http://foo.com/bar.rnc

You can append query options to the URI in the following format, `?option=value&option=value&...`

URI format Camel 2.16

```
jing:someLocalOrRemoteResource
```

From Camel 2.16 the component use jing as name, and you can use the option compactSyntax to turn on either RNG or RNC mode.

Options

Option	Default	Description
compactSyntax	`false`	Whether to validate using RelaxNG compact syntax or not.

Example

The following example shows how to configure a route from the endpoint **direct:start** which then goes to one of two endpoints, either **mock:valid** or **mock:invalid** based on whether or not the XML matches the given RelaxNG Compact Syntax schema (which is supplied on the classpath).

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

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
    <route>
        <from uri="direct:start"/>
        <doTry>
            <to uri="jing:org/apache/camel/component/validator/jing/schema.rnc?compactSyntax=true"/>
            <to uri="mock:valid"/>
            <doCatch>
                <exception>org.apache.camel.ValidationException</exception>
                <to uri="mock:invalid"/>
            </doCatch>
            <doFinally>
                <to uri="mock:finally"/>
            </doFinally>
        </doTry>
    </route>
</camelContext>
```

See Also

- Configuring Camel
- Component
- Endpoint
- Getting Started