# Validation

## Validation Component

The Validation component performs XML validation of the message body using the JAXP Validation API and based on any of the supported XML schema languages, which defaults to XML Schema

Note that the Jing component also supports the following useful schema languages:

- RelaxNG Compact Syntax
- RelaxNG XML Syntax

The MSV component also supports RelaxNG XML Syntax.

### URI format
validator:someLocalOrRemoteResource

Where **someLocalOrRemoteResource** is some URL to a local resource on the classpath or a full URL to a remote resource or resource on the file system which contains the XSD to validate against. For example:

- `msv:org/foo/bar.xsd`
- `msv:file:../foo/bar.xsd`
- `msv:http://acme.com/cheese.xsd`
- `validator:com/mypackage/myschema.xsd`

Maven users will need to add the following dependency to their `pom.xml` for this component when using **Camel 2.8** or older:

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

From Camel 2.9 onwards the Validation component is provided directly in the camel-core.

### Options
confluenceTableSmall

| Option | Default | Description |
|---|---|---|
| resourceResolverFactory | DefaultValidatorResourceResolverFactory | **Camel 2.17**: Reference to a `org.apache.camel.component.validator.ValidatorResourceResolverFactory` which creates a resource resolver per endpoint. The default implementation creates an instance of `org.apache.camel.component.validator.DefaultLSResourceResolver per endpoint` which creates the default resource resolver `org.apache.camel.component.validator.DefaultLSResourceResolver. The default resource resolver` reads the schema files from the classpath and the file system. This option instead of the option `resourceResolver` shall be used when the resource resolver depends on the resource URI of the root schema document specified in the endpoint; for example, if you want to extend the default resource resolver. This option is also available on the validator component, so that you can set the resource resolver factory only once for all endpoints. |
| `resourceResolver` | `null` | **Camel 2.9:** Reference to a `org.w3c.dom.ls.LSResourceResolver` in the Registry. |
| `useDom` | `false` | Whether `DOMSource`/`DOMResult` or `SaxSource`/`SaxResult` should be used by the validator. |
| `useSharedSchema` | `true` | **Camel 2.3:** Whether the `Schema` instance should be shared or not. This option is introduced to work around a JDK 1.6.x bug. Xerces should not have this issue. |
| `failOnNullBody` | `true` | **Camel 2.9.5/2.10.3:** Whether to fail if no body exists. |
| `headerName` | `null` | **Camel 2.11:** To validate against a header instead of the message body. |
| `failOnNullHeader` | `true` | **Camel 2.11:** Whether to fail if no header exists when validating against a header. |

### Example

The following example shows how to configure a route from 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 schema (which is supplied on the classpath).{snippet:id=example|lang=xml|url=camel/trunk/components/camel-spring/src/test/resources/org/apache/camel/component/validator/camelContext.xml}

### Advanced: JMX method clearCachedSchema

Since **Camel 2.17**, you can force that the cached schema in the validator endpoint is cleared and reread with the next process call with the JMX operation `clearCachedSchema.` You can also use this method to programmatically clear the cache. This method is available on the `ValidatorEndpoint` class.

## Advanced: Global Option "CamelXmlValidatorAccessExternalDTD"

Since **Camel 2.19, 2.18.3, and 2.17.6** the default schema factory no longer allows reading external DTDs and external DTD entities. To achieve the old behavior where it was possible to access external DTDs and DTDs entities you can set the CamelContext global option "CamelXmlValidatorAccessExternalDTD" to "true". Prior to 2.19 global options where called properties.

Endpoint See Also