

Spark-rest

Spark-rest Component

Available as of Camel 2.14

The Spark-rest component allows to define REST endpoints using the [Spark REST Java library](#) (not to be mistaken with [Apache Spark](#)) using the [Rest DSL](#).

This component integrates with Spark REST Java library which is a REST library. This is not related to [Apache Spark](#) which is a project about big data.

 Apache Camel also provides a Camel component ([camel-spark](#)) for integrating Camel with Apache Spark. Spark REST Java Library requires Java 8 runtime.

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

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-spark-rest</artifactId>
  <version>${camel-version}</version>
</dependency>
```

URI format

```
spark-rest://verb:path?[options]
```

URI Options

Name	Default Value	Description
verb		get, post, put, patch, delete, head, trace, connect, or options.
path		the content path which support Spark syntax. See further below for examples.
accept		accept type such as: 'text/xml', or 'application/json'. By default we accept all kinds of types.

Path using Spark syntax

The path option is defined using a Spark REST syntax where you define the REST context path using support for parameters and splat. See more details at the [Spark Java Route](#) documentation.

The following is a Camel route using a fixed path

```
from("spark-rest:get:hello")
  .transform().constant("Bye World");
```

And the following route uses a parameter which is mapped to a Camel header with the key "me".

```
from("spark-rest:get:hello/:me")
  .transform().simple("Bye ${header.me}");
```

Mapping to Camel Message

The Spark Request object is mapped to a Camel Message as a `org.apache.camel.component.sparkrest.SparkMessage` which has access to the raw Spark request using the `getRequest` method. By default the Spark body is mapped to Camel message body, and any HTTP headers / Spark parameters is mapped to Camel Message headers. There is special support for the Spark splat syntax, which is mapped to the Camel message header with key `splat`.

For example the given route below uses Spark splat (the asterisk sign) in the context path which we can access as a header form the Simple language to construct a response message.

```
from("spark-rest:get:/hello/*/to/*")
  .transform().simple("Bye big ${header.splat[1]} from ${header.splat[0]}");
```

Rest DSL

Apache Camel provides a new Rest DSL that allow to define the REST services in a nice REST style. For example we can define a REST hello service as shown below:

See more details at the [Rest DSL](#).

More examples

There is a **camel-example-spark-rest-tomcat** example in the Apache Camel distribution, that demonstrates how to use camel-spark-rest in a web application that can be deployed on Apache Tomcat, or similar web containers.

See Also

- [Configuring Camel](#)
- [Component](#)
- [Endpoint](#)
- [Getting Started](#)
- [Rest](#)