

Swagger

Swagger Scala Component (deprecated)

Available as of Camel 2.14

The [Rest DSL](#) can be integrated with the `camel-swagger` module which is used for exposing the REST services and their APIs using [Swagger](#).

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



This component is deprecated. From Camel 2.16 onwards use the new Java based swagger module [Swagger Java](#)

The Scala based `camel-swagger` module is deprecated, and to be removed in a future release.

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

Using with Camel 2.15.x

The default servlet supports any environment using JMX to discover the `CamelContext(s)` to use.

The name of the servlet is `org.apache.camel.component.swagger.DefaultCamelSwaggerServlet`.

Using with Camel 2.14.x

The Swagger servlet is integrated with Spring or [ServletListener Component](#).

Component	Servlet ClassName
spring	<code>org.apache.camel.component.swagger.spring.SpringRestSwaggerApiDeclarationServlet</code>
servetlistener	<code>org.apache.camel.component.swagger.servetlistener.ServletListenerRestSwaggerApiDeclarationServlet</code>

The servlet support the same options when using spring or servetlistener.



The servlets above from Camel 2.14.x is deprecated and replaced with a single default servlet from Camel 2.15 onwards.

For example when using Spring you need to configure the `org.apache.camel.component.swagger.spring.SpringRestSwaggerApiDeclarationServlet` in the `WEB-INF/web.xml` file as shown below:



If you use Camel 2.15 onwards then just use the default servlet in any kind of environment.

```

<!-- to setup Camel Swagger api servlet when using Spring -->
<servlet>

    <!-- Camel 2.14.x -->
    <servlet-name>ApiDeclarationServlet</servlet-name>
    <servlet-class>org.apache.camel.component.swagger.spring.SpringRestSwaggerApiDeclarationServlet</servlet-
class>

    <!-- Camel 2.15 onwards -->
    <servlet-name>ApiDeclarationServlet</servlet-name>
    <servlet-class>org.apache.camel.component.swagger.DefaultCamelSwaggerServlet</servlet-class>

    <!-- Camel 2.14.x -->
    <init-param>
        <param-name>base.path</param-name>
        <param-value>http://localhost:8080/rest</param-value>
    </init-param>
    <init-param>
        <param-name>api.path</param-name>
        <param-value>http://localhost:8080/api-docs</param-value>
    </init-param>

    <!-- Camel 2.15 onwards -->
    <init-param>
        <!-- we specify the base.path using relative notation, that means the actual path will be calculated at
runtime as
            http://server:port/contextpath/rest -->
        <param-name>base.path</param-name>
        <param-value>rest</param-value>
    </init-param>
    <init-param>
        <!-- we specify the api.path using relative notation, that means the actual path will be calculated at
runtime as
            http://server:port/contextpath/api-docs -->
        <param-name>api.path</param-name>
        <param-value>api-docs</param-value>
    </init-param>

    <init-param>
        <param-name>api.version</param-name>
        <param-value>1.2.3</param-value>
    </init-param>
    <init-param>
        <param-name>api.title</param-name>
        <param-value>User Services</param-value>
    </init-param>
    <init-param>
        <param-name>api.description</param-name>
        <param-value>Camel Rest Example with Swagger that provides an User REST service</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
</servlet>

<!-- swagger api declaration -->
<servlet-mapping>
    <servlet-name>ApiDeclarationServlet</servlet-name>
    <url-pattern>/api-docs/*</url-pattern>
</servlet-mapping>

```

Options

The `org.apache.camel.component.swagger.RestSwaggerApiDeclarationServlet` and `org.apache.camel.component.swagger.DefaultCamelSwaggerServlet` supports the following options which can be configured as context-param in the web.xml file.

Option	Type	Description
--------	------	-------------

cors	Boolean	Whether to enable CORS. Notice this only enables CORS for the api browser, and not the actual access to the REST services. Is default false. Instead of using this option is recommended to use the CorsFilter, see further below.
swagger.version	String	Swagger spec version. Is default 1.2.
base.path	String	<p>Required: To setup the base path where the REST services is available. Using Camel 2.14.x this path must be the absolute path to the services. From Camel 2.15 onwards you can use a relative path instead (eg do not start with http/https) and camel-swagger will calculate the absolute base path at runtime, which will be</p> <pre>protocol://host:port/context-path/base.path</pre> <p>So using relative paths is much easier. See above for an example.</p>
api.version	String	The version of the api. Is default 0.0.0.
api.path	String	<p>To setup the path where the API is available (eg /api-docs). Using Camel 2.14.x this path must be the absolute path to the services. From Camel 2.15 onwards you can use a relative path instead (eg do not start with http/https) and camel-swagger will calculate the absolute base path at runtime, which will be</p> <pre>protocol://host:port/context-path/api.path</pre> <p>So using relative paths is much easier. See above for an example.</p>
api.title	String	Required. The title of the application.
api.description	String	Required. A short description of the application.
api.termsOfServiceUrl	String	A URL to the Terms of Service of the API.
api.contact	String	An email to be used for API-related correspondence.
api.license	String	The license name used for the API.
api.licenseUrl	String	A URL to the license used for the API.

CorsFilter

If you use the swagger ui to view the REST api then you likely need to enable support for CORS. This is needed if the swagger ui is hosted and running on another hostname/port than the actual REST apis. When doing this the swagger ui needs to be allowed to access the REST resources across the origin (CORS). The CorsFilter adds the necessary HTTP headers to enable CORS.

To use CORS adds the following filter `org.apache.camel.component.swagger.RestSwaggerCorsFilter` to your `web.xml`.

```
<!-- enable CORS filter so people can use swagger ui to browse and test the apis -->
<filter>
  <filter-name>RestSwaggerCorsFilter</filter-name>
  <filter-class>org.apache.camel.component.swagger.RestSwaggerCorsFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>RestSwaggerCorsFilter</filter-name>
  <url-pattern>/api-docs/*</url-pattern>
  <url-pattern>/rest/*</url-pattern>
</filter-mapping>
```

The CorsFilter sets the following headers for all requests

- Access-Control-Allow-Origin = *
- Access-Control-Allow-Methods = GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH
- Access-Control-Max-Age = 3600

- Access-Control-Allow-Headers = Origin, Accept, X-Requested-With, Content-Type, Access-Control-Request-Method, Access-Control-Request-Headers

Notice this is a very simple CORS filter. You may need to use a more sophisticated filter to set the header values differently for a given client. Or block certain clients etc.

Multiple CamelContexts

Available as of Camel 2.16

When using camel-swagger from Camel 2.16 onwards then it supports detecting all the running CamelContexts in the same JVM. These contexts are listed in the root path, eg `/api-docs` as a simple list of names in json format. To access the swagger documentation then the context-path must be appended with the Camel context id, such as `/api-docs/myCamel`.

Examples

In the Apache Camel distribution we ship the `camel-example-servlet-rest-tomcat` which demonstrates using this Swagger component.

See Also

- [Swagger Java](#)