

Camel 2.14.0 Release

Camel 2.14.0 release



New and Noteworthy

Welcome to the 2.14.0 release which approx 399 issues resolved (new features, improvements and bug fixes such as...)

- Support for running on Java 1.8 JVMs
- Spring 4.x supported, camel-test-spring module can only work with Spring 4.0.x and camel-test-spring3 module is for Spring3.x. Support for testing with Spring 4.1 coming in Camel 2.15 onwards.
- [REST DSL](#) to define REST services using a REST style with verbs such as get/post/put/delete etc. The Rest DSL supports Java and XML DSL.
- [REST DSL](#) integrated with [Swagger](#) to expose the service as swagger apis.
- Simple language defined in routes using resultType as boolean is now evaluated as predicate instead of as expression.
- Introduced RuntimeEndpointRegistry to capture runtime usage of endpoints in use during routing, such as dynamic endpoints from dynamic [EIPs](#).
- Added event notification for routes added and removed.
- Allow to configure cache size in Recipient List, Routing Slip and Dynamic Router EIPs; and as well turn caching off.
- [Netty HTTP](#) producer now supports the CamelHttpPath header to define a dynamic context-path and query parameters to call the remote http server.
- [Polling Consumer](#) allows to configure initial queue size and whether to block when full when using the default [EventDrivenPollingConsumer](#) which most components does. We now block by default, as otherwise the message may be lost if the queue was full, and a new message was offered.
- [Added](#) a generic callback to configure the [APNS](#) service builder.
- Added [StreamList](#) option to the [JDBC](#) component
- Allow to define Simple expressions for sql parameters of the [SQL](#) component.
- A new XML tokenizer that is truly XML-aware (e.g., supporting XML namespaces and complex hierarchical structures)
- Added option [parallelAggregate](#) to [Multicast](#), [Splitter](#) and [Recipient List](#) EIPs
- Support to set the variable map from message header in [Velocity](#), [Freemarker](#), [StringTemplate](#).
- Added [HazelcastInstance](#) option to [Hazelcast](#) component.
- Producers from static [EIPs](#) such as `<to>` is now also enlisted in [JMX](#) when Camel is starting routes.
- Unmarshal json to pojo using [camel-jackson](#) now supports a header to tell Camel what the pojo class name is, to allow more dynamic behavior, as otherwise the class name would be hardcoded.
- Also allow to configure [camel-jackson](#) to skip null values, by setting includes to `NOT_NULL`.
- And [camel-jackson](#) can now easily be configured to unmarshal to a `List<pojo>` or `List<Map>` instead of either a single pojo or `Map` type.
- Introduced [SynchronizationRouteAware](#) allowing to have callbacks before/after an Exchange is being routed. This allows to trigger custom code at these points, for example after a route, but before a Consumer writes any response back to a caller (InOut mode).
- [JDBC](#) now supports `outputClass` for `SelectList` type.
- Routes starting from [JMS](#) will now shutdown the consumer (`MessageListenerContainer`) quicker when `CamelContext` is being stopped. (assuming the `acceptMessagesWhileStopping` hasn't been set to true). This can also help a cleaner shutdown as otherwise some JMS clients may attempt re-connect/failover during shutdown which isn't desired anyway as we are shutting down.
- [OnCompletion](#) can now be configured to use or not use a thread pool, and as well whether to run before or after the route consumer. Running before the consumer allows to modify the exchange before the consumer writes it as response to the callee (if consumer is InOut mode)
- Added Bulk Index message operation to [Elasticsearch](#) component.
- Added [RoutePolicyFactory](#) to create and assign a `RoutePolicy` for every route, instead of having to configure this for every route.
- [MetricsRoutePolicyFactory](#) to expose route statistics using the codehale metrics library.
- Improved `NettyWorkerPoolBuilder#build` [visibility](#).
- The [SJMS](#) component now uses Apache commons-pool for the pool implementation instead of custom code.
- Using [Quartz2](#) as polling consumer (`QuartzScheduledPollConsumerScheduler`) now supports task stored in JDBC data store, which means users can setup quartz in clustered mode
- Camel component API to output json parameter schema now honors the [JSON Schema](#) spec, and also include a description of the javaType for object types.
- Using [PropertyPlaceholder](#) is now supported in `<dataFormats>` in the XML DSLs
- Add support for `Pattern`, `GroupingSeparator` & `DecimalSeparator` to [camel-bindy](#) when parsing/formatting `BigDecimal`
- Added option `aggregateOnException` to [Content Enricher](#) to allow end users to deal with exceptions in the aggregate method, such as suppressing the exception or building a custom message, etc.
- [POJO Consuming](#) now shutdown eager during stopping Camel, which ensures a graceful shutdown when using pojo routing.
- [Bean](#) component caches the bean by default (eg a single instance of the bean is reused). This can be turned off by setting `cache=false`, which now enforces a new instance of the bean is created and used once (no reuse, aka prototype scoped)

- Added support for RFC5424 to [Syslog](#) and as well having encoder/decoders for [Netty](#) out of the box.

Fixed Issues

- Timer consumer is now scheduled after CamelContext has started all routes, to ensure other routes is running when timer routes is triggered to run.
- [Netty HTTP](#) now returns HTTP status 404 if resource not found instead of 503.
- Fixed the [MongoDB](#) `readPreference` option which used to cause an `IllegalArgumentException` to be thrown.
- Fixed issue with Netty producer may leak `HashWheelTimer` instances when having many producers that are not reused.
- Fixed issue with Netty consumer should wait for join to complete when joining a multicast group over UDP
- Improved stability when testing with `camel-test-blueprint` component.
- [Netty HTTP](#) now [removes](#) `headerFilterStrategy` option after resolving it
- Fixed [POJO aggregating](#) when the parameter type was referring to a type that was class annotated.
- Fixed [MyBatis](#) to keep the message body as-is when calling stored procedures, if the procedure returns data as OUT parameters.
- Fixed Camel Karaf commands to work with multiple CamelContext from the same bundle.
- Fixed [Advice-With](#) to work with transacted routes defined in XML DSL
- Fixed [Advice-With](#) to work with Content Based Router when adding outputs to the when clauses.
- Fixed [Advice-With](#) to honor `autoStartup` option.
- Fixed [OnCompletion](#) to route the completed [Exchange](#) even if it was marked to stop / failed due exception / handled by error handler etc.
- Fixed using `@EndpointInject` with ref in `RouteBuilder` classes, to enlist the endpoints in JMX during starting CamelContext when using OSGi
- Fixed Property `Trigger.timerZone` is declared as constant for camel-quartz component but not implemented when endpoint is instantiated
- Fixed [JPA](#) to not share `EntityManager` with concurrent threads or other exchanges, as an `EntityManager` is not thread-safe accordingly to the JPA spec.
- Fixed [MDC Logging](#) to avoid losing MDC values after invoking an Async-Routing-enabled endpoint.
- Fixed [CXFRS](#) consumers to accept custom bindings.

New Enterprise Integration Patterns

- [Circuit Breaker](#) pattern implemented as a Load Balancer policy

New Components

- [camel-ahc-ws](#)
- [camel-atmosphere-websocket](#)
- [camel-box](#)
- [camel-dropbox](#)
- [camel-metrics](#)
- [camel-netty4](#)
- [camel-netty4-http](#)
- [camel-olingo2](#)
- [camel-openshift](#)
- [camel-google-drive](#)
- [camel-gora](#)
- [camel-rest](#) (in camel-core)
- [camel-spark-rest](#)
- [camel-schematron](#)
- [camel-swagger](#)

New Camel Maven Archetypes

New DSL

New Annotations

New Data Formats

- [camel-barcode](#)

New Languages

- [XMLTokenizer](#)

New Examples

- [camel-example-box-osgi](#)
- [camel-example-servlet-rest-blueprint](#)
- [camel-example-servlet-rest-tomcat](#)
- [camel-example-spark-rest-tomcat](#)
- [camel-example-splunk](#)

New Tooling

API changes

- Removed duplicate getter `getMessageHistory` on `ManagedCamelContextMBean`
- Added `org.apache.camel.spi.SynchronizationRouteAware`
- Added `beforeRoute` and `afterRoute` methods to `UnitOfWork`.
- Renamed `org.apache.camel.component.syslog.Rfc3164SyslogDataFormat` to `org.apache.camel.component.syslog.SyslogDataFormat`

Known Issues

Dependency Upgrades

You can see any dependency changes using [Camel Dependencies Version Differences web tool](#) created by Tracy Snell from the Camel Team.

The below list is a best effort, and we may miss some updates, its better to use the [Camel Dependencies Version Differences web tool](#) which uses the released projects for comparison.

- Atomikos 3.9.2 to 3.9.3
- Commons-codec 1.8 to 1.9
- CXF 2.7.11 to 3.0.1
- Deltaspike 0.7 to 1.0.1
- Dozer 5.4.0 to 5.5.1
- EhCache 2.8.1 to 2.8.3
- Groovy 2.2.2 to 2.3.4
- Guava 16.0.1 to 17.0
- Hapi 2.1 to 2.2
- Hazelcast 3.0.3 to 3.2.4
- JRuby 1.7.10 to 1.7.15
- MongoDB-Java-Driver 2.11.4 to 2.12.0
- RabbitMQ AMQ Client 3.2.2 to 3.3.0
- Restlet 2.17 to 2.2.1
- RxJava 0.17.5 to 0.20.4
- Scala 2.10.3 to 2.11.2
- Snmp4j 2.2.5 to 2.3.0
- SSHD 0.10.1 to 0.11.0
- spring-redis 1.1.0 to 1.3.4
- Twitter4j 3.0.5 x to 4.0.1

Internal changes

Changes that may affect end users

- [Polling Consumer](#) allows to configure initial queue size and whether to block when full when using the default `EventDrivenPollingConsumer` which most components does. We now block by default, as otherwise the message may be lost if the queue was full, and a new message was offered.
- `camel-scala` upgraded from Scala 2.10.x to 2.11.x
- The option `useSsl` has been removed from `camel-twitter` as `ssl` is always enabled now, enforced by twitter
- Fixed [Restlet](#) to return error code 405 (instead of 404 if a http method is not allowed)
- Routes starting from [JMS](#) will now shutdown the consumer (`MessageListenerContainer`) quicker when `CamelContext` is being stopped. (assuming the `acceptMessagesWhileStopping` hasn't been set to true). This can also help a cleaner shutdown as otherwise some JMS clients may attempt re-connect/failover during shutdown which isn't desired anyway as we are shutting down. If you want the routes to keep accepting messages during shutdown of Camel, then set `acceptMessagesWhileStopping=true`.
- [OnCompletion](#) now runs without thread pool by default. To restore old behavior set `parallelProcessing` to true.
- Setting `autoStartup` to false, and starting a `CamelContext` the 2nd time now does not start the routes, instead use the new `startAllRoutes` method on `CamelContext` to start all the routes.
- [Bean](#) component caches the bean by default (eg a single instance of the bean is reused). This can be turned off by setting `cache=false`, which now enforces a new instance of the bean is created and used once (no reuse, aka prototype scoped)

Important changes to consider when upgrading

- Java 1.6 no longer supported.
- Unit testing with Spring 3.x now requires using `camel-test-spring3`, as `camel-test-spring` is for Spring 4.0.x. Support for testing with Spring 4.1 coming in Camel 2.15 onwards.
- Support for Jetty 7.x is deprecated and dropped in next release.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.14.0.zip	apache-camel-2.14.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.14.0.tar.gz	apache-camel-2.14.0.tar.gz.asc

Source Distribution

Description	Download Link	PGP Signature file of download
Source	apache-camel-2.14.0-src.zip	apache-camel-2.14.0-src.zip.asc

Getting the Binaries using Maven 2

To use this release in your maven project, the proper dependency configuration that you should use in your [Maven POM](#) is:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-core</artifactId>
  <version>2.14.0</version>
</dependency>
```

Changelog

For a more detailed view of new features and bug fixes, see the:

- [release notes for 2.14.0](#)