

Camel 2.18.0 Release

Camel 2.18.0 release

[red URL](#)

New and Noteworthy

Welcome to the 2.18.0 release which resolved over 500 issues. This is first release that requires Java 8 and comes with a much-improved Spring Boot support, and ships with numerous new features, improvements and bug fixes.

- Java DSL with experimental Java8 functional/lambda support. Check out the [camel-example-java8](#). We love feedback on this DSL and expect to improved the API over the next couple of releases.
- The XSD schema for `<camelContext>` and its other configuration elements are now also documented (before it was only `<routes>` and `<rests>`). The XSD schema now also documents the default values of all the options.
- Introduced `FluentProducerTemplate` using fluent builder style as a alternative to [ProducerTemplate](#)
- All Camel [Components](#) that has options to be configured now supports and include [Spring Boot](#) auto configuration for those options, so they can easily be configured in your application.yaml file when using Spring Boot with Camel.
- Camel [Spring Boot](#) now supports the same [advanced configuration](#) you can do with Spring XML, if the beans have been configured using `@Beans` style in a Spring Boot configuration class.
- Added [Hystrix EIP](#) as EIP pattern that uses native Netflix Hystrix as the Circuit Breaker implementation. This requires having `camel-hystrix` on the classpath.
- Added [Service Call EIP](#) as EIP pattern that allows to call a remote service in a distributed system, where the service is looked up from a service registry of some sorts, such as kubernetes, consul, etcd, zookeeper etc.
- Running Camel with [Spring Boot](#) now includes a Camel Health Indicator (actuator) if `spring-boot-starter-actuator` is on the classpath.
- The [Rest](#) component allows to call REST services now (as a client), where you can pick one of the following Camel components for the HTTP transport: http, http4, netty4-http, jetty, restlet, undertow. You can also refer to a existing swagger api doc and include `camel-swagger-java` on the classpath, for automatic validation of rest endpoint is configured to use a valid rest operation/parameters.
- The Rest DSL now auto discovers which HTTP component to use if no explicit component name has been configured, by the available components on the classpath (by their default name) and if there only exists one, then that is used.
- Spring-DM for OSGi moved out of camel-spring into a separate camel-spring-dm module.
- Improved the [Bean](#) component to better match method parameter types when using [Simple](#) language as parameter values.
- Added `BindyConverter` that allows to implement custom data converters for [Bindy](#)
- The access in the [Rest DSL](#) has been deprecated and no longer in use - its not part of swagger specification anymore.
- [Camel-NATS](#) now uses JNATS client instead of the deprecated `Java_nats` one.
- During startup of [CamelContext](#) the services that are used as part of routes are now deferred being started to the end of the startup process. Some IoC frameworks like Spring can otherwise causes a circular dependency issue if services are started too early. A side effect is that if service startup failures happen when being started later, they are now wrapped in the `FailedToStartupRouteException` to better pin point which route has the problem.
- Improved the startup sequence of [Spring Java Config](#) to be similar to [Spring Boot](#) that helps prevent Spring initialization errors about circular dependencies issues.
- Added PATCH to [Rest DSL](#).
- Added "starts with" and "ends with" operator to the [Simple](#) language.
- Added `BeanIOSplitter` to [BeanIO](#) that can be used with the [Splitter](#) EIP to split big payloads in streaming mode without reading the entire content into memory.
- Some of the [AWS](#) components allows to specify ARN in the endpoint configuration.
- The create operation in [Zookeeper](#) now creates sub paths if missing.
- Added support for async mode for [SERVLET](#) component to leverage Asynchronous Servlet from the Servlet 3.0 spec.
- [Bean](#) component and [Bean Language](#) validates method name must be a valid according to java identifier rules, and also if parameter syntax has an ending parenthesis.
- You can now use `@RunWith(CamelSpringBootTest.class)` to test Camel [Spring Boot](#) applications and use the Camel test annotations from [Spring Testing](#) such as `@MockEndpoints`.
- To turn on logging exhausted message body with the message history you can configure this easily on the CamelContext level with `setLogExhaustedMessageBody`
- [Camel-Infinispan](#) now supports Aggregation Repository: `InfinispanLocalAggregationRepository` and `InfinispanRemoteAggregationRepository`
- The [SQL Component](#) and [EISql](#) now supports `outputType=StreamList` to use an iterator for the output of the SQL query that allows to process the data in a streaming fashion such as with the [Splitter](#) EIP to process the data row by row, and load data from the database as needed.
- [JPA](#) now includes a `JpaPollingConsumer` implementation that better supports [Content Enricher](#) using `pollEnrich` to do a on-demand poll that returns either none, one or a list of entities as the result.

- Calling [Bean](#) with method parameters defined using [Simple](#) parameters, now avoids an intermediate conversion of the parameters to a String value. This ensures the passed in values when calling the bean method is using the parameter type as-is from [Simple](#).
- [Camel CDI](#) now supports importing Camel XML configuration files
- [Camel CDI](#) does not deploy an empty Camel context bean anymore if not route builder beans nor Camel beans are deployed
- [Camel CDI](#) adds the `@Named` qualifier to Camel route management events so that it's possible to observe these events for a specific route with an explicit `id`
- [Camel BeanIO](#) now supports the possibility to use a custom `BeanReaderErrorHandler` implementation in his configuration
- [Camel Kubernetes](#) now supports Kubernetes ConfigMap feature
- The [Tokenizer](#) and [XMLTokenizer](#) language now supports using [Simple](#) expressions as the token / xml tag names so they can be dynamic values.
- Added `filterDirectory` and `filterFile` options to [File2](#) so filtering can be done using [Simple](#) language or predicates.
- Optimize Camel to only enable `AllowUseOriginalMessage` if in use by error handler or [OnCompletion](#). End user who manually access the original message using the Java API must configure `AllowUseOriginalMessage=true`.
- [Camel-AHC](#) [Camel-HTTP](#) [Camel-HTTP4](#) [Camel-Jetty](#) now support a `connectionClose` parameter to allow explicitly adding a Connection Close header to HTTP request
- [Bindy](#) allows to plugin custom formatters for mapping to custom types.
- [Content Enricher](#) using `pollEnrich` now supports consumers configured with `consumer.bridgeErrorHandler=true` to let any exceptions from the poll propagate to the route error handler, to let it be able to perform redeliveries and whatnot.
- [CXF](#) and [CXFRS](#) now support setting of the SSL-context [Using the JSSE Configuration Utility](#)
- [MongoDB](#) now is fully converted to MongoDB 3 although we still use `BasicDBObject` instead of `Document`
- Camel Spring Boot can now scan for classes in Spring Boot FAR jars with embeds third party JARs.
- You can now set the SNI Hostnames using the [Camel Configuration Utilities](#) to indicate the hostnames you try to connect
- The XML DSL will preserve double spaces in the context-path of uri attributes when removing white space noise, when uri's are configured using mutli lines.
- The Camel Catalog module can now load older versions of Camel to be used when querying the catalog. There is a `camel-catalog-maven` module that is able to download catalog JARs from Maven central.
- A new Camel Attachment interface was added that allows propagating headers for attachments. [Camel CXF](#), [Camel Mail](#) (including the [MIME-Multipart](#) data format), and [Camel-Jetty](#) set and consume attachment headers.
- Improved [bean method call](#) to validate if method name with parameters has valid number of parenthesis in the syntax.
- The [JsonPath](#) now supports inlined [Simple](#) language expressions to allow more dynamic expressions.
- Improved Netty4 producer to be fully asynchronous when connecting to remote server.
- The [Websocket](#) component now uses a timeout when sending to websocket channels to avoid potentially blocking for a long time due networking issues with clients.
- [Hazelcast Component](#) now provide a [RoutePolicy](#).
- Saxon has been upgraded to version 9.7

Fixed these issues:

- Fixed [Bean](#) component to avoid ambiguous error for classes that extends generic interface and calling which could lead to falsely duplicate methods (due Java type erasure inserts bridge methods)
- Fixed splitting using `tarfile` could cause OOME if splitting big files which was mistakenly loaded into memory. Now we work on the tar stream directly.
- Fixed [Netty HTTP](#) and [Netty4 HTTP](#) issue when not specifying a port number then port 80 would not be used but an error about port -1 is not allowed.
- Fixed [Swagger Java](#) when using property placeholders in [Rest DSL](#) could cause invalid parameters to be included that was from the placeholder.
- The `threads` EIP now lets [Error handling in Camel](#) perform redeliveries if the thread pool would otherwise reject accepting the task. This allows the error handler to perform redeliveries to attempt to put the task on the thread pool queue, or eventually move the message to a dead letter queue etc.
- Fixed [Rest DSL](#) adding empty header if specifying a non required query parameter that has no default value assigned.
- Fixed [doWhile](#) loop which could potentially loop forever.
- Fixed a NPE in [Zookeeper](#) consumer if no zookeeper node path was set
- When using continued with [onException](#) then dead letter channel endpoint should not be invoked.
- Fixed [Error Handler](#) to not log exceptions when using `continued(true)` by default.
- Fixed so using `shareUnitOfWork` would now also call specialized `AggregationStrategy` for `onTimeout`, `onCompletion` etc.
- Fixed [Jetty](#) consumer incorrectly handle multipart/form data not being mapped as attachments on the Camel Message.
- Fixed [Netty4 HTTP](#) may fail reading the http content from the raw netty stream if the Exchange was routed asynchronously.
- Fixed [Netty4 HTTP](#) leak `ByteBuf`'s on the producer side which was not released in all corner cases before they may be garbage collected.
- Fixed [Dozer](#) not able to use variables in mapping files when using OSGi.
- Fixed a potential dead-lock when doing request/reply over [JMS](#) and requests are timing out concurrently and continued routing the exchanges are calling another [JMS](#) endpoint that is also doing request/reply which also timeout.
- Fixed [Load Balancer](#) EIPs to support using any [property placeholder](#) using the **prop:** prefix.
- Fixed context scoped [OnCompletion](#) would not stop/shutdown its processors when CamelContext is being shutdown.
- Fixed memory leak in [Routing Slip](#) when the slip routes to certain kind of Camel components.
- Fixed [SQL Component](#) query parameter mis-match issue when using IN queries together with other named parameters.
- Fixed a memory leak with [CXF](#) when continuation was expired could cause Camel message not to be unregistered from in-flight registry.
- Fixed a performance regression when using `camel-jaxb`

New Components

- [camel-asterisk](#) - For interacting with Asterisk PBX Servers
- [camel-cm-sms](#) - For sending SMS messages using SM SMS Gateway.
- [camel-consul](#) - For integrating your application with [Consul](#).
- [camel-ehcache](#) - For interacting with Ehcache 3 cache.
- [camel-flink](#) - Bridges Camel connectors with Apache Flink tasks.
- [camel-lumberjack](#) - For receiving logs over the lumberjack protocol (used by Filebeat for instance)
- [camel-ribbon](#) - To use Netflix Ribbon with the [Service Call EIP](#).
- [camel-servicenow](#) - For cloud management with ServiceNow.
- [camel-telegram](#) - For messaging with Telegram.
- [camel-zipkin](#) - For tracking Camel message flows/timings using [zipkin](#).

- [camel-chronicle](#) - For interacting with OpenHFT's [Chronicle-Engine](#).

New DSL

- [hystrix](#) - Using Netflix Hystrix as circuit breaker. This requires having `camel-hystrix` on the classpath.
- [serviceCall](#) - To call a remote service in a distributed system where the service is looked up from a service registry of some sorts.

New Examples

- `camel-example-cdi-xml`: illustrates the use of Camel XML configuration files into a Camel CDI application
- `camel-example-spring-boot-rest-jpa`: Spring Boot example that demonstrates how to use JPA along with Camel's REST DSL to expose a RESTful API that performs CRUD operations on a database
- `camel-example-java8`: Demonstrates the Java DSL with experimental new Java8 lambda support for expression/predicate/processor's. We love feedback on this DSL and expect to improved the API over the next couple of releases.
- `camel-example-java8-rx`: Demonstrates the Java DSL with experimental new Java8 lambda support for typesafe filtering and transforming of messages with Rx-Java. We love feedback on this DSL and expect to improved the API over the next couple of releases.

API breaking

- Removed the deprecated vtdxml language. You can use the generic language("vtdxml") instead

Known Issues

Important changes to consider when upgrading

- **Java 1.8 is now required.**
- Karaf 2.4.x is no longer supported. Karaf 4.0.x is the primary supported OSGi platform.
- Jetty 8.x is no longer supported and `camel-jetty8` has been removed
- spring 4.0 is no longer supported and `camel-test-spring40` has been removed
- spring 3.x is no longer supported
- Upgraded to Spring 4.3.x and Spring Boot 1.4.x
- (only spring-dm using spring 3.2.x as part of camel-spring in osgi/karaf is still in use - but spring-dm is deprecated and we recommend using blueprint)
- Spring-DM for OSGi moved out of camel-spring into a separate camel-spring-dm module. If using Spring XML with Spring-DM then you need to install camel-spring-dm feature in Karaf/ServiceMix.
- The deprecated `@LazyLoadTypeConverter` has been removed from all [Camel Test](#) modules
- `camel-gae` has been removed (was not working anyway)
- The access in the [Rest DSL](#) has been deprecated and no longer in use - its not part of swagger specification anymore.
- Removed the deprecated maven archetypes for SOAP WS using CXF with OSGi Blueprint
- The XML DSL will preserve double spaces in the context-path of uri attributes when removing white space noise, when uri's are configured using mutli lines.
- Optimize Camel to only enable `AllowUseOriginalMessage` if in use by error handler or [OnCompletion](#). End user who manually access the original message using the Java API must configure `AllowUseOriginalMessage=true`.
- The [FTPS](#) component is now using known hosts file from the user.home/.ssh directory by default. Set the option `useUserKnownHostsFile=false` to use old behavior.
- the [MongoDB](#) component is migrated to MongoDB 3. Returned objects might be different e.g. `findAll` always returns an Iterable (FindIterable or ArrayList). Sometimes the returning of headers is remove e.g. for the paged `findAll` the total size isn't fetch
- the [Salesforce](#) component doesn't use jodatime anymore. It uses the java 8 `ZonedDateTime`. DTO's need to be changed or regenerated.
- The camel-testng module is deprecated, use the JUnit based test modules instead.
- The [camel-cache](#) module is deprecated, you should use [camel-ehcache](#) instead.
- The camel-docker module has been removed from Karaf features as it does not work in OSGi
- In [camel-xslt](#) selecting saxon using the object model url does not work anymore (see [doc](#)), you need to explicit seth both the model url and the implementation class or use dsl/xml shortcut to enable saxon.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.18.0.zip	apache-camel-2.18.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.18.0.tar.gz	apache-camel-2.18.0.tar.gz.asc



The above URLs use redirection

The above URLs use the Apache Mirror system to redirect you to a suitable mirror for your download. Some users have experienced issues with some versions of browsers (e.g. some Safari browsers). If the download doesn't seem to work for you from the above URL then try using [Firefox](#)

Source Distributions

Description	Download Link	PGP Signature file of download
Source for Windows	apache-camel-2.18.0-src.zip	apache-camel-2.18.0-src.zip.asc

Getting the Binaries using Maven

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>2.18.0</version>
</dependency>
```

Git Tag Checkout

```
git clone https://git-wip-us.apache.org/repos/asf/camel.git
cd camel
git checkout camel-2.18.0
```

Changelog

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

- [release notes for 2.18.0](#)