Camel 2.20.0 Release

Camel 2.20.0 Release

‹ed URL

New and Noteworthy

Welcome to the 2.20.0 release which resolved over 550 issues including new features, improvements and bug fixes.

- Support for Java 9 as a technical preview. Official support for Java 9 will be forthcoming in the following releases. (source code builds and tests on a Java 9 JVM).
- Many internal optimisations in the Camel routing engine, such as reducing thread contention when updating JMX statistics, reducing internal state objects to claim less memory, and reducing the number of allocated objects to reduce overhead on GC etc, and much more.
- Camel Spring Boot now supports referring to bean's (lookup in Spring Boot) by their id names in the configuration files (application. properties|yaml file) when you configure any of the Camel starter components.
- Camel Spring Boot now also supports using Spring (auto) configuration to configure CamelContext when using Spring XML files with <camelContext>.
- Worked to make Apache Camel more ready and compatible with the upcoming Spring Boot 2 and Spring Framework 5. Officially support for these is expected in Camel 2.21 release.
- The JMS component now includes JMS 2.0 functionality to use shared (durable and non-durable) topic.
- The Camel Maven Plugin can now validate for duplicate route ids in your source code.
- Splitted Twitter component into 4, now directmessage, seach, streaming and timeline has its own endpoint and scheme. See documentation for more details
- Introduced `HeadersMapFactory` SPI which allows to plugin different implementations, or to use case sensitive maps that are faster than the default.
- Allow Kafka consumer to break on first unhandled exception, sync the offset from last known good, and then re-connect after one timeout cycle, to restart consuming again. This avoids loosing the failed message, but retry it again on either this consumer, or another consume which was re-balanced by Kafka. This requires to be turned on with the new option breakOnFirstError which can be set on both component or endpoint level.
- Starting and stoping the CamelContext when used with Spring framework (SpringCamelContext) was revised to ensure that the Camel context is started last - when all resources should be available, and stopped first - while all resources are still available
- The SQL Stored Procedure now supports specifying custom types as FQN classnames and scale in numeric values.
- Using Camel with Spring now supports calling Bean by their FQN name and let Spring instantiate the bean using auto-wired constructor's as opposed to only supporting a no-arg constructor.
- Using Camel with Spring Boot can now easily filter Java RouteBuilder routes via ANT-path pattern style to either include or exclude class names, which can be configured using Spring configuration properties.
- The Wire Tap EIP can now be configured to use static endpoint uri instead of being dynamic evaluated via the Simple language.
- The Wire Tap EIP will now complete any inflight wire tapped exchanges while shutting down to give them time to complete graceful.
- The JSonPath can now split and write each row as a string value (JSon format) instead of using a Map/POJO type with the new writeAsString option.
- The POJO Consuming Consume annotation on POJO classes now support a predicate (using simple language) to filter the message. See the camel-example-spring-boot-pojo for more.
- The internal JSon parser that is used by camel-catalog and runtime camel-catalog (from camel-core) now embeds a simple-json v2 parser which means it can parse any kind of json formatted document (before it was confined to its own dense format)
- Infinispan, Ehcache and Hazelcast caches can automatically discover cache managers in spring-boot
- Introduced an experimental *Health Checks SPI* that can be leveraged in in cloud environments to detect non healthy contexts.
- Introduced an experimental *Cluster SPI* for high availability contexts, out of the box Camel supports: atomix, consul, file, kubernetes and zookeeper as underlying clustering technologies through the respective components.
- Introduced an experimental *Route Controller SPI* which is aimed to provide more fine-grained control of routes, out of the box Camel provides the following implementations:
    - *SupervisingRouteController* which delays startup of the routes after the camel context is properly started and attempt to restart routes that have not been starter successfully
    - *ClusteredRouteController* which leverages *Cluster SPI* to start routes only when the context is elected as leader

Fixed these issues

- Fixed a infinitive recursion in Camel's Error Handler when an onException was routing to another route using direct endpoint and this route would throw a new exception that would circle back to the same onException or at a later point, which will cause an endless recursion.
- Fixed a potential issue with masking password from URI using RAW(xxx) would reveal part of the password if the password contains a & character.
- The Restlet component is now internally using curly brackets for its uri patterns instead of regular parentheses so it works similar to the other REST component and as Restlet framework itself does
- Fixed Hystrix EIP having wrong default for circuitBreakerForceClose when using camel-hystrix-starter with Spring Boot. The default should be false and not true
- Fixed Hystrix EIP when failing and running fallback not signaling to Hystrix itself so it can keep state of the failure and react accordingly to run in half-open mode as well.
- Fixed MDC logging loosing route id after calling a direct route from within a transacted route
- Fixed a regression with Bean and Simple OGNL expressions causing ambiguous method call exception when calling method implemented by super class when method is defined by interface and abstract class
- Fixed Rest DSL (server side) not returning response on all valid uri paths when clients call using a HTTP OPTIONS request
- Fixed Rest producer not using HTTP method (verb such as PUT) from the endpoint uri when calling a remote REST service
- Fixed Timer routes to shutdown more graceful and allow pending tasks to complete while they are in-flight.
- Fixed configuring Rest DSL via application.properties|yaml in Spring Boot not working.
- Fixed Simple Language to add support negative numbers(without single or double quotes) in predicates
- Fixed configuring Rest DSL in Spring Boot application.properties / yaml for properties to data format, component, api, cors etc to use a map structure and make it work.

- Fixed configuring Rest DSL using property placeholders in the path parameters such as the defaultValue.
- Fixed an issue with parallel processing (in non-streaming mode) in some EIPs may cause CPU burning cycles while waiting for pending tasks to complete or timeout.
- Fixed an issue with copying streams could block forever due IBM application server would mistakenly return 0 instead of -1 to indicate EOL for an empty stream.
- Fixed an issue with making JMS and SJMS components work with ActiveMQ Artemis that would otherwise causes a ClassCastException
- Fixed RabbitMQ to better recover connection if exchange/queue has been deleted manually on the broker.
- Fixed Websocket component wasn't working with returning static content

New Components

- camel-atomix - a component to integrate Camel with Atomix
- camel-aws - added lamda component to be used for invoking and working with AWS Lambda functions
- camel-caffeine - a component that allows you to interact with a Caffeine cache
- camel-crypto-cms - a component for cryptographic message syntax
- camel-google-bigquery - Google BigQuery data warehouse for analytics.
- camel-headersmap - a faster implementation of case-insenstive map (used by camel message headers) which can be added to classpath at runtime to be auto installed
- camel-json-validator - validates the payload of a message using Everit JSON schema validator.
- camel-iec60870 - to integrate Camel with IEC 60870-5-104 IoT devices
- camel-ldif - the ldif component allows you to do updates on an LDAP server from a LDIF body content.
- camel-master - a component that leverage *Cluster SPI* to ensure that only a single consumer in a camel cluster is active at any point in time.
- camel-reactor - a reactor based back-end for camel's reactive streams component
- camel-thrift - the Thrift component allows to call and expose remote procedures (RPC) with Apache Thrift data format and serialization mechanism
- camel-twilio - a component that allows you to interact with Twilio REST APIs to call phones, send texts, etc. from a Camel route

New Annotations

- Added predicate to Consume

New Data Formats

- camel-asn1 - the ASN.1 data format is used for file transfer with telecommunications protocols.
- camel-fastjson - JSon data format (using the FastJSon library) is used for unmarshal a JSon payload to POJO or to marshal POJO back to JSon payload.
- camel-thrift - the Thrift data format allows to call and expose remote procedures (RPC) with Apache Thrift data format and serialization mechanism

Important changes to consider when upgrading

- Maven 3.3.3 or newer is required to build the project
- camel-dropbox - upgraded to v2 api as v1 is EOL and no longer possible to use with dropbox. The v2 upgrade was not straightforward so there can be backward compatible issues, which is out of our hands.
- camel-infinispan - the result is not more set in the CamelInfinispanOperationResult header but in the in body. To change this behavior you can set the header CamelInfinispanOperationResultHeader with the name of the header that should contains the result or with the resultHeader uri option
- camel-infinispan - the uri option *command* has been deprecated and replaced by *operation* for consistency
- camel-infinispan - the commands are now int the short form PUT, GET etc. old operation names like CamelInfinispanOperationPut, CamelInfinispanOperationGet etc have been deprecated.
- camel-undertow - matchOnUriPrefix option is defaulted to be FALSE in order to make it consistent with other components like Camel HTTP components.
- Splitted Twitter component into 4, now directmessage, seach, streaming and timeline has its own endpoint and scheme. See documentation for more details
- RuntimeEndpointRegistry is no longer in extended mode by default. To use that you need to set management statistics level to Extended explicit.
- There is no RuntimeEndpointRegistry in use by default. You need to explicit configure a registry to be used, or turn it on via management agent, or set the statics level to extended mode.
- Camel with Spring XML routes will no longer register endpoints in the Spring registry from Camel routes where <from> or <to> have endpoints assigned with an explicit id attribute. The option registerEndpointIdsFromRoute can be set to true on <camelContext> to be backwards compatible. However this registration is deprecated, instead you should use <endpoint> to register Camel endpoints with id's in Spring registry.
- camel-spring-dm has been removed as it was not working properly anyway and was deprecated some releases ago. For XML DSL with OSGi use camel-blueprint instead.
- Copying streams in IOHelper from came-core now regard EOL of data if the first read byte is zero to work around issues on some application servers like IBM WebSphere. This can be turned off by setting JVM system property "camel.zeroByteEOLEnabled=false".
- The camel-jms component now dependes by default on the JMS 2.0 API (geronimo-jms_2.0_spec) instead of JMS 1.1 API (geronimo-jms_1.1_spec). However camel-jms works at runtime with both JMS 1.1 or 2.0 specs so include the JMS spec JARs of your choice.
- camel-kura upgraded to newer OSGi API version
- camel-stomp uses the destination as-is, where as before it would replace all slash characters with colon. But according to the STOMP spec the destination should be used as-is, and is broker specific.
- camel-ignite is updated from using Ignite version 1.9.x to 2.2.x
- camel-dozer has upgraded from Dozer v5 to v6 which requires migration. See Dozer migration guides https://dozermapper.github.io/gitbook/migration/v5-to-v6.html and https://dozermapper.github.io/gitbook/migration/v6-to-v61.html

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.20.0.zip	apache-camel-2.20.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.20.0.tar.gz	apache-camel-2.20.0.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 (zip)	apache-camel-2.20.0-src.zip	apache-camel-2.20.0-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>2.20.0</version>
</dependency>
```

Git Tag Checkout

```
git clone https://git-wip-us.apache.org/repos/asf/camel.git
cd camel
git checkout camel-2.20.0
```

Changelog

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

- Release notes for 2.20.0