Apache Camel Components

Automating Tasks

Component / ArtifactId / URI	Description
Timer / camel-core in org.apache.activemq `timer:timerName[?options]`	Used to generate message exchanges when a timer fires. Events can only be consumed from this endpoint.
Quartz / camel-quartz `quartz://groupName/timerName[?options]`	Provides a scheduled delivery of messages using the Quartz 1.x scheduler.
Quartz2 / camel-quartz2 `quartz2://groupName/timerName[?options]`	Provides a scheduled delivery of messages using the Quartz 2.x scheduler.

Amazon

Component / ArtifactId / URI	Description
AWS-CW / camel-aws `aws-ddb://tableName[?options]`	For working with Amazon's CloudWatch (CW).
AWS-DDB / camel-aws `aws-ddb://tableName[?options]`	For working with Amazon's DynamoDB (DDB).
AWS-SDB / camel-aws `aws-sdb://domainName[?options]`	For working with Amazon's SimpleDB (SDB).
AWS-SES / camel-aws `aws-ses://from[?options]`	For working with Amazon's Simple Email Service (SES).
AWS-SNS / camel-aws `aws-sns://topicName[?options]`	For Messaging with Amazon's Simple Notification Service (SNS).
AWS-SQS / camel-aws `aws-sqs://queueName[?options]`	For Messaging with Amazon's Simple Queue Service (SQS).
AWS-S3 / camel-aws `aws-s3://bucketName[?options]`	For working with Amazon's Simple Storage Service (S3).

Basics

Component / ArtifactId / URI	Description
Bean / camel-core `bean:beanName[?options]`	Uses the Bean Binding to bind message exchanges to beans in the Registry. Is also used for exposing and invoking Plain Old Java Objects (POJOs).
Class / camel-core `class:className[?options]`	Uses the Bean Binding to bind message exchanges to beans based on the class name. Is also used for exposing and invoking POJOs.
Context / camel-context `[context:]camelContextId:localEndpointName[?options]`	Used to refer to endpoints within a separate CamelContext to provide a simple black box composition approach so that routes can be combined into a CamelContext and then used as a black box component inside other routes in other CamelContext.
Data Format / camel-core `dataformat:name:(marshal|unmarshal)[?options]`	For working with Data Formats as if it was a regular Component supporting Endpoints and URIs.
Exec / camel-exec `exec://executable[?options]`	For executing system commands.
Language / camel-core `language://languageName[:script][?options]`	Executes Languages scripts.
Printer / camel-printer `lpr://hostName[:port]/printerPath[?options]`	Facilitates creation of printer endpoints to local, remote and wireless printers. The endpoints provide the ability to print camel directed payloads when utilized on camel routes.
Properties / camel-core `properties://key[?options]`	Facilitates using property placeholders directly in endpoint URI definitions.
Ref / camel-core `ref:name`	Component for lookup of existing endpoints bound in the Registry.

Chat

Component / ArtifactId / URI	Description
IRC / camel-irc `irc:[login@]hostName[:port]/#room[?options]`	For Internet Relay Chat (IRC) communication.
XMPP / camel-xmpp `xmpp://[login@]hostname[:port]/[participant][?options]`	Working with the Extensible Messaging and Presence Protocol (XMPP).

Clusters

Component / ArtifactId / URI	Description
JGroups / camel-jgroups in camel-extra `jgroups:clusterName[?options]`	Provides exchange of messages between Camel infrastructure and JGroups clusters.
ZooKeeper / camel-zookeeper `zookeeper://zookeeperServer[:port][/path]`	Working with ZooKeeper clusters.

Content Repositories

Component / ArtifactId / URI	Description
CMIS / camel-cmis `cmis://cmisServerUrl[?options]`	Uses the Apache Chemistry client API to interface with Content Management Interoperability Services (CMIS).
JCR / camel-jcr `jcr://user:password@repository/path/to/node[?options]`	Storing a message in a Java Content Repository (JCR) compliant repository like Apache Jackrabbit.

Endpoint Communications

Component / ArtifactId / URI	Description
Direct / camel-core `direct:someName[?options]`	Synchronous call to another endpoint from same CamelContext.
Direct-VM / camel-core `direct-vm:someName[?options]`	Synchronous call to another endpoint in another CamelContext running in the same Java virtual machine (JVM).
Disruptor / camel-disruptor `disruptor:someName[?option]`	Provides asynchronous SEDA behavior much like the standard SEDA Component, but utilizes a Disruptor instead of a BlockingQueue.
Disruptor VM / camel-disruptor `disruptor-vm:someName[?option]`	Same as Disruptor, but the buffers of the Disruptor VM endpoints also provide support for communication across CamelContexts instances so you can use this mechanism to communicate across web applications.
SEDA / camel-core `seda:someName[?options]`	Asynchronous call to another endpoint in the same Camel Context; staged event-driven architecture (SEDA).
VM / camel-core `vm:queueName[?options]`	Asynchronous call to another endpoint in the same JVM.

ESB

Component / ArtifactId / URI	Description
JBI / servicemix-camel in org.apache.servicemix `jbi:serviceName[?options]`	For Java Business Integration (JBI) integration such as working with Apache ServiceMix.
NMR / apache-servicemix-nmr in org.apache.servicemix.nmr `nmr://serviceName[?options]`	Integration with the Normalized Message Router bus in Apache ServiceMix.
Vertx / camel-vertx `vertx:eventBusName`	Working with the Vert.x event bus.

Feeds

Component / ArtifactId / URI	Description
Atom / camel-atom `atom:atomUri[?options]`	Working with Apache Abdera for atom integration, such as consuming an atom feed.
RSS / camel-rss `rss:uri[?options]`	Working with ROME for Rich Site Summary (RSS) integration.

File I/O and Transfer

Component / ArtifactId / URI	Description
File / camel-core `file://nameOfFileOrDirectory[?options]`	Sending messages to a file or polling a file or directory.
Flatpack / camel-flatpack `flatpack:[fixed|delim]:configFile[?options]`	Processing fixed width or delimited files or messages using the FlatPack library.
FTP / camel-ftp `ftp:contextPath[?options]`	Sending and receiving files over File Transfer Protocol (FTP).
FTPS / camel-ftp `ftps://[username@]hostName[:port]/directoryName[?options]`	Sending and receiving files over FTP Secure (TLS and SSL).
HDFS / camel-hdfs `hdfs://hostName[:port][/path][?options]`	For reading/writing from/to an Hadoop Distributed File System (HDFS) filesystem.
SCP / camel-jsch `scp://hostName[:port]/destination[?options]`	Support for the scp protocol using the Client API of the Java Secure Channel (JSch) project.
SFTP / camel-ftp `sftp://[username@]hostName[:port]/directoryName[?options]`	Sending and receiving files over FTP Secure (FTP and SSH).
Stream / camel-stream `stream:(in|out|err|file|header|url)[?options]`	Read or write to an input/output/error/file stream rather like unix pipes.

Google

Component / ArtifactId / URI	Description
GAuth / camel-gae `ghttp:contextPath[?options]`	Used by web applications to implement an OAuth consumer.
GHttp / camel-gae `ghttp:contextPath[?options]`	Provides connectivity to the URL fetch service of Google App Engine but can also be used to receive messages from servlets.
GLogin / camel-gae `glogin://hostName[:port][?options]`	Used by Camel applications outside Google App Engine (GAE) for programmatic login to GAE applications.
GTask / camel-gae `gtask://queueName[?options]`	Supports asynchronous message processing on Google App Engine by using the task queueing service as message queue.
GMail / camel-gae `gmail://user@gGoogle]mail.com[?options]`	Supports sending of emails via the mail service of Google App Engine.
Guava EventBus / camel-guava-eventbus `guava-eventbus:busName[?options]`	The Google Guava EventBus allows publish-subscribe style communication between components without requiring the components to explicitly register with one another.

HTTP

Component / ArtifactId / URI	Description
AHC / camel-ahc `ahc:http[s]://hostName[:port][/resourceUri][?options]`	To call external HTTP services using Async Http Client.
Cometd / camel-cometd `cometd://hostName:port/channelName[?options]`	HTTP-based event routing bus used to deliver messages using the Jetty Cometd implementation of the Bayeux Protocol.
HTTP / camel-http `http:hostName[:port][/resourceUri][?options]`	For calling out to external HTTP servers using Apache HTTP Client 3.x.
HTTP4 / camel-http4 `http4:hostName[:port][/resourceUri][?options]`	For calling out to external HTTP servers using Apache HTTP Client 4.x.
Jetty / camel-jetty `jetty:hostName[:port][/resourceUri][?options]`	For exposing services over HTTP.
Servlet / camel-servlet `servlet:relativePath[?options]`	For exposing services over HTTP through the Servlet which is deployed into the Web container.

Java Message Service

Component / ArtifactId / URI	Description
ActiveMQ / activemq-camel in org.apache.activemq `activemq:[topic:]destinationName`	For JMS Messaging with Apache ActiveMQ.
ActiveMQ Journal / activemq-core `activemq.journal:directoryName[?options]`	Uses ActiveMQ's fast disk journaling implementation to store message bodies in a rolling log file.
JMS / camel-jms `jms:[queue:|topic:]destinationName[?options]`	Working with JMS provider.
RabbitMQ / camel-rabbitmq `rabbitmq://hostName[:port]/exchangeName[?options]`	For JMS Messaging with RabbitMQ.
SJMS / camel-sjms `sjms:[queue:|topic:]destinationName[?options]`	From the ground upwards implementation of a JMS client.
Stomp / camel-stomp `stomp:queue:destinationName[?options]`	For communicating with Stomp compliant message brokers, like Apache ActiveMQ or ActiveMQ Apollo.
ZeroMQ / camel-zeromq in camel-extra. `zeromq:(tcp|ipc)://hostName[:port][?options]`	For JMS Messaging with ZeroMQ.

OSGi

Component / ArtifactId / URI	Description
EventAdmin / camel-eventadmin `eventadmin:topic[?options]`	Receiving OSGi EventAdmin events.
Pax Logging / camel-paxlogging `paxlogging:appender`	Receiving Pax-Logging events in OSGi.

LDAP

Component / ArtifactId / URI	Description
LDAP / camel-ldap `ldap:hostName[:port][?options]`	Performing searches on Lightweight Directory Access Protocol (LDAP) servers.
Spring LDAP / camel-spring-ldap `spring-ldap:springLdapTemplateBean[?options]`	Camel wrapper for Spring LDAP. Spring LDAP is a Java library for simplifying LDAP operations, based on the pattern of Spring's JdbcTemplate.

Mail

Component / ArtifactId / URI	Description
IMAP / camel-mail `imap://[username@]hostName[:port][?options]`	Receiving email using Internet Message Access Protocol (IMAP).
IMAPS / camel-mail `imaps://[username@]hostName[:port][?options]`	Receiving email using secured IMAP.
POP3 / camel-mail `pop3://[username@]hostName port[?options]`	Receiving email using Post Office Protocol (POP3) and JavaMail.
POP3S / camel-mail `pop3s://[username@]hostName[:port][?options]`	Receiving email using secured POP3 and JavaMail.
SMTP / camel-mail `smtp://[username@]hostName[:port][?options]`	Sending email using Simple Mail Transfer Protocol (SMTP) and JavaMail.
SMTPS / camel-mail `smtps://[username@]hostName[:port][?options]`	Sending email using secured SMTP and JavaMail.

Maintenance and Monitoring

Component / ArtifactId / URI	Description
Browse / camel-core `browse:someName`	Provides a simple BrowsableEndpoint which can be useful for testing, visualization tools or debugging. The exchanges sent to the endpoint are all available to be browsed.
ControlBus / camel-core `controlbus:command[?options]`	ControlBus EIP that allows to send messages to Endpoints for managing and monitoring your Camel applications.
JMX / camel-jmx `jmx://platform[?options]`	For working with Java Management Extensions (JMX) notification listeners.
Log / camel-core `log:loggingCategory[?options]`	Uses Jakarta Commons Logging to log the message exchange to some underlying logging system like log4j.
Nagios / camel-nagios `nagios://hostName[:port][?options]`	Sending passive checks to Nagios using JSendNSCA. Nagios supports IT Infrastructure Monitoring.
SNMP / camel-snmp `snmp://hostName[:port][?options]`	Polling OID values and receiving traps using SNMP via SNMP4J library.

Messaging

Component / ArtifactId / URI	Description
AMQP / camel-amqp `amqp:[queue:|topic:]destinationName[?options]`	For Messaging with the Advanced Message Queuing Protocol (AMQP).
Esper / camel-esper in camel-extra `esper:name[?options]`	Working with the Esper Library for Event Stream Processing.
JavaSpace / camel-javaspace `javaspace:jini://hostName[?options]`	Sending and receiving messages through JavaSpace.
JT/400 / camel-jt400 `jt400://user:pwd@system/[path_to_dtaq>[?options]`	For integrating with data queues on an AS/400 (i.e., System i, IBM i, i5, etc.) systems.
Kestrel / camel-kestrel `kestrel://[addresslist]/queueName[?options]`	For producing to or consuming from Kestrel queues.
MQTT / camel-mqtt `mqtt:name[?options]`	For communicating with MQ Telemetry Transport (MQTT) machine-to-machine (M2M) message brokers.
SIP / camel-sip `sip://user@hostName[:port][?options]`	Publish/Subscribe communication capability using the Session Initiation Protocol (SIP) protocol.
SIPS / camel-sip `sips://user@hostName[:port][?options]`	Publish/Subscribe communication capability using the secured Session Initiation Protocol (SIP) protocol.
SMPP / camel-smpp `smpp://[username@]hostName[:port][?options]`	To send and receive SMS using Short Messaging Service Center using the JSMPP library.
SMPPS / camel-smpp `smpps://[username@]hostName[:port][?options]`	To send and receive secured SMS using Short Messaging Service Center using the JSMPP library.
Quickfix / camel-quickfix `quickfix:configFile[?options]`	Implementation of the QuickFix for Java engine which allows to send/receive FIX messages.

Networking

Component / ArtifactId / URI	Description
DNS / camel-dns `dns:operation[?options]`	To lookup domain information and run Domain Name System (DNS) queries using DNSJava.
HL7 / camel-hl7 `mina2:tcp://hostName[:port][?options]`	For working with the HL7 MLLP protocol and the HL7 model using the HAPI library.
MINA / camel-mina `mina:tcp[udp|vm>:hostName[:port][?options]`	Working with Apache Multipurpose Infrastructure for Network Applications (MINA) 1.x. Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Virtual Machine (VM) protocol are supported.
MINA2 / camel-mina2 `mina2:tcp[udp|vm>:hostName[:port][?options]`	Working with Apache MINA 2.x. Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Virtual Machine (VM) protocol are supported.
Netty / camel-netty `netty:tcp[udp>://hostName[:port][?options]`	Working with TCP and UDP protocols using Java NIO based capabilities offered by the Netty project.
Netty HTTP / camel-netty-http `netty-http:http[:port]/contextPath[?options]`	Netty HTTP server and client using the Netty project.

Persistence

Component / ArtifactId / URI	Description
CouchDB / camel-couchdb `couchdb:hostName[:port][/database][?options]`	To integrate with Apache CouchDB.
Db4o / camel-db4o in camel-extra `db4o://className[?options]`	For using a db4o datastore as a queue via the db4o (database for objects) library.
EJB / camel-ejb `ejb:ejbName[?options]`	Uses the Bean Binding to bind message exchanges to Enterprise Java Beans (EJBs). It works like the Bean component but just for accessing EJBs. Supports EJB 3.0 onwards.
HBase / camel-hbase `hbase://tableName[?options]`	For reading/writing from/to an HBase store (Hadoop database).
Hibernate / camel-hibernate in camel-extra `hibernate://entityName[?options]`	For using a database as a queue via the Hibernate library.
iBATIS / camel-ibatis `ibatis://statementName[?options]`	Performs a query, poll, insert, update or delete in a relational database using Apache iBATIS.
JDBC / camel-jdbc `jdbc:dataSourceName[?options]`	For performing Java Database Connectivity (JDBC) queries and operations.
JPA / camel-jpa `jpa://entityName[?options]`	For using a database as a queue via the Java Persistence API (JPA) specification for working with OpenJPA, Hibernate or TopLink.
Krati / camel-krati `krati://datastorePath[?options]`	For producing to or consuming to Krati datastores.
MongoDB / camel-mongodb `mongodb:connectionBean[?options]`	Interacts with MongoDB databases and collections. Offers producer endpoints to perform CRUD-style operations and more against databases and collections, as well as consumer endpoints to listen on collections and dispatch objects to Camel routes.
MyBatis / camel-mybatis `mybatis://statementName[?options]`	Performs a query, poll, insert, update or delete in a relational database using MyBatis.
SQL / camel-sql `sql:queryCommand[?options]`	Performing Structured Query Language (SQL) queries using JDBC.
Spring Neo4j / camel-neo4j in camel-extra `spring-neo4j:[http://hostName[:port]/database][?options]`	Component for producing to Neo4j datastore using the Spring Data Neo4j library.

Platform Support

Component / ArtifactId / URI	Description
Activiti / activiti-camel `activiti:camelProcess:serviceTask`	For working with Activiti, a light-weight workflow and Business Process Management (BPM) platform which supports BPMN 2.
APNS / camel-apns `apns:=notify|consumer>[?options]`	For sending notifications to Apple iOS devices.
Salesforce / camel-salesforce `salesforce:topic[?options]`	To integrate with Salesforce.
SAP NetWeaver / camel-sap-netweaver `sap-netweaver:hostName[:port][?options]`	To integrate with SAP NetWeaver Gateway.

Remote Services

Component / ArtifactId / URI	Description
Avro / camel-avro `avro:[transport]:hostName[:port][/messageName][?options]`	Working with Apache Avro for data serialization.
Jclouds / camel-jclouds `jclouds:[blobstore|compute>:[provider id][?options]`	For interacting with cloud compute & blobstore service via Jclouds.
JCIFS / camel-jcifs in camel-extra `smb://user@hostName/shareName[?options]`	This component provides access to remote file systems over the CIFS/SMB networking protocol by using the JCIFS library.
RMI / camel-rmi `rmi://hostName[:port][?options]`	Working with Remote Method Invocation (RMI).
SSH component / camel-ssh `ssh:[username[:password]@]hostName[:port]`	For sending commands to a SSH server.

Search Engines

Component / ArtifactId / URI	Description
ElasticSearch / camel-elasticsearch `elasticsearch://clusterName[?options]`	For interfacing with an ElasticSearch server.
Lucene / camel-lucene `lucene:searcherName:[insert|query>[?options]`	Uses Apache Lucene to perform Java-based indexing and full text based searches using advanced analysis/tokenization capabilities.
Solr / camel-solr `solr://hostName[:port]/solr[?options]`	Uses the Solrj client API to interface with an Apache Lucene Solr server.

Security

Component / ArtifactId / URI	Description
Crypto (Digital Signatures) / camel-crypto `crypto:[sign|verify>:name[?options]`	Used to sign and verify exchanges using the Signature Service of the Java Cryptographic Extension.
XML Security / camel-xmlsecurity `xmlsecurity:[sign|verify>:name[?options]`	Used to sign and verify exchanges using the XML signature specification.

Social Media

Component / ArtifactId / URI	Description
Facebook / camel-facebook `facebook://endpoint[?options]`	Providing access to all of the Facebook APIs accessible using Facebook4J.
Twitter / camel-twitter `twitter://endpoint[?options]`	Enables the most useful features of the Twitter API by encapsulating Twitter4J.
Yammer / camel-yammer `yammer://function[?options]`	Allows you to interact with the Yammer enterprise social network.

Special support

Component / ArtifactId / URI	Description
Cache / camel-cache `cache://cacheName[?options]`	The cache component facilitates creation of caching endpoints and processors using EHCache as the cache implementation.
Geocoder / camel-geocoder `geocoder:<address:latlng>latitude,longitude>[?options]`	Supports looking up geocoders for an address, or reverse lookup geocoders from an address.
Hazelcast / camel-hazelcast `hazelcast:[/type]:cacheName[?options]`	Hazelcast is a data grid supporting map, multimap, SEDA, queue, set, atomic number and simple clusters.
RCode / camel-rcode in camel-extra `rcode://hostName[:port]/operation[?options]`	Uses Rserve to integrate Camel with the statistics environment R.
Routebox / camel-routebox `routebox:routeBoxName[?options]`	Facilitates the creation of specialized endpoints that offer encapsulation and a strategy/map based indirection service to a collection of camel routes hosted in an automatically created or user injected camel context.
Weather / camel-weather `weather://name[?options]`	Polls the weather information from Open Weather Map.

Spring

Component / ArtifactId / URI	Description
Spring Event / camel-spring `spring-event://default`	Working with Spring Application Events.
Spring Batch / camel-spring-batch `spring-batch:jobName[?options]`	To bridge Camel and Spring Batch.
Spring Integration / camel-spring-integration `spring-integration:defaultChannelName[?options]`	The bridge component of Camel and Spring Integration.
Spring Redis / camel-spring-redis `spring-redis://hostName:port[?options]`	Component for consuming and producing from a Redis key-value store.

Templates

Component / ArtifactId / URI	Description
FreeMarker / camel-freemarker `freemarker:templateName[?options]`	Generates a response using a FreeMarker template.
Mustache / camel-mustache `mustache:templateName[?options]`	Generates a response using a Mustache template.
MVEL / camel-mvel `mvel:templateName[?options]`	Generates a response using an MVFLEX Expression Language (MVEL) template.
Scalate / scalate-camel in org.fusesource.scalate `scalate:templateName`	Generates a response using a Scalate template.
StringTemplate / camel-stringtemplate `string-template:templateName[?options]`	Generates a response using a String Template.

Testing

Component / ArtifactId / URI	Description
DataSet / camel-core `dataset:name[?options]`	For load and soak testing, the DataSet provides a way to create huge numbers of messages for sending to components or asserting that they are consumed correctly.
Mock / camel-core `mock:name[?options]`	For testing routes and mediation rules using mocks.
Stub / camel-core `stub:someOtherCamelUri[?options]`	Allows you to stub out some physical middleware endpoint for easier testing or debugging.
Test / camel-spring `test:expectedMessagesEndpointUri[?options]`	Creates a Mock endpoint which expects to receive all the message bodies that could be polled from the given underlying endpoint.

Web Services

Component / ArtifactId / URI	Description
CXF / camel-cxf `cxf:[bean:cxfEndpoint|//someAddress][?options]`	Working with Apache CXF for web services integration.
CXF Bean / camel-cxf `cxfbean:serviceBeanRef[?options]`	Process the exchange using a JAX-WS or JAX-RS annotated bean from the registry.
CXFRS / camel-cxf `cxfrs:[bean:rsEndpoint|//someAddress][?options]`	Working with Apache CXF for RESTful services integration.
Restlet / camel-restlet `restlet:restletUrl[?options]`	Component for consuming and producing Restful resources using Restlet.
Spring Web Services / camel-spring-ws `spring-ws:[mapping-type:]someAddress[?options]`	Client-side support for accessing web services, and server-side support for creating your own contract-first web services using Spring Web Services.
Websocket / camel-websocket `websocket://hostName[:port][/resourceUri][?options]`	Communicating with Websocket clients.

XML

Component / ArtifactId / URI	Description
Bean Validation / camel-bean-validator `bean-validator:something[?options]`	Validates the payload of a message using the Java Validation API (JSR 303). Hibernate Validator is the reference implementation.
FOP / camel-fop `fop:outputformat[?options]`	Renders the message into different output formats using Formatting Objects Processor (FOP), which is driven by XSL formatting objects (XSL-FO).
MSV / camel-msv `msv:someLocalOrRemoteResource[?options]`	Validates the payload of a message using the MSV Library.
RNC / camel-jing `rnc:/relativeOrAbsoluteUri[?options]`	Validates the payload of a message using RelaxNG Compact Syntax.
RNG / camel-jing `rng:/relativeOrAbsoluteUri[?options]`	Validates the payload of a message using RelaxNG.
StAX / camel-stax `stax:(contentHandlerClassName|#myHandler>`	Process messages through a Simple API for XML (SAX) ContentHandler.
XQuery / camel-saxon `xquery:someXQueryResource`	Generates a response using an XQuery template.
XSLT / camel-core `xslt:templateName[?options]`	Generates a response using a Extensible Stylesheet Language Transformations (XSLT) template.
Validation / camel-core `validation:someLocalOrRemoteResource[?options]`	Validates the payload of a message using XML Schema and Java API for XML Processing (JAXP) Validation.