

Component List

Component / ArtifactId / URI	Description
AHC / camel-ahc <pre>ahc:http [s]://hostName [:port][/resourceUri] [?options]</pre>	To call external HTTP services using Async Http Client
AHC-WS / camel-ahc-ws <pre>ahc-ws [s]://hostName [:port][/resourceUri] [?options]</pre>	To exchange data with external Websocket servers using Async Http Client
AMQP / camel-amqp <pre>amqp:[queue: topic:] destinationName[?options]</pre>	For Messaging with AMQP protocol
APNS / camel-apns <pre>apns: <notify consumer>[? options]</pre>	For sending notifications to Apple iOS devices
Atmosphere-Websocket / camel-atmosphere-websocket <pre>atmosphere- websocket:/// relative path [?options]</pre>	To exchange data with external Websocket clients using Atmosphere
Atom / camel-atom <pre>atom:atomUri [?options]</pre>	Working with Apache Abdera for atom integration, such as consuming an atom feed.

Avro / camel-avro <div> <pre> avro: [transport]: [host]:[port] [/messageName] [?options] </pre> </div>	Working with Apache Avro for data serialization.
AWS-CW / camel-aws <div> <pre> aws- cw://namespac e[?options] </pre> </div>	For working with Amazon's CloudWatch (CW) .
AWS-DDB / camel-aws <div> <pre> aws- ddb://tableNa me[?options] </pre> </div>	For working with Amazon's DynamoDB (DDB) .
AWS-DDBSTREAM / camel-aws <div> <pre> aws- ddbstream://t ableName[? options] </pre> </div>	For working with Amazon's DynamoDB Streams (DDB Streams) .
AWS-EC2 / camel-aws <div> <pre> aws- ec2://label[? options] </pre> </div>	For working with Amazon's Elastic Compute Cloud (EC2) .
AWS-SDB / camel-aws <div> <pre> aws- sdb://domainN ame[?options] </pre> </div>	For working with Amazon's SimpleDB (SDB) .
AWS-SES / camel-aws <div> <pre> aws- ses://from[? options] </pre> </div>	For working with Amazon's Simple Email Service (SES) .

AWS-SNS / camel-aws <div>aws-sns://topicName[?options]</div>	For Messaging with Amazon's Simple Notification Service (SNS) .
AWS-SQS / camel-aws <div>aws-sqs://queueName[?options]</div>	For Messaging with Amazon's Simple Queue Service (SQS) .
AWS-SWF / camel-aws <div>aws-swf://<workflow activity>[?options]</div>	For Messaging with Amazon's Simple Workflow Service (SWF) .
AWS-S3 / camel-aws <div>aws-s3://bucketName[?options]</div>	For working with Amazon's Simple Storage Service (S3) .
Bean / camel-core <div>bean:beanName[?options]</div>	Uses the Bean Binding to bind message exchanges to beans in the Registry . Is also used for exposing and invoking POJO (Plain Old Java Objects).
Beanstalk / camel-beanstalk <div>beanstalk:hostname:port/tube[?options]</div>	For working with Amazon's Beanstalk .
Bean Validator / camel-bean-validator <div>bean-validator:label[?options]</div>	Validates the payload of a message using the Java Validation API (JSR 303 and JAXP Validation) and its reference implementation Hibernate Validator

Box / camel-box <div> box://endpoint-prefix/endpoint?[options] </div>	For uploading, downloading and managing files, managing files, folders, groups, collaborations, etc. on Box.com.
Braintree / camel-braintree <div> braintree://endpoint-prefix/endpoint?[options] </div>	Component for interacting with Braintree Payments via Braintree Java SDK
Browse / camel-core <div> browse:someName </div>	Provides a simple BrowsableEndpoint which can be useful for testing, visualisation tools or debugging. The exchanges sent to the endpoint are all available to be browsed.
Cache / camel-cache <div> cache://cacheName[?options] </div>	The cache component facilitates creation of caching endpoints and processors using EHCACHE as the cache implementation.
Cassandra / camel-cassandraql <div> cql:localhost/keyspace </div>	For integrating with Apache Cassandra .
Class / camel-core <div> class:className[?options] </div>	Uses the Bean Binding to bind message exchanges to beans in the Registry . Is also used for exposing and invoking POJO (Plain Old Java Objects).
Chronicle Engine / camel-chronicle <div> chronicle-engine:addresses/path[?options] </div>	Chronicle Engine is a high performance, low latency, reactive processing framework.

Chunk / camel-chunk <div> chunk: templateName [?options] </div>	Generates a response using a Chunk template
CMIS / camel-cmis <div> cmis://cmisServerUrl[?options] </div>	Uses the Apache Chemistry client API to interface with CMIS supporting CMS
Cometd / camel-cometd <div> cometd://hostName:port/channelName[?options] </div>	Used to deliver messages using the jetty cometd implementation of the bayeux protocol
Consul / camel-consul <div> consul: apiEndpoint[?options] </div>	For interfacing with an Consul .
Context / camel-context <div> context: camelContextId: localEndpointName[?options] </div>	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 CamelContexts
ControlBus / camel-core <div> controlbus: command[?options] </div>	ControlBus EIP that allows to send messages to Endpoints for managing and monitoring your Camel applications.
CouchDB / camel-couchdb <div> couchdb: hostName[:port] /database[?options] </div>	To integrate with Apache CouchDB .

Crypto (Digital Signatures) / camel-crypto <div> <pre>crypto: <sign verify> :name[? options]</pre> </div>	Used to sign and verify exchanges using the Signature Service of the Java Cryptographic Extension.
CXF / camel-cxf <div> <pre>cxof:<bean: cxofEndpoint //someAddress >[?options]</pre> </div>	Working with Apache CXF for web services integration
CXF Bean / camel-cxf <div> <pre>cxfbean: serviceBeanRe f[?options]</pre> </div>	Process the exchange using a JAX WS or JAX RS annotated bean from the registry. Requires less configuration than the above CXF Component
CXFRS / camel-cxf <div> <pre>cxfrs:<bean: rsEndpoint //address>[? options]</pre> </div>	Working with Apache CXF for REST services integration
DataFormat / camel-core <div> <pre>dataformat: name: <marshal unma rshal>[? options]</pre> </div>	for working with Data Formats as if it was a regular Component supporting Endpoints and URIs.
DataSet / camel-core <div> <pre>dataset:name [?options]</pre> </div>	For load & soak testing the DataSet provides a way to create huge numbers of messages for sending to Components or asserting that they are consumed correctly
Direct / camel-core <div> <pre>direct: someName[? options]</pre> </div>	Synchronous call to another endpoint from same CamelContext.

Direct-VM / camel-core <div> direct-vm: someName[? options] </div>	Synchronous call to another endpoint in another CamelContext running in the same JVM.
DNS / camel-dns <div> dns:operation [?options] </div>	To lookup domain information and run DNS queries using DNSJava
Disruptor / camel-disruptor <div> disruptor: someName[? <option>] disruptor-vm: someName[? <option>] </div>	To provide the implementation of SEDA which is based on disruptor
Docker / camel-docker <div> docker:// [operation]? [options] </div>	To communicate with Docker
Dozer / camel-dozer <div> dozer://name? [options] </div>	To convert message body using the Dozer type converter library.
Dropbox / camel-dropbox <div> dropbox:// [operation]? [options] </div>	The dropbox: component allows you to treat Dropbox remote folders as a producer or consumer of messages.
EJB / camel-ejb <div> ejb:ejbName[? options] </div>	Uses the Bean Binding to bind message exchanges to EJBs. It works like the Bean component but just for accessing EJBs. Supports EJB 3.0 onwards.

Ehcache / camel-ehcache <pre>ehcache://cacheName[?options]</pre>	The cache component facilitates creation of caching endpoints and processors using Ehcache 3 as the cache implementation.
ElasticSearch / camel-elasticsearch <pre>elasticsearch://clusterName[?options]</pre>	For interfacing with an ElasticSearch server.
Etcd / camel-etcd <pre>etcd:namespace[/path][?options]</pre>	For interfacing with an Etcd key value store.
Spring Event / camel-spring <pre>spring-event://default</pre>	Working with Spring ApplicationEvents
EventAdmin / camel-eventadmin <pre>eventadmin:topic[?options]</pre>	Receiving OSGi EventAdmin events
Exec / camel-exec <pre>exec://executable[?options]</pre>	For executing system commands
Facebook / camel-facebook <pre>facebook://endpoint[?options]</pre>	Providing access to all of the Facebook APIs accessible using Facebook4J

<p>File / camel-core</p> <pre>file://nameOf FileOrDirectory[?options]</pre>	<p>Sending messages to a file or polling a file or directory.</p>
<p>Flatpack / camel-flatpack</p> <pre>flatpack: [fixed delim] :configFile[?options]</pre>	<p>Processing fixed width or delimited files or messages using the FlatPack library</p>
<p>Flink / camel-flink</p> <pre>flink:dataset [?options] flink: datastream[?options]</pre>	<p>Bridges Camel connectors with Apache Flink tasks.</p>
<p>FOP / camel-fop</p> <pre>fop: outputFormat [?options]</pre>	<p>Renders the message into different output formats using Apache FOP</p>
<p>FreeMarker / camel-freemarker</p> <pre>freemarker: templateName [?options]</pre>	<p>Generates a response using a FreeMarker template</p>
<p>FTP / camel-ftp</p> <pre>ftp: contextPath[?options]</pre>	<p>Sending and receiving files over FTP.</p>
<p>FTPS / camel-ftp</p> <pre>ftps:// [username@] hostName[: port] /directoryName[?options]</pre>	<p>Sending and receiving files over FTP Secure (TLS and SSL).</p>

Ganglia / camel-ganglia <div> <pre>ganglia: destination: port[? options]</pre> </div>	Sends values as metrics to the Ganglia performance monitoring system using gmetric4j . Can be used along with JMXetric .
GAuth / camel-gae <div> <pre>gauth://name [?options]</pre> </div>	Used by web applications to implement an OAuth consumer. See also Camel Components for Google App Engine .
GHttp / camel-gae <div> <pre>ghttp: contextPath[? options]</pre> </div>	Provides connectivity to the URL fetch service of Google App Engine but can also be used to receive messages from servlets. See also Camel Components for Google App Engine .
Git / camel-git <div> <pre>git: localRepositoryPath[? options]</pre> </div>	Supports interaction with Git repositories
Github / camel-github <div> <pre>github: endpoint[? options]</pre> </div>	Supports interaction with Github
GLogin / camel-gae <div> <pre>glogin://host name[:port][? options]</pre> </div>	Used by Camel applications outside Google App Engine (GAE) for programmatic login to GAE applications. See also Camel Components for Google App Engine .
GTask / camel-gae <div> <pre>gtask://queue -name[? options]</pre> </div>	Supports asynchronous message processing on Google App Engine by using the task queueing service as message queue. See also Camel Components for Google App Engine .

<p>Google Calendar / camel-google-calendar</p> <pre>google- calendar://en dpoint-prefix /endpoint? [options]</pre>	<p>Supports interaction with Google Calendar's REST API.</p>
<p>Google Drive / camel-google-drive</p> <pre>google- drive://endpo int-prefix /endpoint? [options]</pre>	<p>Supports interaction with Google Drive's REST API.</p>
<p>Google Mail / camel-google-mail</p> <pre>google- mail://endpoi nt-prefix /endpoint? [options]</pre>	<p>Supports interaction with Google Mail's REST API.</p>
<p>GMail / camel-gae</p> <pre>gmail://user@ g[oogle]mail. com[?options]</pre>	<p>Supports sending of emails via the mail service of Google App Engine. See also Camel Components for Google App Engine.</p>
<p>Gora / camel-gora</p> <pre>gora: instanceName [?options]</pre>	<p>Supports to work with NoSQL databases using the Apache Gora framework.</p>
<p>Grape / camel-grape</p> <pre>grape: defaultMavenC oordinates</pre>	<p>Grape component allows you to fetch, load and manage additional jars when CamelContext is running.</p>
<p>Geocoder / camel-geocoder</p> <pre>geocoder: <address latl ng:latitude, longitude>[? options]</pre>	<p>Supports looking up geocoders for an address, or reverse lookup geocoders from an address.</p>

Google Guava EventBus / camel-guava-eventbus <div> <pre>guava-eventbus: busName[? options]</pre> </div>	<p>The Google Guava EventBus allows publish-subscribe-style communication between components without requiring the components to explicitly register with one another (and thus be aware of each other). This component provides integration bridge between Camel and Google Guava EventBus infrastructure.</p>
Hazelcast / camel-hazelcast <div> <pre>hazelcast:// [type]: cachename[? options]</pre> </div>	<p>Hazelcast is a data grid entirely implemented in Java (single jar). This component supports map, multimap, seda, queue, set, atomic number and simple cluster support.</p>
HBase / camel-hbase <div> <pre>hbase://table [?options]</pre> </div>	<p>For reading/writing from/to an HBase store (Hadoop database)</p>
HDFS / camel-hdfs <div> <pre>hdfs://hostName[:port][/path][? options]</pre> </div>	<p>For reading/writing from/to an HDFS filesystem using Hadoop 1.x</p>
HDFS2 / camel-hdfs2 <div> <pre>hdfs2://hostName[:port][/path][? options]</pre> </div>	<p>For reading/writing from/to an HDFS filesystem using Hadoop 2.x</p>
Hipchat / camel-hipchat <div> <pre>hipchat:// [host][: port]?options</pre> </div>	<p>For sending/receiving messages to Hipchat using v2 API</p>
HL7 / camel-hl7 <div> <pre>mina2: tcp://hostName[:port][? options]</pre> </div>	<p>For working with the HL7 MLLP protocol and the HL7 data format using the HAPI library</p>

Infinispan / camel-infinispan <pre>infinispan:// cacheName[? options]</pre>	For reading/writing from/to Infinispan distributed key/value store and data grid
HTTP / camel-http <pre>http:hostname [:port][/resourceUri] [?options]</pre>	For calling out to external HTTP servers using Apache HTTP Client 3.x
HTTP4 / camel-http4 <pre>http4: hostname[: port][/resourceUri] [?options]</pre>	For calling out to external HTTP servers using Apache HTTP Client 4.x
iBATIS / camel-ibatis <pre>ibatis://stat ementName[? options]</pre>	Performs a query, poll, insert, update or delete in a relational database using Apache iBATIS
Ignite / camel-ignite <pre>ignite:[cache /compute /messaging /...][? options]</pre>	Apache Ignite In-Memory Data Fabric is a high-performance, integrated and distributed in-memory platform for computing and transacting on large-scale data sets in real-time, orders of magnitude faster than possible with traditional disk-based or flash technologies. It is designed to deliver uncompromised performance for a wide set of in-memory computing use cases from high performance computing, to the industry most advanced data grid, highly available service grid, and streaming.
IMAP / camel-mail <pre>imap:// [username@] hostname[: port][? options]</pre>	Receiving email using IMAP

IMAPS / camel-mail <div> <pre> imaps:// [username@] hostName[: port][? options] </pre> </div>	...
IRC / camel-irc <div> <pre> irc:[login@] hostName[: port]/#room[? options] </pre> </div>	For IRC communication
IronMQ / camel-ironmq <div> <pre> ironmq: queueName[? options] </pre> </div>	For working with IronMQ a elastic and durable hosted message queue as a service.
JavaSpace / camel-javaspacespace <div> <pre> javaspace: jini://hostNa me[?options] </pre> </div>	Sending and receiving messages through JavaSpace
JBPM / camel-jbpm <div> <pre> jbpm:hostName [:port][/resourceUri] [?options] </pre> </div>	Sending messages through kie-remote-client API to jBPM.
jcache / camel-jcache <div> <pre> jcache: cacheName[? options] </pre> </div>	The JCache component facilitates creation of caching endpoints and processors using JCache / jsr107 as the cache implementation.
jclouds / camel-jclouds <div> <pre> jclouds: <blobstore co mpute>: [provider id] [?options] </pre> </div>	For interacting with cloud compute & blobstore service via jclouds

JCR / camel-jcr <pre>jcr://user: password@repo sitory/path /to/node[? options]</pre>	Storing a message in a JCR compliant repository like Apache Jackrabbit
JDBC / camel-jdbc <pre>jdbc: dataSourceNam e[?options]</pre>	For performing JDBC queries and operations
Jetty / camel-jetty <pre>jetty: hostName[: port][/resourceUri] [?options]</pre>	For exposing or consuming services over HTTP
JGroups / camel-jgroups <pre>jgroups: clusterName[? options]</pre>	The <code>jgroups:</code> component provides exchange of messages between Camel infrastructure and JGroups clusters.
JIRA / camel-jira <pre>jira://endpoi nt[?options]</pre>	For interacting with JIRA
JMS / camel-jms <pre>jms:[queue: topic:] destinationNa me[?options]</pre>	Working with JMS providers
JMX / camel-jmx <pre>jmx://platfor m[?options]</pre>	For working with JMX notification listeners
JPA / camel-jpa <pre>jpa://entityN ame[?options]</pre>	For using a database as a queue via the JPA specification for working with OpenJPA , Hibernate or TopLink

<p>JOLT / camel-jolt</p> <pre>jolt:specName [?options]</pre>	<p>The jolt: component allows you to process a JSON messages using an JOLT specification. This can be ideal when doing JSON to JSON transformation.</p>
<p>Jsch / camel-jsch</p> <pre>scp://hostName[:port] /destination [?options]</pre>	<p>Support for the scp protocol</p>
<p>JT/400 / camel-jt400</p> <pre>jt400://user: pwd@system /<path_to_data> q[?options]</pre>	<p>For integrating with data queues on an AS/400 (aka System i, IBM i, i5, ...) system</p>
<p>Kafka / camel-kafka</p> <pre>kafka://server:port[?options]</pre>	<p>For producing to or consuming from Apache Kafka message brokers.</p>
<p>Kestrel / camel-kestrel</p> <pre>kestrel://[addresslist] /queueName[?options]</pre>	<p>For producing to or consuming from Kestrel queues</p>
<p>Krati / camel-krati</p> <pre>krati://[path to datastore/][?options]</pre>	<p>For producing to or consuming to Krati datastores</p>
<p>Kubernetes / camel-kubernetes</p> <pre>kubernetes:masterUrl[?options]</pre>	<p>For integrating your application with Kubernetes standalone or on top of OpenShift.</p>
<p>Kura / camel-kura</p>	<p>For deploying Camel OSGi routes into the Eclipse Kura M2M container.</p>

Language / camel-core <div> <pre>language://languageName[:script][?options]</pre> </div>	Executes Languages scripts
LDAP / camel-ldap <div> <pre>ldap:host[:port][?options]</pre> </div>	Performing searches on LDAP servers (<scope> must be one of object onelevel subtree)
LinkedIn / camel-linkedin <div> <pre>linkedin://endpoint-prefix/endpoint?[options]</pre> </div>	Component for retrieving LinkedIn user profiles, connections, companies, groups, posts, etc. using LinkedIn REST API.
Log / camel-core <div> <pre>log:loggingCategory[?options]</pre> </div>	Uses Jakarta Commons Logging to log the message exchange to some underlying logging system like log4j
Lucene / camel-lucene <div> <pre>lucene:searcherName:<insert query>[?options]</pre> </div>	Uses Apache Lucene to perform Java-based indexing and full text based searches using advanced analysis/tokenization capabilities
Lumberjack / camel-lumberjack <div> <pre>lumberjack:host[:port]</pre> </div>	Uses the Lumberjack protocol for retrieving logs (from Filebeat for instance)
Metrics / camel-metrics <div> <pre>metrics:[meter counter histogram timer]:metricname[?options]</pre> </div>	Uses Metrics to collect application statistics directly from Camel routes.

MINA / camel-mina <pre>mina: [tcp udp vm]: host[:port][? options]</pre>	Working with Apache MINA 1.x
MINA2 / camel-mina2 <pre>mina2: [tcp udp vm]: host[:port][? options]</pre>	Working with Apache MINA 2.x
Mock / camel-core <pre>mock:name[? options]</pre>	For testing routes and mediation rules using mocks
MLLP / camel-mllp <pre>mllp:host: port[? options]</pre>	The MLLP component is specifically designed to handle the nuances of the MLLP protocol and provide the functionality required by Healthcare providers to communicate with other systems using the MLLP protocol
MongoDB / camel-mongodb <pre>mongodb: connectionBean[?options]</pre>	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
MongoDB GridFS / camel-mongodb-gridfs <pre>mongodb- gridfs:dbName [?options]</pre>	Sending and receiving files via MongoDB's GridFS system. Note: for Camel < 2.19, the URI syntax is gridfs:dbName[?options]
MQTT / camel-mqtt <pre>mqtt:name[? options]</pre>	Component for communicating with MQTT M2M message brokers

MSV / camel-msv <pre>msv: someLocalOrRe moteResource [?options]</pre>	Validates the payload of a message using the MSV Library
Mustache / camel-mustache <pre>mustache: templateName [?options]</pre>	Generates a response using a Mustache template
MVEL / camel-mvel <pre>mvel: templateName [?options]</pre>	Generates a response using an MVEL template
MyBatis / camel-mybatis <pre>mybatis://sta tementName[? options]</pre>	Performs a query, poll, insert, update or delete in a relational database using MyBatis
Nagios / camel-nagios <pre>nagios://host Name[:port][? options]</pre>	Sending passive checks to Nagios using JSendNSCA
NATS / camel-nats <pre>nats://server s[?options]</pre>	For messaging with the NATS platform.
Netty / camel-netty <pre>netty: <tcp udp> //host[:port] [?options]</pre>	Working with TCP and UDP protocols using Java NIO based capabilities offered by the Netty project

Netty4 / camel-netty4 <div> <pre> netty4: <tcp udp> //host[:port] [?options] </pre> </div>	Working with TCP and UDP protocols using Java NIO based capabilities offered by the Netty project
Netty HTTP / camel-netty-http <div> <pre> netty-http: http:[port] /context-path [?options] </pre> </div>	Netty HTTP server and client using the Netty project
Netty4 HTTP / camel-netty4-http <div> <pre> netty4-http: http:[port] /context-path [?options] </pre> </div>	Netty HTTP server and client using the Netty project 4.x
Olingo2 / camel-olingo2 <div> <pre> olingo2: endpoint /resource-path[?options] </pre> </div>	Communicates with OData 2.0 services using Apache Olingo 2.0 .
Openshift / camel-openshift <div> <pre> openshift: clientId[?options] </pre> </div>	To manage your Openshift applications.
OptaPlanner / camel-optaplanner <div> <pre> optaplanner: solverConfig [?options] </pre> </div>	Solves the planning problem contained in a message with OptaPlanner .
Paho / camel-paho <div> <pre> paho:topic[?options] </pre> </div>	Paho component provides connector for the MQTT messaging protocol using the Paho library.

Pax-Logging / camel-l-paxlogging <pre>paxlogging: appender</pre>	Receiving Pax-Logging events in OSGi
PDF / camel-pdf <pre>pdf:operation [?options]</pre>	Allows to work with Apache PDFBox PDF documents
PGEvent / camel-pgevent <pre>pgevent: dataSource[? options]</pre>	Allows for Producing/Consuming PostgreSQL events related to the LISTEN/NOTIFY commands added since PostgreSQL 8.3

POP3 / camel-mail <pre>pop3s://[username@] hostName port][?options]</pre>	Receiving email using POP3 and JavaMail
POP3S / camel-mail <pre>pop3s://[username@] hostName port][?options]</pre>	...
Printer / camel-printer <pre>lpr://host:port/path/to /prINTER[?options]</pre>	The printer component 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 <pre>properties://key[?options]</pre>	The properties component facilitates using property placeholders directly in endpoint URI definitions.
Quartz / camel-quartz <pre>quartz://groupName /timerName[?options]</pre>	Provides a scheduled delivery of messages using the Quartz 1.x scheduler
Quartz2 / camel-quartz2 <pre>quartz2://groupName /timerName[?options]</pre>	Provides a scheduled delivery of messages using the Quartz 2.x scheduler

Quickfix / camel-quickfix <div>quickfix:configFile[?options]</div>	Implementation of the QuickFix for Java engine which allow to send/receive FIX messages
RabbitMQ / camel-rabbitmq <div>rabbitmq://hostname[:port]/exchangeName[?options]</div>	Component for integrating with RabbitMQ
Ref / camel-core <div>ref:name</div>	Component for lookup of existing endpoints bound in the Registry .
Rest / camel-core <div>rest:verb:path[?options]</div>	Component for consuming Restful resources supporting the Rest DSL and plugins to other Camel rest components.
Restlet / camel-restlet <div>restlet:restletUrl[?options]</div>	Component for consuming and producing Restful resources using Restlet
REST Swagger / camel-rest-swagger <div>rest-swagger:[specificationUri#]operationId[?options]</div>	Component for accessing REST resources using Swagger specification as configuration.
RMI / camel-rmi <div>rmi://hostName[:port][?options]</div>	Working with RMI
RNC / camel-jing <div>rnc:/relativeOrAbsoluteUri[?options]</div>	Validates the payload of a message using RelaxNG Compact Syntax
RNG / camel-jing <div>rng:/relativeOrAbsoluteUri[?options]</div>	Validates the payload of a message using RelaxNG
Routebox / camel-routebox <div>routebox:routeBoxName[?options]</div>	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

RSS / camel-rss <div>rss:uri[?options]</div>	Working with ROME for RSS integration, such as consuming an RSS feed.
Salesforce / camel-salesforce <div>salesforce:topic[?options]</div>	To integrate with Salesforce
SAP NetWeaver / camel-sap-netweaver <div>sap-netweaver:hostName[:port][?options]</div>	To integrate with SAP NetWeaver Gateway
Scheduler / camel-core <div>scheduler://name?[options]</div>	Used to generate message exchanges when a scheduler fires. The scheduler has more functionality than the timer component.
schematron / camel-schematron <div>schematron://path?[options]</div>	Camel component of Schematron which supports to validate the XML instance documents.
SEDA / camel-core <div>seda:someName[?options]</div>	Asynchronous call to another endpoint in the same CamelContext
ServiceNow / camel-servicenow <div>servicenow:instanceName[?options]</div>	Camel component for ServiceNow
SERVLET / camel-servlet <div>servlet:relativePath[?options]</div>	For exposing services over HTTP through the servlet which is deployed into the Web container.
SFTP / camel-ftp <div>sftp://[username@]hostName[:port]/directoryName[?options]</div>	Sending and receiving files over SFTP (FTP over SSH).
Sip / camel-sip <div>sip://user@hostName[:port][?options]</div>	Publish/Subscribe communication capability using the Telecom SIP protocol. RFC3903 - Session Initiation Protocol (SIP) Extension for Event

SIPS / camel-sip <pre>sips://user@hostName[:port] [?options]</pre>	...
SJMS / camel-sjms <pre>sjms:[queue: topic:] destinationName[?options]</pre>	A ground up implementation of a JMS client
SJMS Batch / camel-sjms <pre>sjms-batch:[queue:] destinationName[?options]</pre>	A specialized JMS component for highly-performant transactional batch consumption from a queue.
Slack / camel-slack <pre>slack:#channel[?options]</pre>	The slack component allows you to connect to an instance of Slack and delivers a message contained in the message body via a pre established Slack incoming webhook .
SMTP / camel-mail <pre>smtps://[username@]hostName [:port][?options]</pre>	Sending email using SMTP and JavaMail
SMTP / camel-mail <pre>smtps://[username@]hostName [:port][?options]</pre>	...
SMPP / camel-smpp <pre>smpp://[username@]hostName [:port][?options]</pre>	To send and receive SMS using Short Messaging Service Center using the JSMPP library
SMPPS / camel-smpp <pre>smpps://[username@]hostName [:port][?options]</pre>	...
SNMP / camel-snmp <pre>snmp://hostName[:port][? options]</pre>	Polling OID values and receiving traps using SNMP via SNMP4J library

Solr / camel-solr <div> solr://hostName[:port]/solr[?options] </div>	Uses the Solrj client API to interface with an Apache Lucene Solr server
Apache Spark / camel-spark <div> spark:{rdd dataframe hive}[?options] </div>	Bridges Apache Spark computations with Camel endpoints.
Spark-rest / camel-spark-rest <div> spark-rest://verb:path[?options] </div>	For easily defining REST services endpoints using Spark REST Java library.
Splunk / camel-splunk <div> splunk://[endpoint][?options] </div>	For working with Splunk
SpringBatch / camel-spring-batch <div> spring-batch://jobName[?options] </div>	To bridge Camel and Spring Batch
SpringIntegration / camel-spring-integration <div> spring-integration:defaultChannelName[?options] </div>	The bridge component of Camel and Spring Integration
Spring LDAP / camel-spring-ldap <div> spring-ldap:springLdapTemplateBean[?options] </div>	Camel wrapper for Spring LDAP
Spring Redis / camel-spring-redis <div> spring-redis://hostName:port[?options] </div>	Component for consuming and producing from Redis key-value store Redis
Spring Web Services / camel-spring-ws <div> spring-ws:[mapping-type:]address[?options] </div>	Client-side support for accessing web services, and server-side support for creating your own contract-first web services using Spring Web Services

<p>SQL / camel-sql</p> <pre>sql:select * from table where id=#{?options}</pre>	Performing SQL queries using JDBC
<p>SQL Stored Procedure / camel-sql</p> <pre>sql-stored:template[? options]</pre>	Performing SQL queries using Stored Procedure calls
<p>SSH component / camel-ssh</p> <pre>ssh:[username[:password]@] hostName[:port][?options]</pre>	For sending commands to a SSH server
<p>StAX / camel-stax</p> <pre>stax: (contentHandlerClassName #m yHandler)</pre>	Process messages through a SAX ContentHandler .
<p>Stream / camel-stream</p> <pre>stream: [in out err file header url][?options]</pre>	Read or write to an input/output/error/file stream rather like unix pipes
<p>Stomp / camel-stomp</p> <pre>stomp:queue:destinationName [?options]</pre>	For communicating with Stomp compliant message brokers, like Apache ActiveMQ or ActiveMQ Apollo
<p>StringTemplate / camel-stringtemplate</p> <pre>string-template: templateName[?options]</pre>	Generates a response using a String Template
<p>Stub / camel-core</p> <pre>stub:someOtherCamelUri[? options]</pre>	Allows you to stub out some physical middleware endpoint for easier testing or debugging
<p>Telegram / camel-telegram</p> <pre>telegram://bots/authToken[? options]</pre>	Allows to exchange data with the Telegram messaging network

Test / camel-spring <div> test: expectedMessagesEndpointUri [?options] </div>	Creates a Mock endpoint which expects to receive all the message bodies that could be polled from the given underlying endpoint
Timer / camel-core <div> timer:timerName[?options] </div>	Used to generate message exchanges when a timer fires You can only consume events from this endpoint.
Twitter / camel-twitter <div> twitter://endpoint[? options] </div>	A twitter endpoint
Undertow / camel-undertow <div> undertow://host:port /context-path[?options] </div>	HTTP server and client using the light-weight Undertow server.
Validation / camel-core (camel-spring for Camel 2.8 or older) <div> validation: someLocalOrRemoteResource[? options] </div>	Validates the payload of a message using XML Schema and JAXP Validation
Velocity / camel-velocity <div> velocity:templateName[? options] </div>	Generates a response using an Apache Velocity template
Vertx / camel-vertx <div> vertx:eventBusName </div>	Working with the vertx event bus
VM / camel-core <div> vm:queueName[?options] </div>	Asynchronous call to another endpoint in the same JVM
Weather / camel-weather <div> wweather://name[?options] </div>	Polls the weather information from Open Weather Map
Websocket / camel-websocket <div> websocket://hostname[:port] [/resourceUri][?options] </div>	Communicating with Websocket clients

XML Security / camel-xmlsecurity <div> xmlsecurity:<sign verify>: name[?options] </div>	Used to sign and verify exchanges using the XML signature specification.
XMPP / camel-xmpp <div> xmpp://[login@]hostname[: port][/participant][? options] </div>	Working with XMPP and Jabber
XQuery / camel-saxon <div> xquery:someXQueryResource </div>	Generates a response using an XQuery template
XSLT / camel-core (camel-spring for Camel 2.8 or older) <div> xslt:templateName[?options] </div>	Generates a response using an XSLT template
Yammer / camel-yammer <div> yammer://function[?options] </div>	Allows you to interact with the Yammer enterprise social network
Zookeeper / camel-zookeeper <div> zookeeper://zookeeperServer [:port][/path][?options] </div>	Working with ZooKeeper cluster(s)