

# Etcd

## Etcd Component



Available since Camel 2.17.0

**Etcd** is a distributed key value store that provides a reliable way to store data across a cluster of machines.

Maven users will need to add the following dependency to their `pom.xml` for this component.

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-etcd</artifactId>
  <!-- use the same version as your Camel core version -->
  <version>x.y.z</version>
</dependency>
```

### URI format

```
etcd:namespace[/path][?options]
```

Where **namespace** represents the etcd context to which the etcd-component should operate and path is an optional attribute to define which node is being impacted.

Supported namespaces are:

- keys
- watch
- stats

### Options

| Name                       | Default Value  | Description   |
|----------------------------|--|---|
| uris                       | <a href="http://localhost:2379">http://localhost:2379</a> ,<br><a href="http://localhost:4001">http://localhost:4001</a> | Defines the URIs the component should connect to.   |
| sslContextParameters       | null   | To use a custom <code>org.apache.camel.util.jsse.SSLContextParameters</code> . See <a href="#">Using the JSSE Configuration Utility</a> |
| userName                   | null   | The user name to use for basic authentication   |
| password                   | null   | The password to use for basic authentication  |
| sendEmptyExchangeOnTimeout | false  | To send an empty message in case of timeout watching for a key (consumer only)  |
| recursive                  | false  | To apply an action recursively  |
| timeToLive                 | null   | To set the lifespan of a key in milliseconds  |
| timeout                    | null   | To set the maximum time an action could take to complete.   |

### Headers

| Name               | Type             | Description   |
|--------------------|------------------|---|
| CamelEtcdAction    | java.lang.String | The action to perform, supported values are set, delete, deleteDir, get |
| CamelEtcdNamespace | java.lang.String | The etcd context an exchange was generated/processed from               |

|                   |                   |   |
|-------------------|-------------------|---|
| CamelEtcPath      | java.lang.String  | For keys namespace, it is used to determine the node subject the the action, if not set the path from URI endpoint is used. For stats and watch namespaces it contains the path of the node beign processed |
| CamelEtcTimeout   | java.lang.Long    | To set the maximum time an action could take to complete. If not present, the timeout option is taken into account  |
| CamelEtcTtl       | java.lang.Integer | To set the lifespan of a key in milliseconds. If not present, the timeToLive option is taken into account   |
| CamelEtcRecursive | java.lang.Boolean | o apply an action recursively. If not present, the recursive option is taken into account   |

### Keys namespace example:

```
CamelContext context = new DefaultCamelContext();
context.addRoutes(new RouteBuilder() {
    public void configure() {
        from("direct:keys-set")
            .to("etcd:keys")
            .to("log:camel-etcd?level=INFO");
    }
});

Map<String, Object> headers = new HashMap<>();
headers.put(EtcdConstants.ETCD_ACTION, EtcdConstants.ETCD_KEYS_ACTION_SET);
headers.put(EtcdConstants.ETCD_PATH, "/camel/etcd/myKey");

ProducerTemplate template = context.createProducerTemplate();
template.sendBodyAndHeaders("direct:keys-set", "camel-etcd", headers);
```

### Stats namespace example:

```
CamelContext context = new DefaultCamelContext();
context.addRoutes(new RouteBuilder() {
    public void configure() {
        from("etcd:stats/leader?consumer.delay=50&consumer.initialDelay=0")
            .to("log:etcd-leader-stats?level=INFO");
        from("etcd:stats/self?consumer.delay=50&consumer.initialDelay=0")
            .to("log:etcd-self-stats?level=INFO");
        from("etcd:stats/store?consumer.delay=50&consumer.initialDelay=0")
            .to("log:etcd-store-stats?level=INFO");
    }
});
```

### Watch namespace example:

```
CamelContext context = new DefaultCamelContext();
context.addRoutes(new RouteBuilder() {
    public void configure() {
        from("etcd:watch/recursive?recursive=true")
            .marshall().json()
            .to("log:etcd-event?level=INFO");
    }
});
```

