ServicePool

ServicePool

Available as of Camel 2.0

Camel supports pluggable pools for services. At this time of write we have pools for:

- Producer

The default producer service pool is `org.apache.camel.impl.DefaultProducerServicePool` and is used by default in Camel for pooling Producer.

The need for pooling Producer is only apparent in some Components in Camel to support thread safe producers and support concurrency. And using pooling we can improve performance as we do not have the overhead of creating, starting and stopping the Producer at each invocation. So where is it needed then? Well these components uses pooled producers:

- FTP
- Mina

Usage

By default Camel uses a shared `org.apache.camel.impl.DefaultProducerServicePool` provided by `CamelContext`. This default pool will pool up till 100 producers per Endpoint. So if you have 6 endpoints there can be up till 6 x 100 in total in the pool, where each distinct Endpoint have a limit of up till 100 producers.

Pluggable producer pooling

Camel supports using a 3rd part pool implementation. What is needed is to implement an adapter by implementing the `org.apache.camel.spi.ServicePool` interface and define the generic as `Endpoint` as the key and `Producer` as the service.

Then set your custom pooling on the `CamelContext` with the `setProducerServicePool` method.

Developing pooleable producers

The producer service pool identify a producer as being pool capable if the producer implements the marker interface `org.apache.camel.SerivcePoolAware`.
As the producer will be pooled and thus long lived, your producer should be able to automatic safely recover lost connection. Usually you implement logic that re connects if needed.

See Also

- Architecture