STOMP: A Brief Overview

Christopher R. Wicks
9/19/2017

STOMP, the Simple (or Streaming) Text Oriented Message Protocol, is a text-based message passing protocol from the same school of design as HTTP [1]. It is designed so that applications in a distributed software system can communicate easily (from the perspective of an application developer) over a network.

STOMP is used primarily in the domain of message-oriented middleware. In these scenarios, many clients communicate with one or more servers, or message brokers, via some protocol. The message brokers then contain the necessary routing and transformation logic to pass that information on to other interested clients who may or may not be using the same protocol.

STOMP is designed to be a very simple and easy to use protocol. It supports topic subscription (and unsubscription), sending and acknowledging (or not acknowledging) messages, and heartbeat check-ins between the server and the client. It supports arbitrary binary encodings in messages, but the server has to know what to do with the specified encodings in order to perform any translations; such is considered business logic and is not part of the protocol specification. Alternatively, it can pass the frame data on to interesting clients without translation and assume that interested clients will know what to do with the data.

One interesting feature of STOMP is that it supports transactional sends. There are three frames that the client can pass to initiate, commit, or abort a transaction – BEGIN, COMMIT, and ABORT, respectively. Then they can include the transaction identifier in any of the messages to the server that they send to specify that those messages are intended to be a part of that transaction. The server must then block any processing on those received messages until a COMMIT frame is received for that transaction from the client.

References

[1] STOMP Specification. Retrieved from https://stomp.github.io/.