

# RQTT

---

A Rust based MQTT server with focuses on

- Asynchronous scaling using the new `async/await` syntax in Rust
- Zero-copy parsing using asynchronous buffered reads

## Development Approach

RQTT tries to follow the MQTT guidelines to the letter where possible and follows Rust code best practices where the MQTT 5.0 specification is ambiguous. The guidelines for MQTT version 5.0 are specified [here](https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#_Toc3901000). ([https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#\\_Toc3901000](https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#_Toc3901000))

## Timeline

---

Progress:

- [x] 1.0 Data Representation - 9/1/2019
- [ ] 2.0 Control Packet Format Parsing - 9/16/2019
- Reason Codes
- Payload Specifications and Properties
- [ ] 3.0 MQTT Control Packet Handling - 10/1/2019
- Core routing functionality
- Publishing / Receiving
- Subscribing/Unsubscribing from a channel
- Ping / Response handling
- [ ] 4.0 Operational Behavior - 10/15/2019
- Session state management
- Saving undelivered messages
- Quality of service level
- Topic Names/ Topic Filters
- [ ] 5.0 Security - 10/29/2019
- Authorization via 3rd party firms
- [ ] 6.0 Ensuring Conformance with websockets -11/12/2019
- Add a testing utility in github for each browser
- [ ] 7.0 Beuaracracy and other conformance - TBD - we'll see how much of this is actually coding

## Motivation

---

The world's need for low latency high throughput connections for practical devices is growing everyday. MQTT started out as a protocol for industrial machinery to communicate with a host of managing and regulating servers but quickly progressed into not only what most IoT devices use but also mainstream messaging applications. The need is clear for a robust version of this popular protocol and what better language to write it in than the one that claims to be the language for software that will stand the test of time. This project aims to create a safe, scalable and robust server for these small but important communications in our everyday lives.

This project also aims to include as few 3rd party libraries as possible and to be no-std compatible since many instances of rqtt will be running on devices without an OS.