New simplified API for IBM MQ applications in Golang!

Matt Roberts  
Published on 05/02/2019

We are very excited to introduce this new API for developers writing Golang applications to connect to IBM MQ.

This new API builds on the existingmq-golang client libraryto present a developer friendly interface based on the popular JMS 2.0 specification. Developers with experience using JMS in Java will be immediately familiar with using this JMS style Golang API. All Golang developers benefit from a simple messaging API that is based on industry best practice and experience, with no prior knowledge of Java or JMS required!

The API uses intuitive concepts like Producers and Consumers to hide the low level IBM MQ constructs. The result is easily readable code that can send and receive messages in only a few lines as shown here;

```go
// Send a message to the queue that contains the specified text string
context.CreateProducer().SendString(queue, "My first message")

// Create a consumer, using Defer to make sure it gets closed at the end of the method
consumer, _ := context.CreateConsumer(queue)
if consumer != nil { defer consumer.Close() }

// Receive a message from the queue and return the string from the message body
rcvBody := consumer.ReceiveStringBodyNoWait()
```

Easily send and receive IBM MQ messages in Golang

The repo includes detailed instructions on how to get started and a set of reusable code samples for common patterns such as sending and receiving messages, request/reply messaging and how to handle error conditions.

Try it out today on MacOS, Linux or Windows, and we look forward to hearing your comments either here or via github.com!

github.com/ibm-messaging/mq-golang-jms20