**Logotipo

Descrição gerada automaticamente**

Serviço Nacional de Aprendizagem Industrial

SENAI “Gaspar Ricardo Junior”

Curso

TÉCNICO EM DESENVOLVIMENTODE SISTEMAS

*MQTT*

Matheus Alves

VEDILSON e lEANDRO

Sorocaba

Abril – 2024

1- O que é MQTT?

O MQTT é um protocolo de mensagens baseado em padrões, ou conjunto de regras, usado para comunicação de computador para computador. Sensores inteligentes, dispositivos acessórios e outros dispositivos da Internet das Coisas (IoT) normalmente precisam transmitir e receber dados por meio de uma rede com limitação de recursos e largura de banda limitada. Esses dispositivos IoT usam o MQTT para transmissão de dados, pois é fácil de implementar e pode comunicar dados IoT com eficiência. O MQTT oferece suporte a mensagens entre dispositivos para a nuvem e da nuvem para o dispositivo. MQTT, abreviação de Message Queuing Telemetry Transport, é um protocolo de mensagens projetado para comunicação eficiente entre dispositivos. É leve, o que o torna ideal para situações em que a largura de banda é limitada ou os dispositivos têm baixo poder de processamento, como na Internet das Coisas (IoT).

## 2- Por que o protocolo MQTT é importante?

O MQTT tem recursos integrados que reduzem o tempo que o dispositivo IoT leva para se reconectar à nuvem. Além disso, define três níveis diferentes de qualidade de serviço para garantir a confiabilidade para casos de uso de IoT: no máximo uma vez, pelo menos uma vez e exatamente uma vez. O MQTT é importante porque permite uma comunicação eficiente em situações onde a largura de banda e o poder de processamento são limitados. Isso faz dele ideal para a Internet das Coisas (IoT), onde muitos dispositivos pequenos precisam enviar e receber dados.

3- História por trás do protocolo MQTT

Ele foi inventado em 1999 para ser usado no setor de petróleo e gás. Engenheiros precisavam de um protocolo para largura de banda mínima e perda mínima de bateria para monitorar oleodutos via satélite. Inicialmente, o protocolo era conhecido como Message Queuing Telemetry Transport devido ao produto da IBM MQ Series que oferecia suporte a sua fase inicial. Em 2010, a IBM lançou o MQTT 3.1 como um protocolo gratuito e aberto para qualquer pessoa implementar, que foi encaminhado, em 2013, ao órgão de especificação Organization for the Advancement of Structured Information Standards (OASIS) para manutenção. Em 2019, o OASIS lançou uma versão atualizada do MQTT 5. Agora o MQTT não é mais um acrônimo, mas é considerado o nome oficial do protocolo. Desenvolvido na década de 1990 pela IBM, o MQTT usa um padrão de publicação-assinatura. Dispositivos (editores) enviam mensagens (tópicos) para um servidor central (corretor). Outros dispositivos (assinantes) interessados ​​nesses temas podem receber as mensagens da corretora. Essa abordagem de publicação-assinatura torna a comunicação mais eficiente.

4- Broker MQTT

O broker é o elemento responsável por gerir as publicações e as subscrições do protocolo MQTT. Ele é como uma espécie de mediador entre as máquinas, capaz de fazer com que a comunicação de fato ocorra entre elas. O broker permite um desacoplamento entre as partes, novidade entre esse tipo de sistema. O Broker é o servidor intermediário da informação. É ele quem recebe os dados enviados pelos sensores e é nele onde esses dados são tratados e passados adiante. Um MQTT Broker é um servidor que atua como um hub central de mensagens. Dispositivos (publicadores) conectam-se ao corretor e enviam mensagens (tópicos) que outros dispositivos (assinantes) podem receber.

5- O MQTT é bom para IoT?

O MQTT tem muito suporte da comunidade IoT. Trabalhando com base no protocolo TCP/IP, a arquitetura iot simples do MQTT tem sido uma virada de jogo em um setor com um número crescente de objetos pequenos, baratos e de baixo consumo de energia entrando no mercado. O protocolo MQTT é executado sobre TCP/IP e é baseado em um modelo bidirecional de publicação-assinatura com comunicação sem perdas, garantindo que as mensagens MQTT sejam entregues de forma eficiente e sem perdas. O MQTT foi construído para ser um protocolo de baixo overhead que considerava fortemente a largura de banda e as limitações da CPU. MQTT é muito adequado para aplicações IoT devido a alguns motivos como: Design leve, Padrão de mensagens de publicação-assinatura e Opções flexíveis de qualidade de serviço (QoS)

6- O que é QoS no MQTT?

O MQTT (Message Queuing Telemetry Transport) é um protocolo de mensagens leve e flexível, projetado para ser eficiente em termos de largura de banda e energia. Qualidade de Serviço (QoS) em MQTT refere-se ao nível de garantia fornecido para entrega de mensagens. Existem três níveis de QoS:

No máximo uma vez (dispare e esqueça)

Pelo menos uma vez (entrega garantida, mas possíveis duplicatas)

Exatamente uma vez (entrega garantida de uma única mensagem)

7- Como o MQTT funciona?

Abaixo, há uma visão geral de como o MQTT funciona.

Um cliente MQTT estabelece uma conexão com o agente MQTT.

Depois de conectado, o cliente pode publicar mensagens, assinar mensagens específicas ou fazer as duas coisas.

Ao receber uma mensagem, o agente MQTT a encaminha aos assinantes interessados. MQTT usa um padrão de publicação-assinatura para comunicação: Os dispositivos se conectam ao corretor MQTT, Os editores enviam mensagens (tópicos) ao corretor,

Os assinantes se cadastram na corretora para receber tem as específicos, O corretor roteia mensagens dos editores para os assinantes correspondentes.

8- O que é MQTT over WSS?

O MQTT over WebSockets (WSS) é uma implementação MQTT para receber dados diretamente em um navegador da Web. O protocolo MQTT define um cliente JavaScript para fornecer suporte WSS para navegadores. Nesse caso, o protocolo funciona normalmente, mas adiciona cabeçalhos às mensagens MQTT para também oferecer suporte ao protocolo WSS. Pense nisso como a carga útil da mensagem MQTT envolvida em um envelope WSS. MQTT sobre WSS (WebSockets Secure Sockets Layer) é uma maneira segura de conectar-se a um corretor MQTT. Ele é alguém que criptografa a comunicação entre os dispositivos e o corretor, protegendo a privacidade e a integridade dos dados.

9- Header

O Header do MQTT pode variar de 2 a 5 bytes. Em relação ao primeiro byte obrigatório, os 4 primeiros bits referem-se ao tipo de mensagem, o bit seguinte refere-se ao indicador de mensagem duplicada, dois bits para identificar o QoS(qualidade de serviço) do pacote e bit para indicar de se a mensagem deve ser retida ou não para quando alguém se conectar receber a última mensagem enviada. Os próximos 4 bytes irão definir o tamanho do resto do pacote, podendo ir de 0 a 268 435 455 bits. O restante são informações que podem variar e não existe um padrão. O MQTT Message Header é um pequeno pacote de dados no início de uma mensagem que contém informações sobre a mensagem, como o tópico e o nível de QoS.

10- Casos Específicos

Caso um tópico não possua nenhum subscriber e o Broker receber uma informação referente a este tópico, tal informação será deletada. Isso só não acontecerá caso seja especificado pelo publisher que tal dado deve ser armazenado, o que é uma prática muito usada, pois permite que os subscribers possam ter a informação mais atualizada sem ter que esperar o publisher enviar a nova informação. Um outro caso é quando um publisher se conecta pela primeira vez a um Broker. Durante essa primeira conexão ele tem a oportunidade de definir uma mensagem padrão que será enviada aos subscribers caso o Broker perceba que esse publisher se desconectou dele. Existem muitos casos de uso específicos para MQTT em IoT, incluindo: Automação residencial inteligente, Monitoramento de sensores industriais, Acompanhamento de bens, Dispositivos vestíveis

11- Padrão Publish-Subscribe

O Padrão Publish-Subscribe é muito parecido com o padrão Observe, porém nesse caso adicionamos o papel do Broker, que é responsável por filtrar as mensagens e saber exatamente para quem enviar. Dessa forma, o publisher e subscriber não precisam se conhecer diretamente e apenas precisam conhecer o Broker, que é quem fará a notificação da mudança de estados e enviará essa informação para aqueles que tiverem inscritos no tópico referenciado. Por fim, o publish precisa se preocupar apenas com enviar as informações e estabelecer a conexão exclusivamente com o Broker.

É comum em alguns casos os sistemas terem atuações tanto de Publisher quanto de Subscriber. Considerando um caso de sistema que possui um sensor de temperatura e está ligado a um ar-condicionado e um sistema que acompanha a temperatura e envia o sinal para ligar o ar, os dois terão as duas situações, considerando que o sistema 1 precisa publicar as mudanças de temperatura e receber o sinal para ligar o ar, enquanto o sistema 2 precisa receber as informações da temperatura e enviar o sinal para ligar o ar.

BIBLIOGRAFIA

**aws.amazon.com** - Disponível em <https://aws.amazon.com/pt/what-is/mqtt/> > Acessado em: 30/04/2024.