EBook Gratis

APRENDIZAJE MQTT

Free unaffiliated eBook created from
Stack Overflow contributors.

#mqtt

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: mqtt

It is an unofficial and free MQTT ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official MQTT.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con MQTT

Observaciones

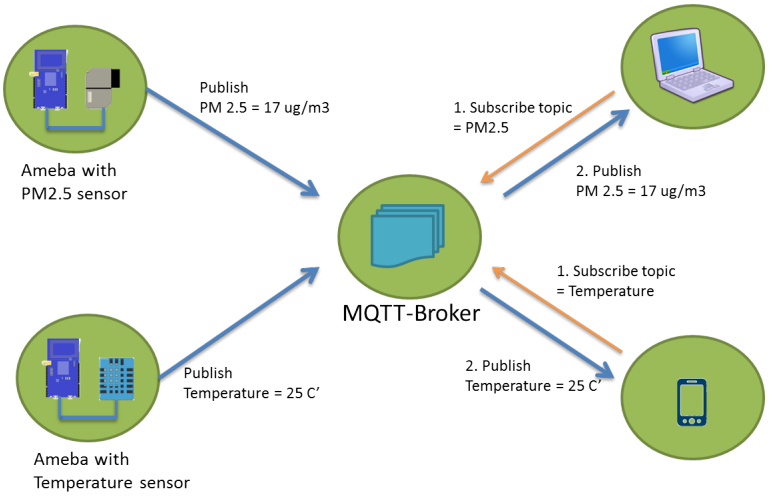
Esta sección proporciona una descripción general de qué es mqtt y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de mqtt, y vincular a los temas relacionados. Dado que la Documentación para mqtt es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Introducción

MQTT (Message Queue Telemetry Transport) es un protocolo de mensajería "ligero" basado en Publicación-Suscripción que se utiliza sobre la pila TCP / IP .



> Es muy útil para las conexiones con ubicaciones remotas donde se requiere una pequeña huella de código y / o el ancho de banda de la red es una prima.

Hay muchos Brokers y Clientes diferentes que implementan el protocolo MQTT.

Puede encontrar una lista de Brokers, Clientes y Herramientas en el sitio web mqtt.org aquí , aunque no es definitivo, ofrece una muestra representativa. También hay una lista curada en github.com en MQTT.

Puede encontrar noticias sobre las especificaciones de MQTT en mqtt.org .

MQTT v3.1.1 es un estándar de Oasis disponible aquí

Lea Empezando con MQTT en línea: https://riptutorial.com/es/mqtt/topic/8187/empezando-con-mqtt

Capítulo 2: Características de MQTT

Introducción

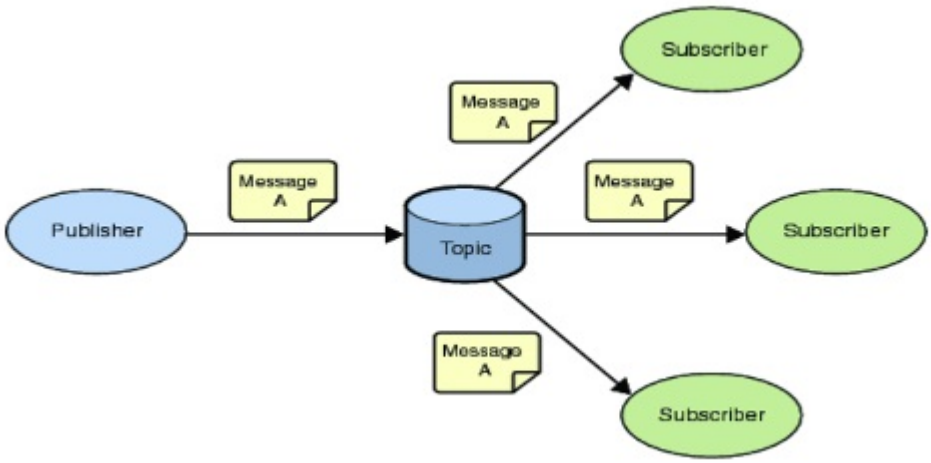
El protocolo se ejecuta sobre TCP / IP, o sobre otros protocolos de red que proporcionan conexiones bidireccionales ordenadas, sin pérdidas.

Examples

Modelo público / suscripción simple en MQTT

Sus características clave incluyen:

- Uso del patrón de mensaje de publicación / suscripción que proporciona la distribución de uno a varios mensajes y el desacoplamiento de aplicaciones.
- Un transporte de mensajería que es contrario al contenido de la carga útil. Tres calidades de servicio para la entrega de mensajes.
- Una pequeña sobrecarga de transporte e intercambios de protocolos minimizados para reducir el tra red.



Generalmente hay dos tipos de servicio de mensajería.

- Cola (conexión uno a uno)
- Tema (uno a uno / uno a muchos)

MQTT no admite la cola, que es confiable, pero MQTT es compatible con el tema. Por defecto, el tema no es confiable, pero podemos usar las características y los métodos de MQTT para hacerlo confiable.

Diferencia entre tema y cola

Cola:

- Modelo punto a punto
- Solo un consumidor recibe el mensaje.
- Los mensajes deben ser entregados en el orden enviado.
- Una cola solo garantiza que cada mensaje sea procesado una sola vez.
- La Cola sabe quién es el consumidor o el cliente JMS. El destino es conocido.
- El cliente JMS (el consumidor) no tiene que estar activo o conectado a la cola todo el tiempo para recibir o leer el mensaje.
- Cada mensaje procesado con éxito es reconocido por el consumidor.

Tema:

- Modelo de publicación / suscripción
- Varios clientes se suscriben al mensaje.
- No hay garantía de que los mensajes se entreguen en el orden enviado.
- No hay garantías de que cada mensaje sea procesado una sola vez. Como esto se puede sentir desde el modelo.
- El tema, tiene múltiples suscriptores y existe la posibilidad de que el tema no conozca a todos los suscriptores. El destino es desconocido
- El suscriptor / cliente necesita el activo cuando los mensajes son producidos por el productor, a menos que la suscripción sea una suscripción duradera.
- No, el consumidor / suscriptor no confirma todos los mensajes procesados con éxito.

  pero podemos reducir las desventajas del tema utilizando MQTT. El tema puede ser confiable y controlar los duplicados en las características de MQTT

Lea Características de MQTT en línea: https://riptutorial.com/es/mqtt/topic/9129/caracteristicas-de-mqtt

Capítulo 3: Implementación de MQTT

Examples

Ejemplo de publicación / suscriptor en java

crear proyecto web dinámico en sts / eclipse descargar el archivo eclipse paho jar desde aquí para descargar y pegar el archivo jar en webcontent-> webinf-> folder-> lib

Ejemplo de publicación

```
 String broker = "tcp://localhost:1883";
String topicName = "test/topic";
int qos = 1;

 MqttClient mqttClient = new MqttClient(broker,String.valueOf(System.nanoTime()));
//Mqtt ConnectOptions is used to set the additional features to mqtt message

  MqttConnectOptions connOpts = new MqttConnectOptions();

    connOpts.setCleanSession(true); //no persistent session
    connOpts.setKeepAliveInterval(1000);


MqttMessage message = new MqttMessage("Ed Sheeran".getBytes());
```

// aquí ed sheeran es un mensaje

```
    message.setQos(qos);     //sets qos level 1
    message.setRetained(true); //sets retained message

MqttTopic topic2 = mqttClient.getTopic(topicName);

    mqttClient.connect(connOpts); //connects the broker with connect options
    topic2.publish(message);    // publishes the message to the topic(test/topic)
```

Ejemplo de Suscripción

```
//We're using eclipse paho library  so we've to go with MqttCallback
 MqttClient client = new MqttClient("tcp://localhost:1883","clientid");
     client.setCallback(this);
MqttConnectOptions mqOptions=new MqttConnectOptions();
     mqOptions.setCleanSession(true);
     client.connect(mqOptions);      //connecting to broker
        client.subscribe("test/topic"); //subscribing to the topic name  test/topic

//Override methods from MqttCallback interface
@Override
public void messageArrived(String topic, MqttMessage message) throws Exception {
        System.out.println("message is : "+message);
    }
.
.//other override methods
```

.

Lea Implementación de MQTT en línea: https://riptutorial.com/es/mqtt/topic/9132/implementacion-de-mqtt

Capítulo 4: Instalación y configuración

Introducción

Implementar MQTT

Necesitamos MQTT Broker y MQTT client Library

Examples

MQTT Bibliotecas y MQTT Broker

> Para usar MQTT en la aplicación, tenemos una variedad de bibliotecas disponibles para diferentes lenguajes de programación.

Biblioteca MQTT

BIBLIOTECA	IDIOMA	DESCRIPCIÓN
Eclipse Paho	C, C ++, Java, Javascript, Python, Go, C #	Los clientes Paho se encuentran entre las implementaciones de bibliotecas de clientes más populares.
Cliente MQTT Fusesource	Java	El cliente MQTT de Fusesource es un cliente MQTT de Java con 3 estilos de API diferentes: Bloqueo, Basado en el futuro y Basado en la devolución de llamada.
MQTT.js	Javascript	MQTT.js es una biblioteca cliente de MQTT para Node.js y aplicaciones web, disponible como un módulo npm.
ruby-mqtt	Rubí	ruby-mqtt es un cliente MQTT disponible como una gema Ruby. No es compatible con QoS> 0.

MQTT Broker

El intermediario es el principal responsable de recibir todos los mensajes (el intermediario es como el servidor de mensajería), filtrarlos, decidir quién está interesado en ellos y luego enviar el mensaje a todos los clientes suscritos. Implementaciones de MQTT Broker: La tabla a continuación muestra algunas de las implementaciones más populares de código abierto y de broker comercial.

Corredor	Descripción
Apache ActiveMQ	ActiveMQ es un intermediario de mensajes multiprotocolo de código abierto con un núcleo escrito alrededor de JMS. Es compatible con MQTT y asigna semántica de MQTT sobre JMS.
mosquitto	
Conejo MQ	RabbitMQ es una implementación de cola de mensajes de código abierto y escalable, escrita en Erlang. Es un agente de mensajes AMQP pero tiene un complemento MQTT disponible. No es compatible con todas las funciones de MQTT (por ejemplo, QoS 2).
HiveMQ	HiveMQ es un agente MQTT escalable y de alto rendimiento adecuado para implementaciones de misión crítica. Es totalmente compatible con MQTT 3.1 y MQTT 3.1.1 y tiene características como websockets, clústeres y un sistema de código abierto para desarrolladores de Java.
WebsphereMQ / IBM MQ	Websphere MQ es un middleware comercial orientado a mensajes de IBM. Totalmente compatible con MQTT.

pasos para instalar el agente ActiveMQ

Vaya al sitio web de ActiveMQ y descargue la última versión estable de activeMQ

Haga clic aquí para descargar activeMQ

- despues de descargar, descomprimirlo

si estas usando windows 32

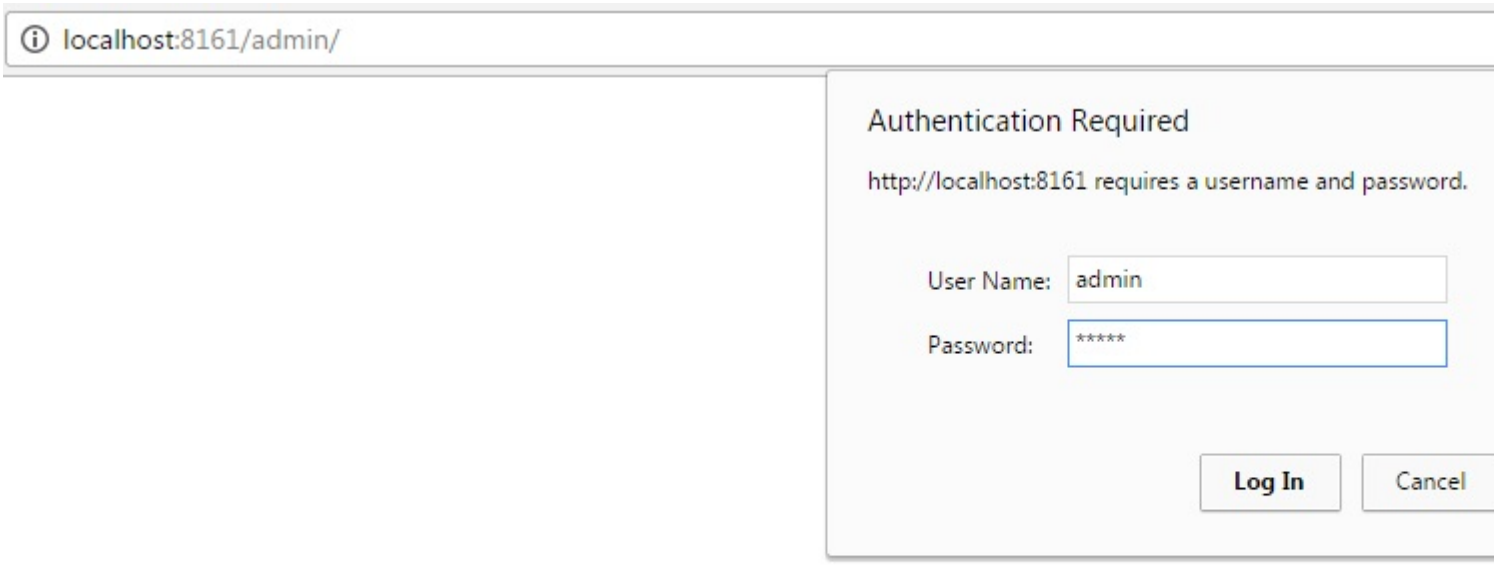
- **Vaya a apache-activemq-5.14.3 \ bin \ win32**

si windows 64

- **apache-activemq-5.14.3 \ bin \ win64**
- ejecutar el archivo por lotes **activemq**
- eso es todo, el servidor activeMQ se está ejecutando en el símbolo del sistema

Si desea ver la consola de interfaz de usuario para activeMQ. Para saber cómo se organizan y envían los mensajes.

llegó a http: // localhost: 8161 / admin /

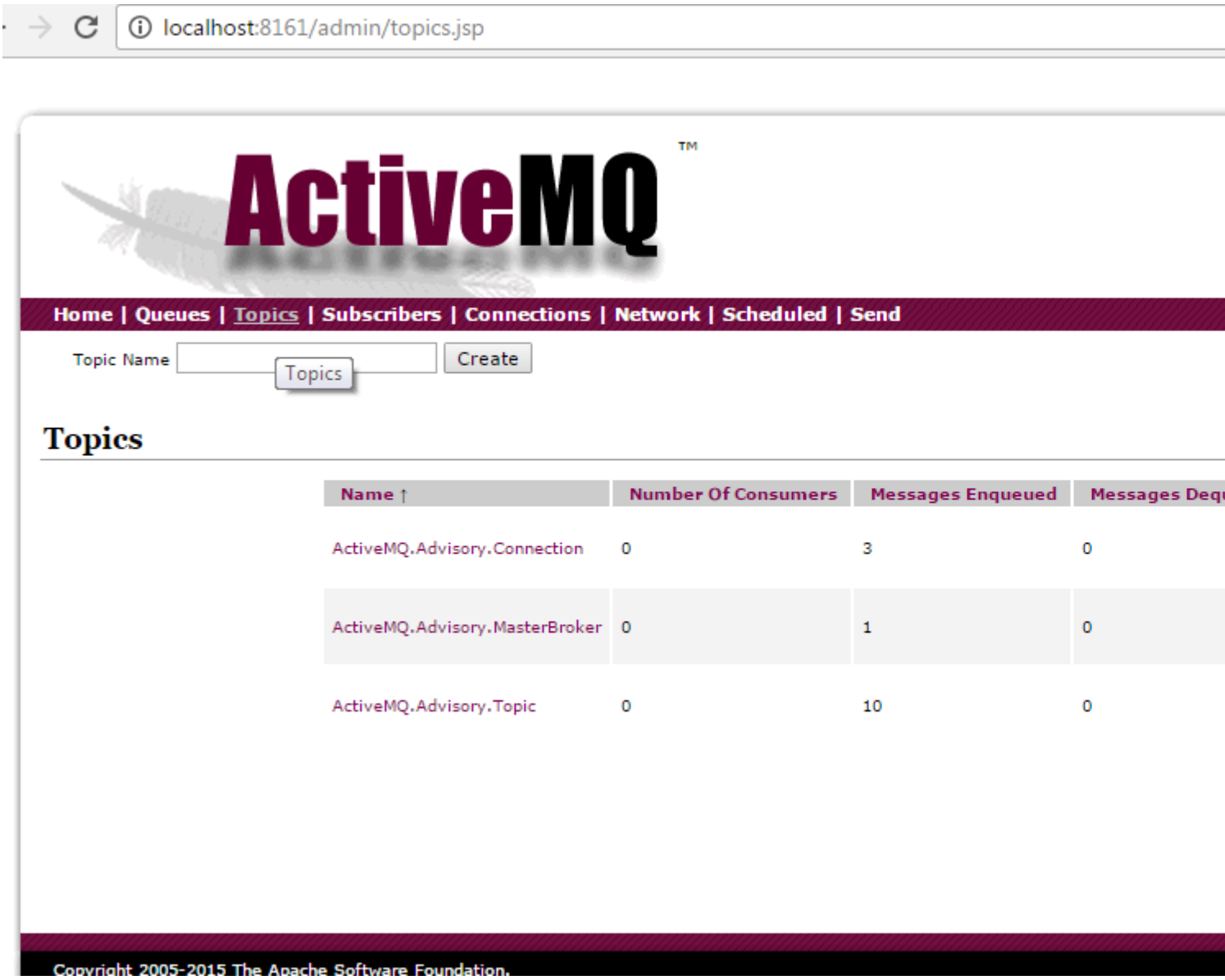


- por defecto

nombre de usuario = admin

contraseña = admin

- a continuación, haga clic en la pestaña tema.



La pestaña Tema proporciona información sobre cuántos temas hay y consumidores activos, productos, mensajes entregados o no.

Lea Instalación y configuración en línea: https://riptutorial.com/es/mqtt/topic/9130/instalacion-y-configuracion

Creditos

S. No	Capítulos	Contributors
1	Empezando con MQTT	Community, ed sheeran, hardillb, Ivo Limmen, Trishant Pahwa
2	Características de MQTT	ed sheeran
3	Implementación de MQTT	ed sheeran
4	Instalación y configuración	ed sheeran