eBook Gratuit

APPRENEZ MQTT

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#mqtt

Table des matières

À propos

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

Chapitre 1: Démarrer avec MQTT

Remarques

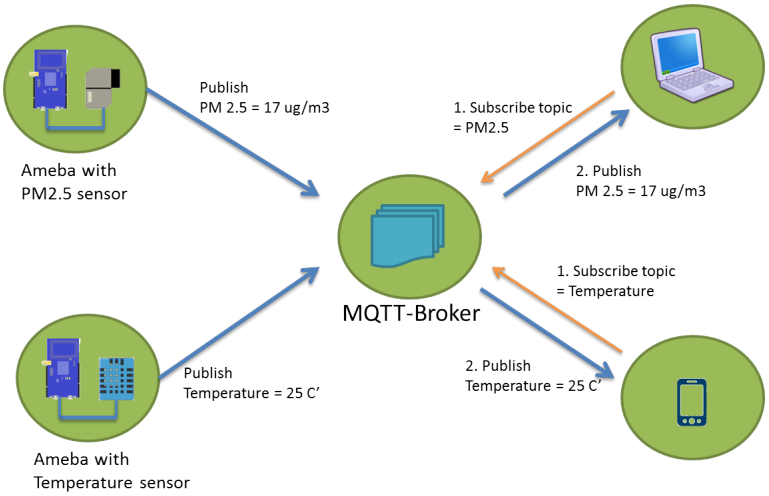
Cette section fournit une vue d'ensemble de ce qu'est mqtt et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans mqtt, et établir un lien avec les sujets connexes. La documentation de mqtt étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Examples

introduction

MQTT (Message Queue Telemetry Transport) est un protocole de messagerie "léger" basé sur Publish-Subscribe à utiliser sur la pile TCP / IP .



> Il est très utile pour les connexions avec des sites distants où une faible empreinte de code est requise et / ou la bande passante du réseau est une priorité.

Il existe de nombreux courtiers et clients différents qui implémentent le protocole MQTT.

Une liste des courtiers, des clients et des outils peut être trouvée sur le site Web de mqtt.org ici , bien qu'elle ne soit pas définitive, elle offre un échantillon représentatif. Il existe également une liste organisée sur github.com sur MQTT.

Des nouvelles sur les spécifications MQTT peuvent être trouvées sur mqtt.org .

MQTT v3.1.1 est une norme Oasis disponible ici

Lire Démarrer avec MQTT en ligne: https://riptutorial.com/fr/mqtt/topic/8187/demarrer-avec-mqtt

Chapitre 2: Caractéristiques de MQTT

Introduction

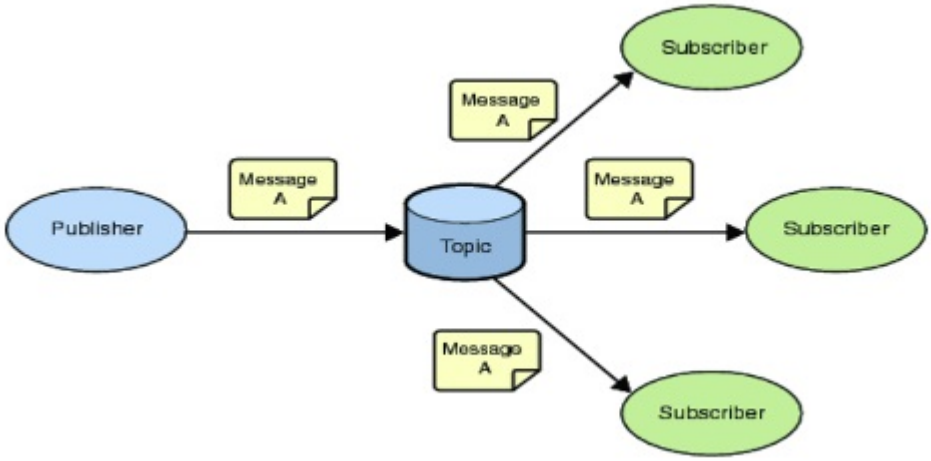
Le protocole s'exécute sur TCP / IP ou sur d'autres protocoles réseau qui fournissent des connexions bidirectionnelles ordonnées et sans perte.

Examples

Modèle public / abonnement simple dans MQTT

Ses principales caractéristiques comprennent:

- Utilisation du modèle de message de publication / abonnement qui fournit une distribution de messages et un découplage un à plusieurs des applications.
- Un transport de messagerie qui est indépendant du contenu de la charge utile. Trois qualités de service pour la livraison des messages
- Un petit surcoût de transport et des échanges de protocoles minimisés pour réduire les tra



Il existe généralement deux types de service de messagerie.

- File d'attente (une à une connexion)
- Sujet (un à un / un à plusieurs)

MQTT ne prend pas en charge la file d'attente qui est fiable, mais MQTT prend en charge la rubrique. Par défaut, Topic n'est pas fiable, mais nous pouvons utiliser les fonctionnalités et méthodes de MQTT pour la rendre fiable.

Différence entre le sujet et la file d'attente

Queue:

- Modèle point à point
- Un seul consommateur reçoit le message
- Les messages doivent être livrés dans la commande envoyée
- Une file d'attente garantit uniquement que chaque message est traité une seule fois.
- La file d'attente sait qui est le client ou le client JMS. La destination est connue.
- Le client JMS (le consommateur) ne doit pas nécessairement être actif ou connecté à la file d'attente pour recevoir ou lire le message.
- Chaque message traité avec succès est reconnu par le consommateur.

Sujet:

- Modèle de publication / abonnement
- Plusieurs clients s'abonnent au message
- Il n'y a pas de garantie les messages doivent être livrés dans la commande envoyée
- Il n'y a aucune garantie que chaque message ne soit traité qu'une seule fois. Comme cela peut être détecté à partir du modèle
- Le sujet, a plusieurs abonnés et il est possible que le sujet ne connaisse pas tous les abonnés. La destination est inconnue
- L'abonné / client doit être actif lorsque les messages sont produits par le producteur, sauf si l'abonnement est un abonnement durable.
- Non, le consommateur / abonné ne reconnaît pas chaque message traité avec succès.

  mais nous pouvons réduire les inconvénients du sujet en utilisant MQTT. Le sujet peut être fiable et contrôler les doublons dans les fonctionnalités MQTT

Lire Caractéristiques de MQTT en ligne: https://riptutorial.com/fr/mqtt/topic/9129/caracteristiques-de-mqtt

Chapitre 3: Installation et configuration

Introduction

Mettre en œuvre MQTT

Nous avons besoin de MQTT Broker et de la bibliothèque client MQTT

Examples

Bibliothèques MQTT et courtier MQTT

> Pour utiliser MQTT dans l'application, nous disposons de plusieurs bibliothèques disponibles pour différents langages de programmation.

Bibliothèque MQTT

BIBLIOTHÈQUE	LA LANGUE	LA DESCRIPTION
Eclipse Paho	C, C ++, Java, Javascript, Python, Go, C #	Les clients Paho figurent parmi les implémentations de bibliothèques client les plus populaires.
Fuse Source MQTT Client	Java	Le client Fusesource MQTT est un client Java MQTT avec 3 styles d'API différents: bloquant, basé sur l'avenir et basé sur le rappel.
MQTT.js	Javascript	MQTT.js est une bibliothèque client MQTT pour Node.js et les applications Web, disponible en tant que module npm.
ruby-mqtt	Rubis	ruby-mqtt est un client MQTT disponible en tant que bijou Ruby. Il ne prend pas en charge QoS> 0.

Courtier MQTT

Le courtier est principalement responsable de la réception de tous les messages (le courtier est comme le serveur de messagerie), de leur filtrage, de leur choix, puis de l'envoi du message à tous les clients abonnés. Implémentations du courtier MQTT: Le tableau ci-dessous présente certaines des implémentations de courtiers open source et commerciaux les plus populaires.

Courtier _____	La description
Apache ActiveMQ	ActiveMQ est un courtier de messages multiprotocole à code source ouvert avec un noyau écrit autour de JMS. Il supporte MQTT et mappe la

Courtier _____	La description
	sémantique MQTT sur JMS.
mosquitto	
Lapin MQ	RabbitMQ est une implémentation évolutive de la file d'attente de messages open source, écrite en Erlang. C'est un courtier de messages AMQP, mais un plug-in MQTT est disponible. Ne prend pas en charge toutes les fonctionnalités MQTT (par exemple, QoS 2).
HiveMQ	HiveMQ est un courtier MQTT évolutif et haute performance adapté aux déploiements stratégiques. Il prend entièrement en charge MQTT 3.1 et MQTT 3.1.1 et dispose de fonctionnalités telles que Websockets, clustering et un système de plug-in open-source pour les développeurs Java.
WebsphereMQ / IBM MQ	Websphere MQ est un middleware commercial IBM. Soutient pleinement MQTT.

étapes pour installer le courtier ActiveMQ

Accédez au site Web ActiveMQ et téléchargez la dernière version stable d'activeMQ

Cliquez ici pour activerMQ téléchargements

- après le téléchargement, décompressez-le

si vous utilisez Windows 32

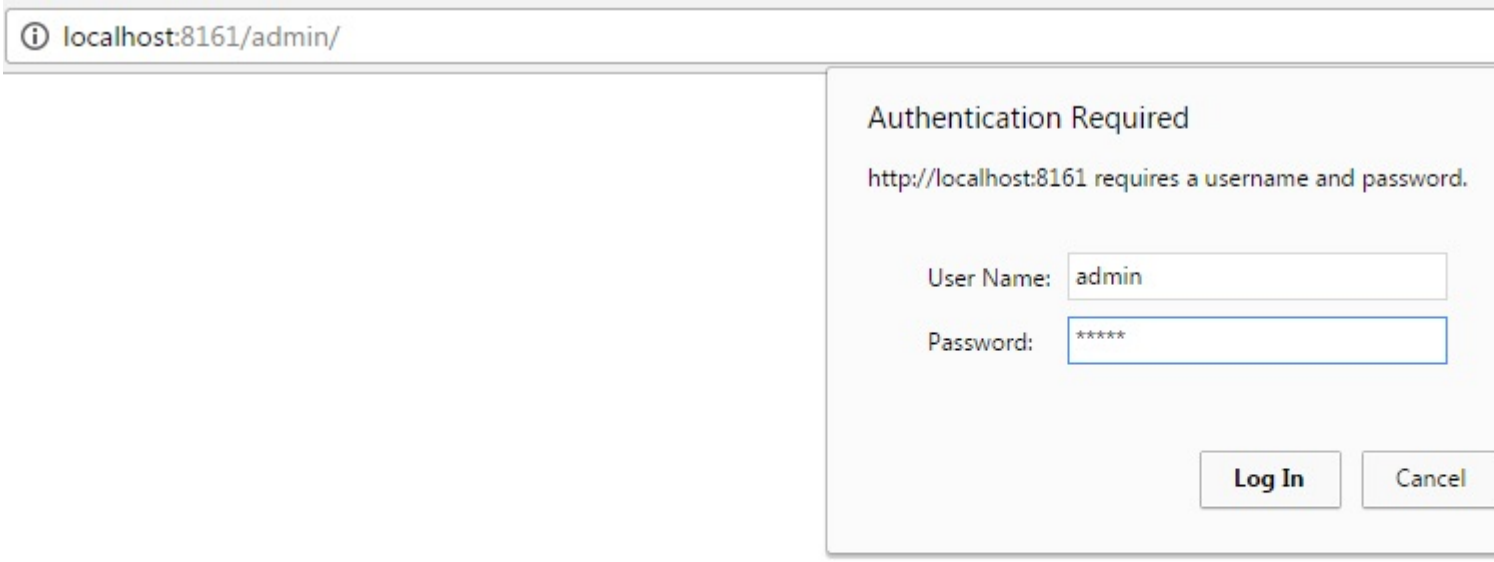
- **Aller à apache-activemq-5.14.3 \ bin \ win32**

si windows 64

- **apache-activemq-5.14.3 \ bin \ win64**
- exécuter le fichier batch **activemq**
- thats it, le serveur activeMQ s'exécute à l'invite de commande

si vous souhaitez voir l'interface utilisateur Consle pour activeMQ. pour savoir comment les messages sont organisés et envoyés

a eu http: // localhost: 8161 / admin /

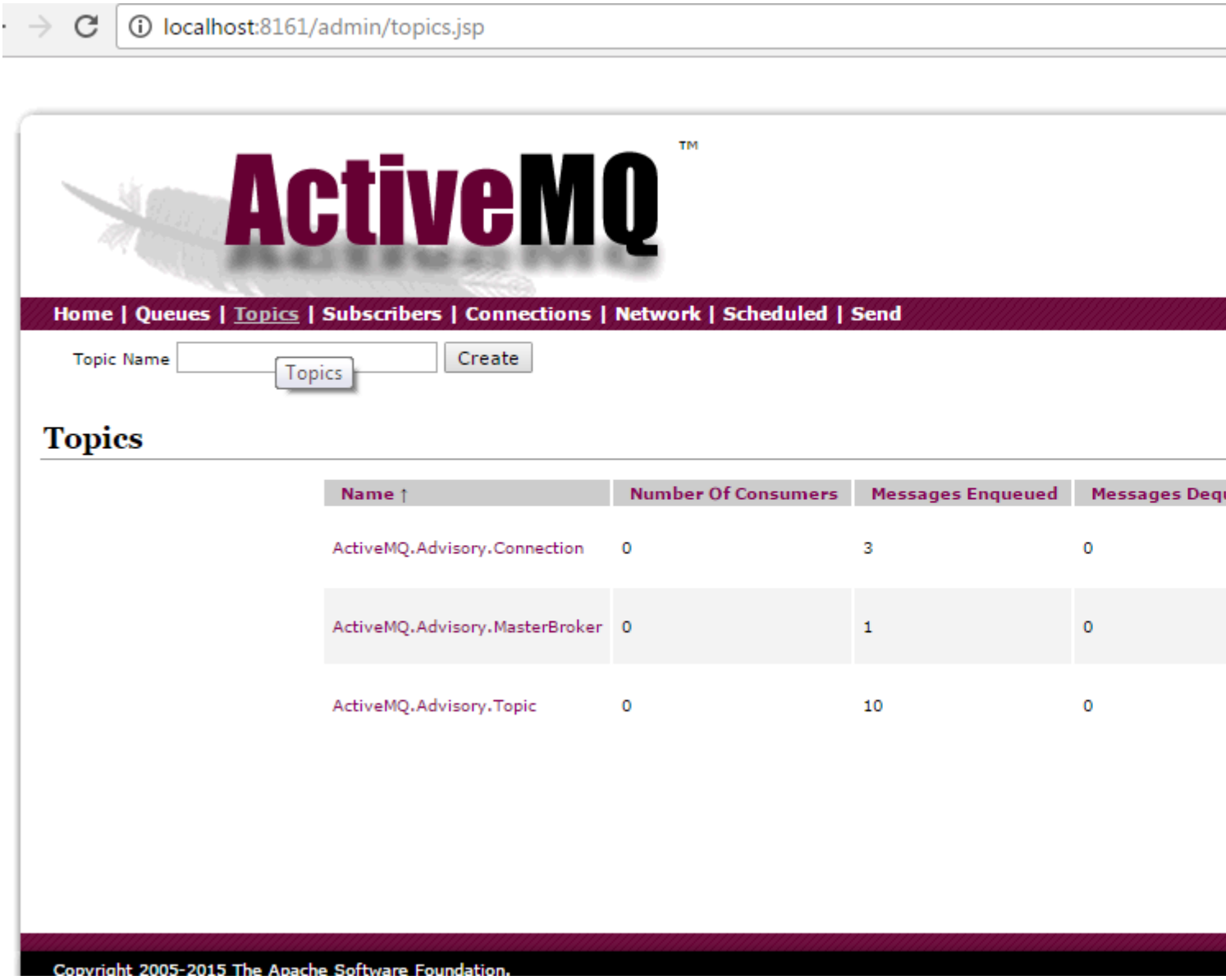


- par défaut

nom d'utilisateur = admin

mot de passe = admin

- puis cliquez sur l'onglet sujet.



L'onglet Sujet donne des informations sur le nombre de sujets et les consommateurs actifs, produits, messages livrés ou non.

Lire Installation et configuration en ligne: https://riptutorial.com/fr/mqtt/topic/9130/installation-et-configuration

Chapitre 4: Mise en place de MQTT

Examples

Exemple de publication / abonné en Java

créer un projet web dynamique dans sts / eclipse télécharger le paho jar eclipse de cliquer ici pour télécharger et coller le fichier jar dans webcontent-> webinf-> folder-> lib

Exemple de publication

```
 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());
```

// ici ed sheeran est un message

```
    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)
```

Exemple d'abonnement

```
//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
```

.

Lire Mise en place de MQTT en ligne: https://riptutorial.com/fr/mqtt/topic/9132/mise-en-place-de-mqtt

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec MQTT	Community, ed sheeran, hardillb, Ivo Limmen, Trishant Pahwa
2	Caractéristiques de MQTT	ed sheeran
3	Installation et configuration	ed sheeran
4	Mise en place de MQTT	ed sheeran