EasyJMqttForIot-V0.0.1 用户文档

1.软件简介

> *EasyJMqttForIot是一个用Java实现的轻量级Mqtt服务器，核心技术用了高的性能的异步网络框架Nwtty，我们把它用在物联网场景中，而且为物联网场景量身定制多个插件，以便于实现更多功能。*
>
> *在以往的项目中，我们做的大部分业务是基于Java来实现的，同时用到了Mqtt，我们尝试了EMQ，Activemq，但是发现要么是编程语言不熟悉，要么是重量级，不适合物联网场景，我们希望能有一个适用于Java业务的Mqtt服务器，于是参考了各种基于Java的框架以后，发起了EasyJMqttForIot项目。旨在为Java程序员提供一个优秀的Iot解决方案。*

2.编译运行

> *本项目核心框架用了Netty，部分功能用了Groovy（WEB接口部分），因此需要安装Groovy环境，关于配置Groovy，请自行网上查阅，这里不做赘述。*
>
> *环境搭建成功以后，安装maven，推荐是maven3。代码下载完毕以后，然后运行以下命令:*
>
> ```
> mvn clear
> mvn packge
> ```
>
> *然后会在target目录生成一个jar，windows下CMD（Linux直接执行）直接运行jar就可以:*
>
> ```
> java -jar 生成的包.jar
> ```
>
> *确认执行成功以后，访问:localhost:2580,就可以看到管理界面，如果看到管理界面就表示运行成功。*

3.测试运行

> *运行成功以后，登陆控制台可以看到一些关键信息，目前0.0.1版只显示节点信息/连接的客户端数，以及CPU负载，这些基础信息足够测试应用，下面给出一个简单的Python客户端用来测试：*
>
> ```
> import paho.mqtt.client as mqtt
> ```

```
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("/test")

def on_message(client, userdata, msg):
    print(msg.topic+" " + ":" + str(msg.payload))

client = mqtt.Client("C1")
client.on_connect = on_connect
client.on_message = on_message
client.connect("127.0.0.1", 1883, 60)

def run():
    client.loop_forever()

import time
import threading
main=threading.Thread(target = run)
main.start()
while 1:
    time.sleep(2)

 client.publish(topic="/test",payload="HelloWorld")
```

注意：本demo基于Python3，需要安装paho库:

```
pip/pip3 install paho-mqtt
```

本Demo运行成功以后会在控制台不断输出下面的信息:

```
C:\Users\wwhai>"E:/Program
Files/Python36/python.exe"
c:/Users/wwhai/Desktop/SDK/SDK.py.py
Connected with result code 0
/test :b'HelloWorld'
/test :b'HelloWorld'
```

到这一步，恭喜你，你已经成功运行起来EasyJMqttForIot了，接下来看看高级操作吧。

EasyJMqttForIot-V0.0.1 开发文档

1.客户端配置

> *注意：这里请务必认真看一下，因为EasyJMqttForIot 某些方面不是标准的Mqtt协议实现，所以有一些坑需要谨慎：*
>
> *1.Mqtt所有的Topic必须以 / (撇斜杠)开头,标准的Mqtt协议是没有限制的，但是我在开发的时候为了方便设计过滤器，设计成这样了，比如你要订阅一个test,标准的写法:test,而EasyJMqttForIot的写法是:/test,具体可以参考使用文档里面的python代码；*
>
> *2.客户端必须提供clientId，因为clientid是认证插件的一部分，所以强制要求必须提供一个clientid，不管是认证插件有没有开启，标准的Mqtt是不需要提供clientid的；*

2.集群配置

> *关于集群：集群用了Ignite实现，本质就是内存逻辑上共享，这里配置相对麻烦，请暂时按照默认配置，不要手动改动。等后期功能完善以后再提供具体文档*

3.压测

> *1.压测工具推荐使用 Apache Jmeter；*
>
> *2.因为Mqtt数据交换全部是在内存中完成的，当客户端的数量比较大的时候，内存会占的很大，建议运行在4G内存以上的服务器。*

4.二次开发

1.开发注意事项

> 1. *遵守规范:代码风格请参考阿里巴巴开发手册*
> 2. *面向接口:建议使用面向接口的形式，不要硬编码*

2. 设计思路

> *Mqtt核心服务参考了Netty的设计哲学，认为每一个客户端都是一个channel，每一个处理都是一个Handler；*
>
> *Web层的设计就是典型的Spring特色，MVC风格。*

3. 二次开发

以下是一个Service层的实现,首先，实现一个标准的CURD模板，然后再对应的控制器里面应用,`Cache.Entry<String, T>`这个内部类表示的是存进Ignite的数据,如下所示:

```
/**
```

```
 * 用户Ignite查询的基础Service
 */
interface BaseIgniteService<T> {
    /**
     * 根据id查找当前Model
     *
     * @param id
     * @return
     */
    T findOneById(long id);
    /**
     * 根据id删除当前Model
     *
     * @param id
     */
    void deleteById(long id);
    /**
     * 批量删除
     *
     * @param ids
     */
    void deleteByIds(long[] ids);
    /**
     * 在数据库存储当前 Model
     *
     * @param T
     */
    void save(T T);
    /**
     * 更新Model
     *
     * @param T
     */
    void update(T T);
    /**
     * 分页查询
     *
     * @param page
     * @param size
     * @return
     */
    List<Cache.Entry<String, T>> listAll(int page, int
size);

}
```

4.数据结构

1.一个在线设备的描述

> *字段解释key:ChannelID，value:channel对应的客户端信息*

```
[{
    "key": "00000000000000e0-000024b8-00000004-
3e59026e2753953e-8630061e",
    "value": {
        "channelId": "00000000000000e0-000024b8-00000004-
3e59026e2753953e-8630061e",
        "channelToJson": {
            "active": true,
            "address": {
                "address": "0:0:0:0:0:0:0:1",
                "port": 8885
            },
            "id": "00000000000000e0-000024b8-00000004-
3e59026e2753953e-8630061e"
        },
        "cleanSession": true,
        "clientId": "0.2808018",
        "willMessageToJson": {
            "will": {
                "willRetain": false,
                "cleanSession": true,
                "willFlag": false
            }
        }
    }
}]
```

> *重点:这里有个设计思路：根据Netty的设计哲学，每一个连接进来的客户端，Netty都认为它是一个Channel，并且给这个Channel一个ID，从而实现标识。EasyMqttServer在设计的过程中参考了这种哲学吗，用ChannelId来表示每一个链接进来的客户端。*

如果有问题或者发现Bug，请及时反馈:751957846@qq.com