

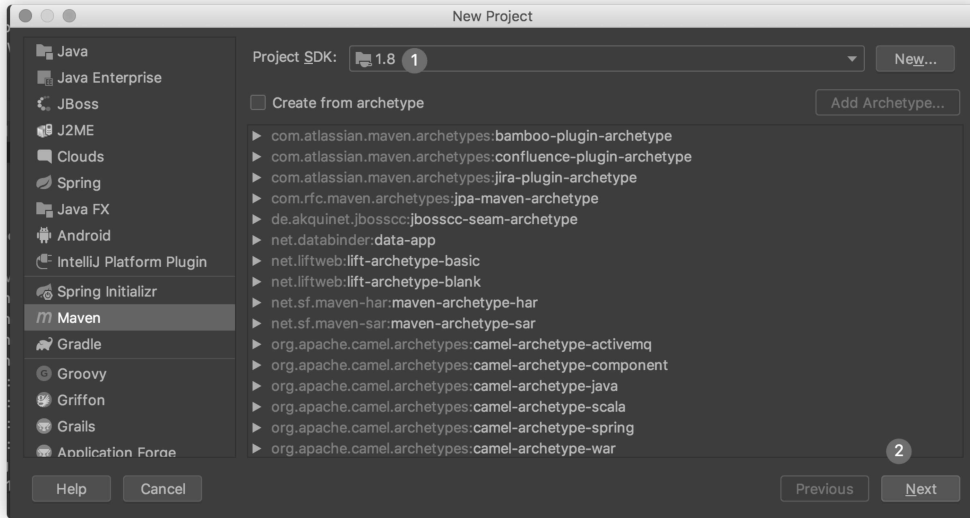
RabbitMQ系列 (三) RabbitMQ進階-Queue隊列特性 (一)簡單隊列

原創 © jzjie007 2020-06-21 06:29

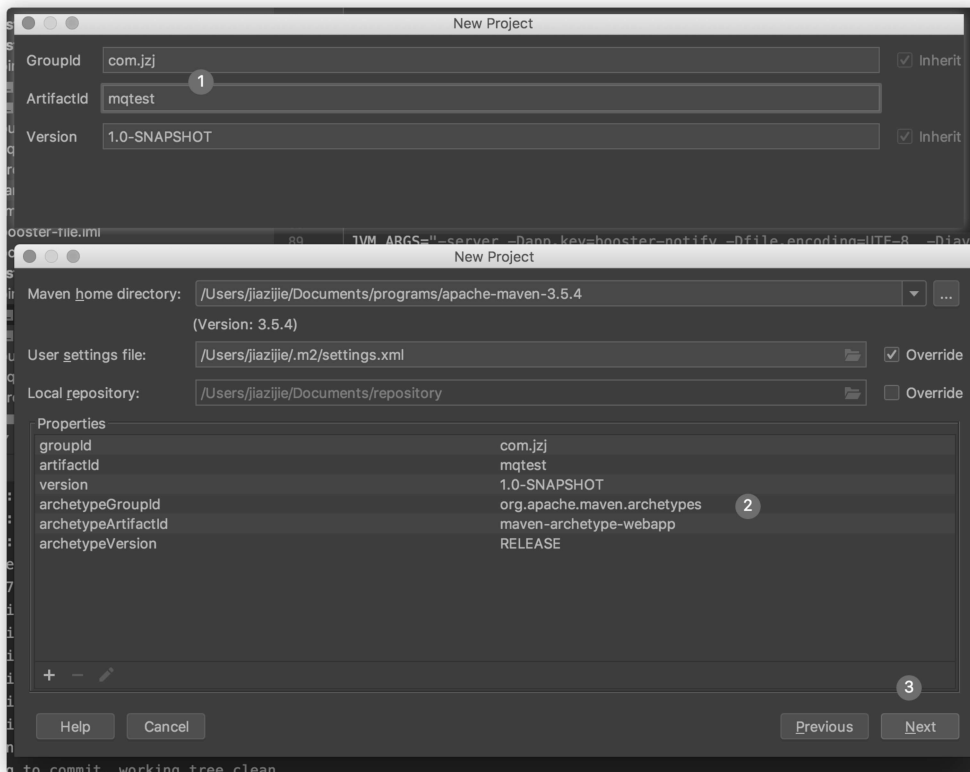
RabbitMQ Spring 項目整合

1.新建mave項目

1.1 新建 maven項目 · 後期咱們再整合spring boot amqp新建spring boot項目 · 路要一步一步走 · 先從簡單的maven項目 · 瞭解原理 · 再從spring boot項目 看spring boot封裝了什麼東西 · 到底簡化了我們的哪些操作



1.2 設置 groupId和artifactId 及maven配置



24小時熱門文章

[來自某男的騷擾 · 沒想到可以給我帶來那麼大恐懼](#)

[放學後女兒哭着走出校門 · Chat GPT寫了封信安慰她 · 卻又讓她紅了眼睛](#)

[230208《一年頂十年》7：通過講課輸出](#)

[快遞小哥與芯片女 \(117\)](#)

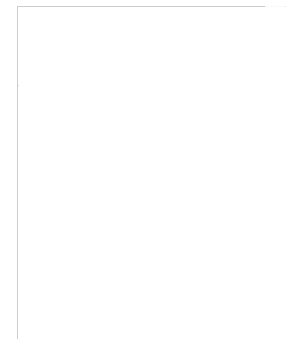
[【柳樹青】早春](#)

[【塵世】中年危機與美麗的能力 中年辭職後關於再就業的一點思考](#)

[2.4 當上級兩面三刀 · 我們怎麼辦？](#)

[02-06 如何做到忙而不亂？](#)

[是誰這麼缺德](#)



最新文章

[RabbitMQ系列 \(三\) RabbitMQ進階-Queue隊列特性 \(一\)簡單隊列](#)

[RabbitMQ系列 \(四\) RabbitMQ進階-Queue隊列特性 \(二\)工作隊列 Work模式](#)

[RabbitMQ系列 \(一\) 啟動及網頁、外網訪問配置](#)

[RabbitMQ系列 \(二\) VirtualHost作用及角色權限管理實戰](#)

[RabbitMQ系列-簡介](#)

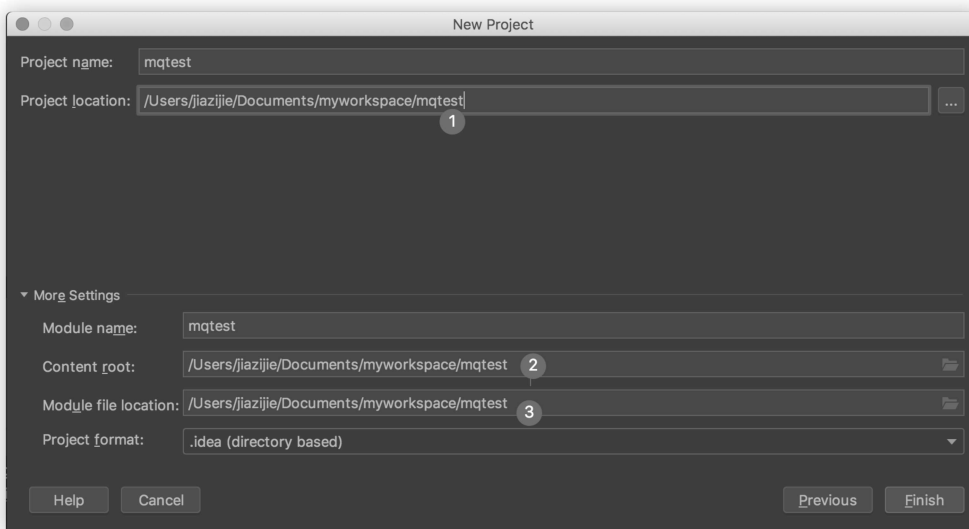
最新評論文章

[linux以太網驅動總結](#)

[【python 圖片搜索】python 快速計算兩個圖片的相似度](#)



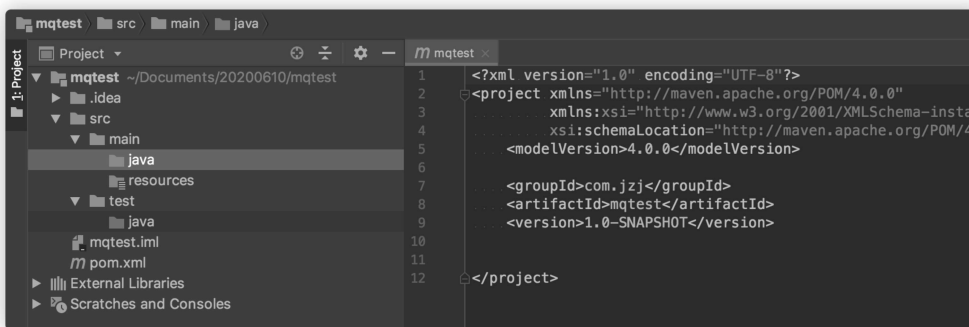
3 選擇項目文件夾



洋甘菊極致精

NT\$1,1

1.4 項目新建完成 · 目錄結構如下



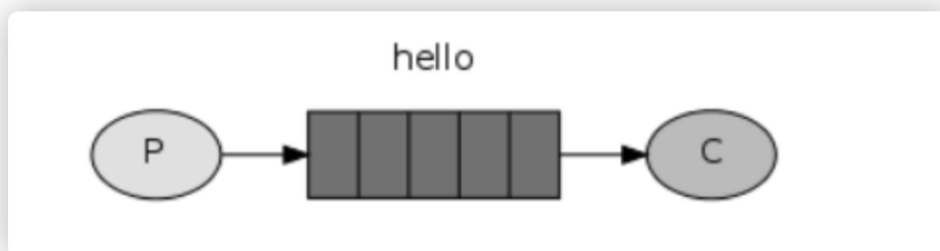
1.5 pom中添加 RabbitMQ 客戶端依賴

```
<dependency>
  <groupId>com.rabbitmq</groupId>
  <artifactId>amqp-client</artifactId>
  <version>3.4.1</version>
</dependency>
```

RabbitMQ 五種 Queue 隊列

1. 簡單隊列

簡單隊列就是 1對1 · 1個生產者對應1個消費者



簡單隊列上代碼

1.1 connect 工具類

```

package com.jzj.mq.conn;

import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

/**
 * 當前描述:
 *
 * @author: jiazijie
 * @since: 2020/6/10 下午11:12
 */
public class MqConnectUtil {
    public static Connection getConnectionDefault() throws Exception {
        //定義連接工廠
        ConnectionFactory factory = new ConnectionFactory();
        //設置服務地址
        factory.setHost("127.0.0.1");
        //端口
        factory.setPort(5672);
        //設置賬號信息, 用戶名、密碼、vhost
        factory.setVirtualHost("/");
        factory.setUsername("admin");
        factory.setPassword("admin");
        // 通過工程獲取連接
        Connection connection = factory.newConnection();
        return connection;
    }

    public static Connection getConnectionJzj() throws Exception {
        //定義連接工廠
        ConnectionFactory factory = new ConnectionFactory();
        //設置服務地址
        factory.setHost("127.0.0.1");
        //端口
        factory.setPort(5672);
        //設置賬號信息, 用戶名、密碼、vhost
        factory.setVirtualHost("jzj");
        factory.setUsername("admin");
        factory.setPassword("admin");
        // 通過工程獲取連接
        Connection connection = factory.newConnection();
        return connection;
    }
}

```

1.2 Producer 生產者-消息產生->發送到交換機->轉發到隊列存着

```

package com.jzj.mq.produce;

import com.jzj.mq.conn.MqConnectUtil;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;

/**
 * 當前描述: 生產者
 *
 * @author: jiazijie
 * @since: 2020/6/10 下午11:14
 */
public class Producer {
    private final static String QUEUE_NAME = "queue_test";

    public static void main(String[] argv) throws Exception {
        // 獲取到連接以及mq通道
        Connection connection = MqConnectUtil.getConnectionDefault();
        // 從連接中創建通道
        Channel channel = connection.createChannel();

        /* 聲明 (創建) 隊列 queueDeclare( String queue, boolean durable, boolean exclusive,
        boolean autoDelete, Map<String, Object> arguments)
        * queue - 隊列名
        * durable - 是否是持久化隊列, 隊列的聲明默認是存放到內存中的, 如果rabbitmq重啓會丟失
        * exclusie - 是否排外的, 僅限於當前隊列使用
        * autoDelete - 是否自動刪除隊列, 當最後一個消費者斷開連接之後隊列是否自動被刪除, 可以通過界面
        查看某個隊列的消費者數量, 當consumers = 0時隊列就會自動刪除
        * arguments - 隊列攜帶的參數 比如 ttl-生命週期, x-dead-letter 死信隊列等等
        */
        channel.queueDeclare(QUEUE_NAME, false, false, false, null);

        // 消息內容
        String message = "Hello World!";

        /* 發送消息 String exchange, String routingKey, BasicProperties props, byte[] body
        * exchange - 交換機, "" 空時候指定的是 獲取的virtualHost 虛擬服務器的 默認的exchange, 每個
        virtualHost都有一個AMQP default type:direct 直接轉發
        * queuename - 隊列信息
        * props - 參數信息
        * message 消息體 byte[]類型
        */
        channel.basicPublish("", QUEUE_NAME, null, message.getBytes());

        System.out.println("**** Producer Sent Message: '" + message + "'");
        //關閉通道和連接
        channel.close();
        connection.close();
    }
}

```

1.3 消費者->監聽隊列·有消息就從隊列中取出來消息進行消費

```
package com.jzj.mq.consumer;

import com.jzj.mq.conn.MqConnectUtil;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.QueueingConsumer;

/**
 * 當前描述: 消費者
 *
 * @author: jiazijie
 * @since: 2020/6/10 下午11:30
 */
public class Consumer {
    private final static String QUEUE_NAME = "queue_test";

    public static void main(String[] argv) throws Exception {
        Connection connection = MqConnectUtil.getConnectionDefault();
        Channel channel = connection.createChannel();

        /*確保這裏的隊列是存在的*/
        channel.queueDeclare(QUEUE_NAME, false, false, false, null);
        System.out.println(" **** Consumer Waiting for messages. To exit press CTRL+C");

        QueueingConsumer consumer = new QueueingConsumer(channel);

        /* 消息確認機制
         * autoAck true:表示自動確認，只要消息從隊列中獲取，無論消費者獲取到消息後是否成功消費，都會認為消息已經成功消費
         * autoAck false:表示手動確認，消費者獲取消息後，服務器會將該消息標記為不可用狀態，等待消費者的反饋，如果消費者一直沒有反饋，那麼該消息將一直處於不可用狀態
         * 並且服務器會認為該消費者已經掛掉，不會再給其發送消息，直到該消費者反饋
         */
        channel.basicConsume(QUEUE_NAME, true, consumer);

        while (true) {
            QueueingConsumer.Delivery delivery = consumer.nextDelivery();
            String message = new String(delivery.getBody());
            System.out.println(" **** Consumer Received '" + message + "'");
        }
    }
}
```

1.4 執行結果：

在這裏插入圖片描述

可以看到 · Producer生產者生產一條消息 · Consumer 就能夠消費一條消息

!!!!!! 注意 我們在Producer中並沒有指定 exchange · 傳入的exchange也是字符串 · 它使用的是每一個VirtualHost 下面默認的AMQP default exchange

如下圖：

在這裏插入圖片描述

下一篇 我們介紹 工作隊列 work模式



Java Web SprintBoot MQ中間件 java rabbitmq queue spring

發表評論

登錄以後才評論...

登錄

所有評論

還沒有人評論，想成為第一個評論的人麼？請在上方評論欄輸入並且點擊發布。

相關文章

springbootwebsocket

spring.websocket簡單實踐 Spring提供的類和接口 spring 定義了一些接口和抽象類，我們只要實現這些接口和抽象類，就能夠完成一個目錄比較清晰的 WebSocket的服務器端，並且它裏面有很多封裝的功能，我們可以直接

🕒 原創 🕒 2023-02-02 01:32:31

Camunda簡介及開源協議

Camunda (Github · 官網)是一個工作流引擎。目前提供二個主線版本7.X 與 8.X，二個主線版本架構上並不相同。7.X與Activiti, Flowable 等開源工作流類似，以數據庫為基礎，迭代完善的BPMN,DMN等功

🕒 原創 🕒 2023-02-01 23:36:06

O2OA(翻途)開發平臺服務器下載及安裝部署 For WindowsServer

O2OA(翻途)開發平臺[下稱O2OA開發平臺或者O2OA]支持公有云、私有云和混合雲部署，也支持複雜的網絡結構下的分佈式部署。O2OA(翻途)開發平臺安裝部署非常方便，只需要簡單的三步即可完成安裝。平臺內部集成了多項管理命令，可以

🕒 原創 🕒 2023-02-09 22:34:19

Solon 的插件熱插拔管理機制 (H-Spi)



插件熱插拔管理機制，簡稱：H-Spi，是框架提供的生產時用的另一種高級擴展方案。相對E-Spi、H-Spi更側重隔離、熱插熱拔、及管理性。應用時，是以一個業務模塊為單位進行開發，且封裝為一個獨立插件包。1、特點說明
所有插件包獨享C

🕒 原創 🕒 2023-02-08 21:13:43

R2M分佈式鎖原理及實踐

作者：京東科技 張石磊 1 案例簡介：資源：可以理解為一條內容，或者圖+文字+鏈接的載體。檔位ID：資源的分類組；資源必須歸屬於檔位。問題描述：當同一個檔位下2條資源同時審批通過時，收到擎天審批系統2條消息，消費者應用部

🕒 原創 🕒 2023-02-07 23:35:19

Java單元測試技巧之JSON序列化

前言《論語》中孔子有言：“工欲善其事，必先利其器。”今年7月，作者希望迎接更大的挑戰，從高德地圖數據轉崗到共享出行後，接手並維護了幾個Java後端項目。在熟悉業務和代碼的過程中，快速地對原有項目進行單元測試用例的補充，使其單元測試覆蓋

🕒 原創 🕒 2023-02-06 12:41:30

那些年，我們寫過的無效單元測試

前言 那些年，爲了學分，我們學會了面向過程編程；那些年，爲了就業，我們學會了面向對象編程；那些年，爲了生活，我們學會了面向工資編程；那些年，爲了升職加薪，我們學會了面向領導編程；

🕒 原創 🕒 2023-02-06 11:59:24

Java編程技巧之單元測試用例簡化方法

前言 清代譚實小說家吳趼人在《痛史》中寫道：“卷帙浩繁，望而生畏。”意思是：“一部書的篇幅太長，讓人看見就害怕。”編寫單元測試用例也是如此，如果單元測試用例寫起來又長又複雜，自然而然地會讓人“望而生畏”，於是開始反感甚至於最終放棄。爲

🕒 原創 🕒 2023-02-04 00:00:52

我的新書《高性能Java系統權威指南》

各大書店正在銷售，停止內卷，學點實用的 本書代碼最新地址 <https://gitee.com/xiandafu/java-performance> 每章的readme.md 包含了主要代碼的說明 節選 書中章節 第1章 Java代

🕒 原創 🕒 2023-02-03 01:28:34

springwebsocket簡單demo

基於註解的websocket WebSocket 通信流程 ws:// 開頭 Upgrade:websocket Connection:upgrade 狀態碼：101協議 變更 websocket沒有同源限制。1. 服務終端類: 用java

🕒 原創 🕒 2023-02-01 13:28:31

java: JPS incremental annotation processing is disabled

在setting_compile中修改大小

🕒 原創 🕒 2023-01-19 01:17:57

Java泛型01：基礎知識

1. 泛型程序設計 泛型是Java程序設計中一個重要的思想，它可以被用在類、接口、方法中。泛型簡單來說就是：1) 所編寫的代碼在不用修改的前提下，可以被多種不同類型的對象所重用。2) 相較於雜亂的使用Object變量，泛型機制編寫的程序具

🕒 原創 🕒 2023-01-14 22:54:03

Java泛型02：泛型和虛擬機（類型擦除）

Java虛擬機（JVM，Java Virtual Machine）中並不存在泛型，Java語言中的泛型只在程序源碼中存在，在編譯後的字節碼文件（Class文件）中，全部泛型都被替換爲原始類型，並且在相應的地方插入了強制轉型代碼以及

👍 ④ 原創 ⌚ 2023-01-14 22:54:02



記錄一次還算優雅的代碼設計

作者：京東零售 常文標 商卡聚合服務是一個小巧的rpc應用，功能是統一查詢商品的促銷、自營包郵、價格信息、區域庫存、區域可配送等等利益點或其他信息。本文重點分享商卡聚合服務的代碼設計，包括合理的Sirector線程調度（cpu使用率低），和

④ 原創 ⌚ 2023-02-09 12:13:58

Redis實戰之session共享

當線上集羣時候，會出現session共享問題。雖然Tomcat提供了session copy的功能，但是缺點比較明顯：1：當Tomcat多的時候，session需要大量同步到多臺集羣上，佔用內網寬帶 2：同一個用戶session，需要在

④ 原創 ⌚ 2023-02-06 22:52:33
