

# 努力+坚持，而且还很年轻

梦在翱翔，心在路上。关于学习，贵在坚持。

目录视图 摘要视图 RSS 订阅

请作者喝杯咖啡！

如果觉得文章不错，请作者喝杯咖啡，谢谢！

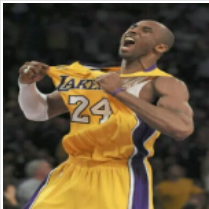
微信扫一扫 支付



向标是我的小号,小平蛋(\*\*\*亮) 转账

¥2.00

个人资料



于亮



访问： 1988348次  
积分： 18162  
等级： BLOG > 7  
排名： 第388名

原创： 175篇  
转载： 6篇  
译文： 2篇  
评论： 3025条

博客专栏



zookeeper边  
边实践  
文章： 5篇  
阅读： 8105



深入浅出Maven  
文章： 4篇  
阅读： 74261



数据库  
文章： 9篇  
阅读： 89902

## 深入浅出JMS(三)--ActiveMQ简单的HelloWorld实例

标签： 实例 helloworld jms 消息中间件 activeMQ

2015-09-26 13:37 30719人阅读 评论(34) 收藏 举报

分类：

【ActiveMQ】 (3) ▼

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?) [+]

第一篇博文[深入浅出JMS\(一\)-JMS基本概念](#)，我们介绍了JMS的两种消息模型：点对点和发布订阅模型，以及消息被消费的两个方式：同步和异步，JMS编程模型的对象，最后说了JMS的优点。

第二篇博文[深入浅出JMS\(二\)-ActiveMQ简单介绍以及安装](#)，我们介绍了消息中间件ActiveMQ，安装，启动，以及优缺点。

这篇博文，我们使用ActiveMQ为大家实现一种点对点的消息模型。如果你对点对点模型的认识较浅，可以看一下第一篇博文的介绍。

JMS其实并没有想象的那么高大上，看完这篇博文之后，你就知道什么叫简单，下面直接进入主题

### 开发环境

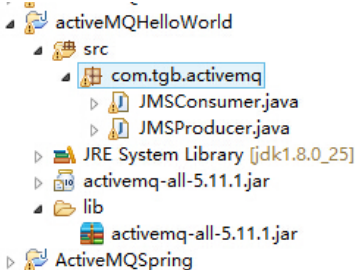
我们使用的是ActiveMQ 5.11.1 Release的Windows版，官网最新版是ActiveMQ 5.12.0 Release，大家可以自行下载，[下载地址](#)。

需要注意的是，开发时候，要将apache-activemq-5.11.1-bin.zip解压缩后里面的activemq-all-5.11.1.jar包加入到classpath下面，这个包包含了所有jms接口api的实现。

### 搭建开发环境

- 建立项目

我们只需要建立一个java项目就可以了，导入jar包，项目截图：



点对点的消息模型，只需要一个消息生成者和消息消费者，下面我们编写代码。

- 编写生产者

```
1  
2 package com.tgb.activemq;  
3  
4 import javax.jms.Connection;
```

SSH

ssh进阶之路  
文章: 33篇  
阅读: 609646



Java读书笔记  
+项目实战  
文章: 25篇  
阅读: 427247

软考之路

软考之路  
文章: 9篇  
阅读: 46084



jQuery EasyUI  
文章: 11篇  
阅读: 185434

UML

UML学习笔记详  
解  
文章: 8篇  
阅读: 306475

发送邮件



文章分类

**【Git】** (1)  
**【.NET】** (13)  
**【Maven】** (4)  
**【Scrum】** (3)  
**【Struts】** (7)  
**【Spring】** (12)  
**【Hibernate】** (10)  
**【SpringMVC】** (2)  
**【Mybatis】** (3)  
**【SSH架构设计】** (28)  
**【Jquery+EasyUI】** (9)  
**【JavaScript】** (5)  
**【MongoDB】** (2)  
**【ActiveMQ】** (4)  
**【WebService】** (2)  
**【Mule ESB】** (2)  
**【ElasticSearch】** (3)  
**【ZooKeeper】** (5)  
**【数据库】** (9)  
**【Tomcat】** (2)  
**【UML建模】** (8)  
**【java实战】** (27)  
**【设计模式】** (6)  
**【算法和数据结构】** (9)  
**【软件工程】** (5)  
**【成长之路】** (13)  
**【编程学习方法】** (11)  
**【Short Brain】** (2)  
**【自考感悟】** (4)  
**【项目总结】** (3)  
**【时间管理】** (1)  
**【提高班】** (3)  
**【编程的快乐】** (4)  
**【VB基础教程】** (14)  
**【Flex】** (2)  
**【微服务】** (2)  
**【NIO】** (1)

```

5  import javax.jms.ConnectionFactory;
6  import javax.jms.Destination;
7  import javax.jms.JMSException;
8  import javax.jms.MessageProducer;
9  import javax.jms.Session;
10 import javax.jms.TextMessage;
11
12 import org.apache.activemq.ActiveMQConnection;
13 import org.apache.activemq.ActiveMQConnectionFactory;
14 /**
15  * 消息的生产者（发送者）
16  * @author liang
17  *
18  */
19 public class JMSProducer {
20
21     //默认连接用户名
22     private static final String USERNAME = ActiveMQConnection.DEFAULT_USER;
23     //默认连接密码
24     private static final String PASSWORD = ActiveMQConnection.DEFAULT_PASSWORD;
25     //默认连接地址
26     private static final String BROKEURL = ActiveMQConnection.DEFAULT_BROKER_URL;
27     //发送的消息数量
28     private static final int SENDNUM = 10;
29
30     public static void main(String[] args) {
31         //连接工厂
32         ConnectionFactory connectionFactory;
33         //连接
34         Connection connection = null;
35         //会话 接受或者发送消息的线程
36         Session session;
37         //消息的目的地
38         Destination destination;
39         //消息生产者
40         MessageProducer messageProducer;
41         //实例化连接工厂
42         connectionFactory = new ActiveMQConnectionFactory(JMSProducer.USERNAM
43
44         try {
45             //通过连接工厂获取连接
46             connection = connectionFactory.createConnection();
47             //启动连接
48             connection.start();
49             //创建session
50             session = connection.createSession(true, Session.AUTO_ACKNOWLEDGE);
51             //创建一个名称为HelloWorld的消息队列
52             destination = session.createQueue("HelloWorld");
53             //创建消息生产者
54             messageProducer = session.createProducer(destination);
55             //发送消息
56             sendMessage(session, messageProducer);
57
58             session.commit();
59
60         } catch (Exception e) {
61             e.printStackTrace();
62         } finally {
63             if (connection != null) {
64                 try {
65                     connection.close();
66                 } catch (JMSException e) {
67                     e.printStackTrace();
68                 }
69             }
70         }
71     }
72 }
73 /**
74  * 发送消息
75  * @param session
  
```

阅读排行

浅谈UML的概念和模型之

(218905)

【SSH进阶之路】Hiberr

(120589)

【持久化框架】SpringM

(97475)

【持久化框架】Mybatis

(88195)

深入浅出JMS(一)--JMS基

(69730)

详解SQL Server连接（

(54763)

【java项目实战】一步步

(48840)

深入浅出JMS(二)--Active

(47614)

【项目管理和构建】——

(45631)

深入浅出JMS(四)--Spring

(40340)

文章存档

2017年02月 (5)

2016年09月 (1)

2016年08月 (2)

2016年04月 (4)

2015年10月 (4)

展开

评论排行

万里长征，始于足下——

(102)

【持久化框架】SpringM

(87)

基于asp.net + easyui框

(68)

深入浅出JMS(四)--Spring

(68)

我的软考之路（三）——

(55)

【SSH进阶之路】Hiberr

(52)

我的软考之路（一）——

(50)

我的软考之路（四）——

(46)

浅谈UML的概念和模型之

(45)

详解SQL Server连接（

(43)

最新评论

【SSH进阶之路】Hibernate基本

栗振娟: 写的真好，学习了。

【SpringMVC架构】SpringMVC

happydream\_C: 博主，跪求

SpringMVC的下篇博文!!! 不

完整啊

Zookeeper的java客户端API使用

盖丽男: 还没用过zookeeper呢

【SSH进阶之路】一步步重构容

qq\_35784766: 学习了

基于asp.net + easyui框架，一步

xiangcns: 写得好，学习一下

【持久化框架】SpringMVC+Spr

qq\_16677263:

@jiuqiuliang:。。。大神，没解

决了，我那个是tomcat就启动不

起来，不用参数

【持久化框架】SpringMVC+Spr

于亮: @qq\_16677263:但愿这篇

文章能帮助你:

http://bbs.csdn.net/topics...

【持久化框架】SpringMVC+Spr

于亮: @qq\_16677263:不知道是

不是你的注解使用的有问题

【持久化框架】SpringMVC+Spr

qq\_16677263: @qq\_16677263:

我用 junit测试的时候 可以获取到

dao的对象

【持久化框架】SpringMVC+Spr

76

\* @param messageProducer 消息生产者

77

\* @throws Exception

78

\*/

79

public static void sendMessage(Session session, MessageProducer messageProducer) throws

80

for (int i = 0; i < JMSPProducer.SENDNUM; i++) {

81

//创建一条文本消息

82

TextMessage message = session.createTextMessage("ActiveMQ 发送消息" + i);

83

System.out.println("发送消息: Activemq 发送消息" + i);

84

//通过消息生产者发出消息

85

messageProducer.send(message);

86

}

87

}

88

}

89

}

• 编写消费者

1

package com.tgb.activemq;

2

3

4

import javax.jms.Connection;

5

import javax.jms.ConnectionFactory;

6

import javax.jms.Destination;

7

import javax.jms.JMSException;

8

import javax.jms.MessageConsumer;

9

import javax.jms.Session;

10

import javax.jms.TextMessage;

11

12

import org.apache.activemq.ActiveMQConnection;

13

import org.apache.activemq.ActiveMQConnectionFactory;

14

/\*\*

15

\* 消息的消费者（接受者）

16

\* @author liang

17

\*

18

\*/

19

public class JMSConsumer {

20

21

private static final String USERNAME = ActiveMQConnection.DEFAULT\_USER;//用户名

22

private static final String PASSWORD = ActiveMQConnection.DEFAULT\_PASSWORD;//密码

23

private static final String BROKEURL = ActiveMQConnection.DEFAULT\_BROKER\_URL;//默认连

24

25

public static void main(String[] args) {

26

ConnectionFactory connectionFactory;//连接工厂

27

Connection connection = null;//连接

28

29

Session session;//会话 接受或者发送消息的线程

30

Destination destination;//消息的目的地

31

32

MessageConsumer messageConsumer;//消息的消费者

33

34

//实例化连接工厂

35

connectionFactory = new ActiveMQConnectionFactory(JMSConsumer.USERNAME, JMSConsumer

36

37

try {

38

//通过连接工厂获取连接

39

connection = connectionFactory.createConnection();

40

//启动连接

41

connection.start();

42

//创建session

43

session = connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);

44

//创建一个连接HelloWorld的消息队列

45

destination = session.createQueue("HelloWorld");

46

//创建消息消费者

47

messageConsumer = session.createConsumer(destination);

48

49

while (true) {

50

TextMessage textMessage = (TextMessage) messageConsumer.receive(100000);

51

if(textMessage != null){

52

System.out.println("收到的消息:" + textMessage.getText());

qq\_16677263: dao接口不能实例化是什么原因? No qualifying bean of type 'com.sp...

```
53         }else {
54             break;
55         }
56     }
57
58
59     } catch (JMSException e) {
60         e.printStackTrace();
61     }
62
63     }
64 }
```

运行

- 1. 首先，启动ActiveMQ，如何启动ActiveMQ如何启动，请看第二篇博文。在浏览器中输入：<http://localhost:8161/admin/>，然后开始执行：
- 2. 运行发送者，eclipse控制台输出，如下图：

```
<terminated> JMSProducer (2) [Java Application] D:\jdk1.8.0_25\bin\javaw.exe (2015年9月20E
log4j:WARN No appenders could be found for lo
log4j:WARN Please initialize the log4j system
log4j:WARN See http://logging.apache.org/log4
发送消息: Activemq 发送消息0
发送消息: Activemq 发送消息1
发送消息: Activemq 发送消息2
发送消息: Activemq 发送消息3
发送消息: Activemq 发送消息4
发送消息: Activemq 发送消息5
发送消息: Activemq 发送消息6
发送消息: Activemq 发送消息7
发送消息: Activemq 发送消息8
发送消息: Activemq 发送消息9
```

生产者生产10条消息

此时，我们先看一下ActiveMQ服务器，Queues内容如下：

Home
Queues
Topics
Subscribers
Connections
Network
Scheduled
Send

Queue Name
Create

点对点消息

Queues

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
HelloWorld	10	0	10	0	<div> Browse Active Consumers Active Producers </div>	Send To Purge Delete

生产了10条消息

有10条消息在排队，被消费

查看是哪10条消息

我们可以看到创建了一个名称为HelloWorld的消息队列，队列中有10条消息未被消费，我们也可以通过Browse查看是哪些消息，如下图：

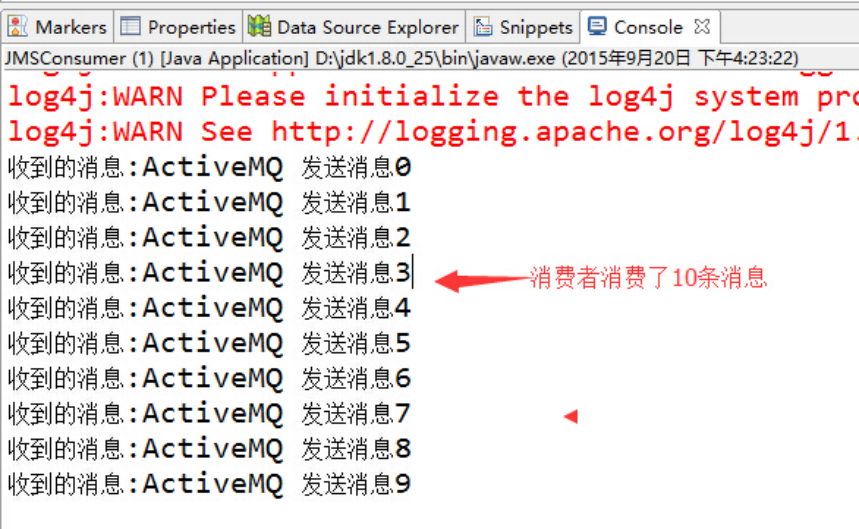
Browse HelloWorld

Message ID ↑	Correlation ID	Persistence	Priority	Redelivered	Reply To	Timestamp	Type	Operations
ID:lenovo-pc-2326-1442742107633-1:1:1:1:1		Persistent	4	false		2015-09-20 17:41:47:886 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:10		Persistent	4	false		2015-09-20 17:41:47:892 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:2		Persistent	4	false		2015-09-20 17:41:47:888 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:3		Persistent	4	false		2015-09-20 17:41:47:889 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:4		Persistent	4	false		2015-09-20 17:41:47:889 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:5		Persistent	4	false		2015-09-20 17:41:47:890 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:6		Persistent	4	false		2015-09-20 17:41:47:890 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:7		Persistent	4	false		2015-09-20 17:41:47:891 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:8		Persistent	4	false		2015-09-20 17:41:47:891 CST		Delete
ID:lenovo-pc-2326-1442742107633-1:1:1:1:9		Persistent	4	false		2015-09-20 17:41:47:891 CST		Delete

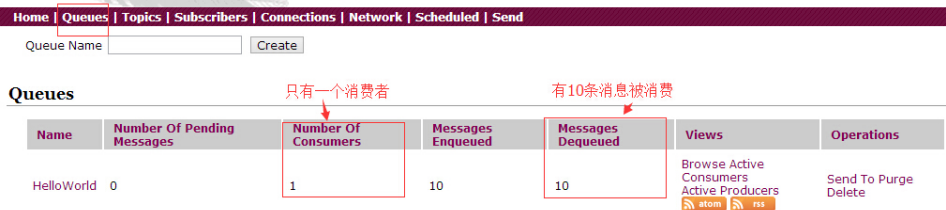
View Consumers

如果这些队列中的消息，被删除，消费者则无法消费。

3. 我们继续运行一下消费者，eclipse控制台打印消息，如下：

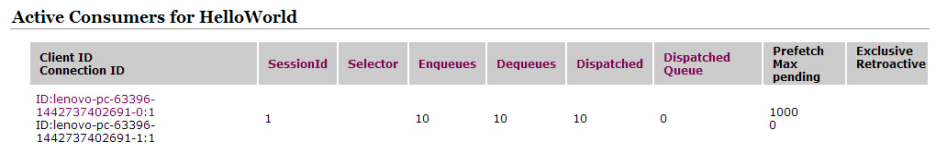


此时，我们先看一下ActiveMQ服务器，Queues内容如下：



我们可以看到HelloWorld的消息队列发生变化，多一个消息者，队列中的10条消息被消费了，点击Browse查看，已经为空了。

点击Active Consumers，我们可以看到这个消费者的详细信息：



我们的实例到此就结束了，大家可以自己多点ActiveMQ服务器的内容，进一步熟悉ActiveMQ。

总结

这篇博文我们实现了点对点的消息模型以及发送的一个同步消息，是不是非常的简单？

下面博文，我们将实现一个ActiveMQ和spring整合的实例。

源码下载

顶

52

踩

1

上一篇

分享作为程序猿的快乐

下一篇

深入浅出JMS(四)--Spring和ActiveMQ整合的完整实例

我的同类文章

【ActiveMQ】（3）

- 深入浅出JMS(四)--Spring...  
2015-09-27  
阅读 40017
  - 深入浅出JMS(一)--JMS基...  
2015-06-30  
阅读 69333
- 深入浅出JMS(二)--ActiveM...  
2015-07-31  
阅读 47310

参考知识库



Java SE知识库  
23246 关注 | 468 收录



Java EE知识库  
15552 关注 | 1260 收录



Java 知识库  
23480 关注 | 1449 收录

猜你在找

- Java 开发环境搭建(JDK、Eclipse、Tomcat、Mysql)
- 深入浅出JMS三--ActiveMQ简单的HelloWorld实例
- EasyDarwin开源流媒体服务器：编译、配置、部署
- 深入浅出JMS三--ActiveMQ简单的HelloWorld实例
- Nginx服务器入门
- 深入浅出JMS三--ActiveMQ简单的HelloWorld实例
- 高效使用eclipse调试工具
- 深入浅出JMS三--ActiveMQ简单的HelloWorld实例
- 全网服务器数据备份解决方案案例实践
- 深入浅出JMS四--Spring和ActiveMQ整合的完整实例

查看评论

24楼 xiehmjl1 2016-12-08 11:30发表



有点问题哦，接受到消息以后如果没有session.commit()的话，消息就会被认为没有被消费的。所以处理完消息后要调用session.commit();

23楼 hahahzzzzz 2016-12-06 18:44发表



java.lang.UnsupportedClassVersionError: org/apache/activemq/ActiveMQConnection : Unsupported major version 51.0，网上说包版本不匹配，这个MQ是啥版本，我Eclipse用的1.6

22楼 wshiwangtie 2016-11-10 14:10发表



经典呀，简直就是我们这些菜鸟的福音呀！

21楼 wshiwangtie 2016-11-10 10:42发表



mark

20楼 志于青筱梦于马 2016-10-31 19:04发表



可以转载吧，我转载了，如果不能转载我再删除哈（我已经注明转载并加你链接了）

19楼 T-A-N 2016-09-21 09:40发表



写的很好，一目了然

18楼 极客on之路 2016-09-06 13:13发表



很不错

17楼 坦GA 2016-09-05 12:49发表



写的好

16楼 Paranoia\_ZK 2016-08-30 11:53发表



博主，我的信息被消费之后 点击active producers为什么都没有显示 eclipse打印信息的

Re: 于亮 2016-08-30 13:00发表



回复Paranoia\_ZK: 看看哪出错了



15楼 [ccc43542876](#) 2016-08-25 17:05发表



博主想请问下，， user和password 我设置为任何值都可以发送成功。。这个应该不对吧。只有user和password都对才能连上activemq吧。。我该如何配置呢。

14楼 [影非弦](#) 2016-08-25 14:28发表



消息消费者里面连接不用关闭吗？

13楼 [我这么帅你知道吗](#) 2016-08-18 11:31发表



很不错

12楼 [damenwudia](#) 2016-07-19 09:14发表



很好的入门教程，感谢LZ分享！

11楼 [yysx](#) 2016-06-01 15:14发表



图解很详细啊

10楼 [锦落浮萍](#) 2016-05-26 14:59发表



简单，清晰，易于理解。

9楼 [qkillq](#) 2016-05-12 13:42发表



很赞！受益匪浅呀。。

8楼 [wanghongyu123](#) 2016-04-15 11:28发表



//创建session

```
session=conn.createSession(false, Session.AUTO_ACKNOWLEDGE);
```

创建生产者和消费者的session的时候，这里设置不一样，能详细说一下么？

Re: [于亮](#) 2016-04-15 11:44发表



回复wanghongyu123: createSession(paramA,paramB);

paramA是设置事务的，paramB设置acknowledgment mode

paramA设置为false时：paramB的值可为Session.AUTO\_ACKNOWLEDGE，Session.CLIENT\_ACKNOWLEDGE，DUPS\_OK\_ACKNOWLEDGE其中一个。

paramA设置为true时：paramB的值忽略，acknowledgment mode被jms服务器设置为SESSION\_TRANSACTED。

Session.AUTO\_ACKNOWLEDGE为自动确认，客户端发送和接收消息不需要做额外的工作。

Session.CLIENT\_ACKNOWLEDGE为客户确认。客户端接收到消息后，必须调用javax.jms.Message的acknowledge方法。jms服务器才会删除消息。

DUPS\_OK\_ACKNOWLEDGE允许副本的确认模式。一旦接收方应用程序的方法调用从处理消息处返回，会话对象就会确认消息的接收；而且允许重复确认。在需要考虑资源使用时，这种模式非常有效。

7楼 [VernonC](#) 2015-11-09 17:17发表



```
TextMessage textMessage = (TextMessage) messageConsumer.receive(1000);
```

这里的textMessage 获取为空

Re: [于亮](#) 2015-11-11 10:35发表



回复VernonC：你的activeMQ启动了吗？

6楼 [VernonC](#) 2015-11-09 17:16发表



博主 你好。我看了你的P2P的实例，在实际操作时，生产者能将消息发送出去，但消费者接收为空，求解？

Re: [萧乡月夜](#) 2016-04-19 00:45发表



回复VernonC：else里面都break了还收什么。。。

Re: [yaokun0627](#) 2016-11-21 17:54发表



回复萧乡月夜：我也是生产者可以发送信息，消费者不能接收信息。这是为啥

Re: [yaokun0627](#) 2016-11-21 17:52发表



回复萧乡月夜：不是 esle 的原因，else只是接收完之后跳出循环。我也试了下，也是生产者可以发送信息，消费者不能接收信息，不知道为什么啊

Re: [yaokun0627](#) 2016-11-21 18:45发表



回复yaokun0627：知道了，消费者类的 `session` 以及 工厂的连接在获取队列消息后不关闭就可以获取到信息

5楼 李晓娜 2015-10-31 19:57发表



有例子，很棒，赞一个

4楼 mayfla 2015-09-30 02:05发表



棒棒哒

3楼 张宏杰 2015-09-29 17:47发表



楼主厉害啊！

2楼 邱慕夏 2015-09-28 22:01发表



学习中，谢谢师哥分享！

1楼 左弦月 2015-09-23 17:08发表



博主，说好的ActiveMQ和Spring整合的实例了？

Re: 于亮 2015-09-23 21:05发表



回复左弦月：例子已经实现完了，还没有写博文，过两天吧

Re: 左弦月 2015-09-24 10:26发表



回复于亮：谢谢博主，我会一直关注你的

Re: 于亮 2015-09-27 00:15发表



回复左弦月：博文已经更新了哦，谢谢关注

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

...

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap