

博客 (//blog.csdn.net/)

学院 (//edu.cs(d/hwwett).csd亦載t/http://download.csdn.net)

GitChat (http://gitbook.cn/?ref=csdn)

论坛 (http://bbs.csdn.net)

Q





登录 (https://passport.csdn.net/account/mobileregister?action=mobileRegister) utm_source=csdnblog1)

Java对象序列化详解^三

原创 2016年08月10日 14:47:

标签: java (http://so.csdn.net//sp/search/s.do?q=java&t=blog) /

分布式应用 (http://so.csdn.net/so/search/s.do?q=分布式应用&t=blog) /

序列化 (http://so.csdn.net/so/search/s.do?q=序列化&t=blog)

4346

所有分布式应用常常需要跨平台,跨网络,因此要求所有传的参数、返回值都必须实现序列化。

序列化:把Java对象转换为字节序列的过程。

反序列化:把字节序列恢复为Java对象的过程。

对象的序列化主要有两种用途:

- 1) 把对象的字节序列永久地保存到硬盘上,通常存放在一个文件中;(持久化对象)
- 2) 在网络上传送对象的字节序列。(网络传输对象)

实现了如下两个接口之一的类的对象才能被序列化:

- 1) Serializable
- 2) Externalizable

序列化: ObjectOutputStream代表对象输出流,它的writeObject(Object obj)方法可对参数指定的 obj对象进行序列化,把得到的字节序列写到一个目标输出流中。

反序化: ObjectInputStream代表对象输入流,它的readObject()方法从一个源输入流中读取字节序 列,再把它们反序列化为一个对象,并将其返回。

注:使用writeObject()和readObject()方法的对象必须已经被序列化

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!



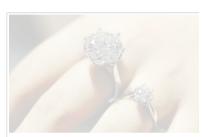
zcl_love_wx (http://blo...

+ 关注

(http://blog.csdn.net/zcl_love_wx)

未开诵

原创 粉丝 (https://gi 113 66 145 utm_sourc



·克拉钻戒价格











他的最新文章

更多文章 (http://blog.csdn.net/zcl_love_wx)

《Java设计模式与实践》——工厂模式 (http://blog.csdn.net/zcl love wx/artic le/details/79054617)

高性能的索引实现——《深究MySQL》 (http://blog.csdn.net/zcl_love_wx/artic le/details/78966200)

优化MySQL数据类型——《深究MySQ L》 (http://blog.csdn.net/zcl_love_wx/ article/details/78947899)

Mybatis中 Integer 值为0时,默认为空字 符串的解决办法 (http://blog.csdn.net/z cl_love_wx/article/details/78799719)

JavaScript中Console的9个常用调试命令 (http://blog.csdn.net/zcl_love_wx/artic le/details/78651560)

文章分类

 \triangle 内容举报

AngularJS (http://blog.csdn.... _{「for} 18篇

JavaEE (http://blog.csdn.net返回顶部15篇

多线程 (http://blog.csdn.net...

Solr (http://blog.csdn.net/zc... 9篇

集合 (http://blog.csdn.net/z...

登录展开~ 注册 X

2篇

四、深入理解

一、为什么要序列化?

Java平台允许我们在内存中创建可复用的Java对象,但只有当JVM(Java虚拟机)处于运行时,这些 对象才可能存在,也就是这些对象的生命周期不会比JVM的生命周期更长。但在现实应用中,就可能要求 在JVM停止运行之后能够保存指定的对象(持久化对象),并在将来重新读取被保存的对象。Java对象序列化 就实现了该功能。

网络通信时,无论是何种类型的数据,都会转成字节序列的形式在网络上传送。发送方需要把这个 Java对象转换为字节序列,才能在网络上传送;接收方则需要把字节序列再恢复为Java对象。Java对象序 ··· 列化也实现了该功能。

所以序列化机制会把内存中的Java对象转换成与平台无关的二进制流,从而永久地保存在磁盘上或是 通过网络传输到另一个网络节点。

二、序列化的意义

客户端访问了某个能开启会话功能的资源 ,web服务器就会创建一个与该客户端对应的HttpSession对 象,每个HttpSession对象都要站用一定的内存空间。如果在某一时间段内访问站点的用户很多,web服务 器内存中就会积累大量的HttpSession对象,消耗大量的服务器内存,即使用户已经离开或者关闭了浏览 器,web服务器仍要保留与之对应的HttpSession对象,在他们超时之前,一直占用web服务器内存资源。

web服务器通常将那些暂时不活动但未超时的HttpSession对象转移到文件系统或数据库中保存,服务器要 使用他们时再将他们从文件系统或数据库中装载入内存,这种技术称为Session的持久化。

将HttpSession对象保存到文件系统或数据库中,需要采用序列化的方式将HttpSession对象中的每个属性 对象保存到文件系统或数据库中;将HttpSession对象从文件系统或数据库中装载如内存时,需要采用反序 列化的方式,恢复HttpSession对象中的每个属性对象。所以存储在HttpSession对象中的每个属性对象必 须实现Serializable接口

三、Session的持久化的作用

1.提高服务器内存的利用率,保证那些暂停活动的客户端在会话超时之前继续原来的会话

- 2,在多台web服务器协同对外提供服务的集群系统中,使用Session的持久化技术,某台服务器可以将其 中发生改变的Session对象复制给其他服务器。保证了在某台服务器停止工作后可以由其他服务器来接替它 与客户端的会话
- 3,在一个web应用程序重启时,服务器也会持久化该应用程序中所有HttpSession对象,保证客户端的会 话活动仍可以继续。

Tomcat使用Session Manager 类来管理Session的持久化,他提供了两个SessionManager类 org.apache.catalina.session.StandardManager

org. apache. catalina. session. PersistentManager

StandardManager是tomcat默认使用的,在web应用程序关闭时,对内存中的所有HttpSession对象进行 持久化,把他们保存到文件系统中。默认的存储文件为

四、serialVersionUID

如果serialVersionUID没有显式生成,系统就会自动生成一个。此时,如果在序列化后我们将该类作添加 或减少一个字段等的操作,系统在反序列化时会重新生成一个serialVersionUID然后去和已经序列化的对 加入家沪科山、泰受鼎獲、佛幹肉等樵葉不与黎的有程等。另方開模技、种问题,一般系统都会要求实现serialiable接

博主专栏

(https://passpo



angularJS应用详解 (http://blog.csdn.net/colum data.html)

(http://blog.csdn?net/column/detail data.html)

文章存档

2018年1月 (http://blog.csdn.... 2017年12月 (http://blog.csd... 2篇 2017年11月 (http://blog.csd... 7篇 2017年10月 (http://blog.csd... 2篇 2017年9月 (http://blog.csdn.... 2篇 展开~

他的热门文章

深究AngularJS——自定义服务详解(fact ory、service、provider) (http://blog.cs dn.net/zcl_love_wx/article/details/514 04390)

33047

深究AngularJS——ui-router详解 (htt p://blog.csdn.net/zcl_love_wx/article/ details/52034193)

24156

深究AngularJS——如何获取input的焦点 (自定义指令) (http://blog.csdn.net/z cl_love_wx/article/details/51382197) **12761**

深究AngularJS---ng-checked (回 写:带真实案例代码) (http://blog.csd n.net/zcl_love_wx/article/details/5139 3980)

10983

solr管理界面详解 (http://blog.csdn.net/ zcl_love_wx/article/details/52092098)

10822













返回顶部

联系我们

(http://wpa.qq.com/msgrd? v=3&uin=2431299880&site=qq&r ► webmaster@csdn.net (mailto:webmaster@csdn.net) 登录 注册

口的类显式的生明一个serialVersionUID。

显式定义serialVersionUID的两种用途:

- 1、希望类的不同版本对序列化兼容时,需要确保类的不同版本具有相同的serialVersionUID;
- 2、 不希望类的不同版本对序列化兼容时,需要确保类的不同版本具有不同的serialVersionUID。

五、序列化机制算法

- 1. 所有保存到磁盘中的对象都有一个序列化编号
- 2. 当程序试图序列化一个对象时,程序先检查该对象是否已经被序列化过。如果从未被序列化过,系统就会将该对象转换成字节序列并输出;如果已经序列化过,将直接输出一个序列化编号。

六、示例

要被序列化的对象对应的类的代码:

```
public class Person implements Serializable {
 1
2
         private String name = null;
3
         private Integer age = null;
 4
5
         public Person() {
             System. out. println("无参构造");
6
7
         public Person(String name, Integer age) {
             this. name = name;
8
             this age = age:
9
         //getter setter方法省略...
10
         @Override
11
        public String toString() {
   return "[" + name + ",
12
13
```

MySerilizable 是一个简单的序列化程序,它先将一个Person对象保存到文件person.txt中,然后再从该文件中读出被存储的Person对象,并打印该对象。

```
public class MySerilizable {
2
        public static void main(String[] args) throws Exception {
3
           File file = new File("person.txt");
4
            //序列化持久化对象
            ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(file));
5
            Person person = new Person("Peter", 27);
6
           out.writeObject(person);
7
            out.close();
8
            //反序列化,并得到对象
9
            ObjectInputStream in = new ObjectInputStream(new FileInputStream(file));
            Object newPerson = in.readObject(); // 没有强制转换到Person类型
10
            in. close():
11
            System.out.println(newPerson);
12
13
```

输出结果:

```
1 [Peter, 27]
```

结果没有打印"无参构造",说明反序列化机制无需通过构造器来初始Java对象。

注:

1.) 反序列化读取的仅仅是Java对象的数据,而不是Java类,所以在反序列化时必须提供该Java对象加入**所满类**的**原因现代**的**企**。

> 关于 招聘 广告服务 【] 阿里豆 ©2018 CSDN 京ICP证09002463号 (http://www.miibeian.gov.cn/)

🕜 经营性网站备案信息

⚠
内容举报

(記) (返回)(近部

登录 注册 🕻

2).当重新读取被保存的Person对象时,并没有调用Person的任何构造器,说明反序列化机制无须通过构造器来初始化对象。

七、选择序列化

transient

2 **:=**

当对某个对象进行序列化时,系统会自动将该对象的所有属性依次进行序列化,如果某个属性引用到别一个对象,则被引用的对象也会被序列化。如果被引用的对象的属性也引用了其他对象,则被引用的对象也会被序列化。 这就是递归序列化。

有时候,我们并不希望出现递归序列化,或是某个存敏感信息(如银行密码)的属性不被序列化,我们就可通过transient关键字修饰该属性来阻止被序列化。

将上面的Person类的age属性用transient修饰:

```
1 transient private Integer age = null;
```

再去执行MySerilizable的结果为:

```
1 [Peter, null] //返序列化时没有值,说明age字段未被序列化
```

writeObject()方法与readObject()方法

使用transient关键字阻止序列化虽然简单方便,但被它修饰的属性被完全隔离在序列化机制之外,导致了在反序列化时无法获取该属性的值,而通过在需要序列化的对象的Java类里加入writeObject()方法与readObject()方法可以控制如何序列化各属性,甚至完全不序列化某些属性(此时就transient一样)。

如果我们想要上面的Person类里的name属性在序列化后存在文件里不让别人知道具体是什么(加密),我们就可在Person类里加如下代码:

```
//自定义序列化
   private void writeObject(ObjectOutputStream out) throws IOException
2
       out.defaultWriteObject(); // 将当前类的非静态和非瞬态字段写入此流。
     //如果不写,如果还有其他字段,则不会被序列化
3
4
     out.writeObject(new StringBuffer(name).reverse());
      //将name简单加密(反转),这样别人就知道是怎么回事,当然实际应用不可能这样加密。
5
6
      out.writeInt(age);
7
8
   //反序列化
9
   //in.defaultReadObject();// 从此流读取当前类的非静态和非瞬态字段。
10
     //如果不写,其他字段就不能被反序列化
11
12
      name = ((StringBuffer)in.readObject()).reverse().toString(); //解密
13
      age = in.readInt();
14
```

详细的自定义序列化与反序列化可参见ObjectOutputStream 和ObjectInputStream 类的JDK文档。

Externalizable接口

Externalizable接口 与Serializable 接口类似,只是Externalizable接口需要强制自定义序列化。要序列化对象的代码:

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

⚠
内容举报

命 返回顶部

登录

```
public class Teacher implements Externalizable{
 1
 2
         private String name;
         private Integer age;
 3
 4
         public Teacher() {
 5
             System.out.println("无参构造");
 6
         public Teacher(String pame, Integer age) {
System.out.printLL(有参构造");
 7
             System.out.prin...
this.name = name;
 8
 9
              this.age = age;
10
11
         //setter、getter方法省略
12
         @Override
public void writeExternal(ObjectOutput out) throws IOException {
13
              out.writeObject(new StringBuffer(name).reverse()); //将name简单加密
14
              //out.writeInt(age) //注掉这句后,age属性将不能被序化
15
16
         @Override
17
         public void readExternal(ObjectInput in) throws IOException,
18
                 ClassNotFoundException
              name = ((StringBuffer) in.readObject()).reverse().toString();
19
              //age = in.readInt();
20
21
         @Override
22
         public String toString() {
    return "[" + name + ", " + age+ "]";
23
24
25
```

主函数代码改为:

```
public class MySerilizable {
2
       public static void main(String[] args) throws Exception {
3
           File file = new File("person.txt");
4
            //序列化持久化对象
5
           ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(file));
           Teacher person = new Teacher("Peter", 27);
6
           out.writeObject(person);
7
            out.close();
8
            //反序列化,并得到对象
9
           ObjectInputStream in = new ObjectInputStream(new FileInputStream(file));
           Object newPerson = in.readObject(); // 没有强制转换到Person类型
10
            in.close();
11
            System.out.println(newPerson);
12
13
```

打印结果:

```
1 有参构造
无参构造 //与Serializable 不同的是,还调用了无参构造
2 [Peter, null] //age未被序列化,所以未取到值
```

八、单例模式的序列化

当我们使用Singleton模式时,应该是期望某个类的实例应该是唯一的,但如果该类是可序列化的,那么情况可能略有不同。此时对第2节使用的Person类进行修改,使其实现Singleton模式,如下所示:

⚠
内容举报

命 返回顶部

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

登录

注册 🗙

```
public class Person implements Serializable {
 1
2
         private static class InstanceHolder {
            private static final Person instatuce = new Person("John", 31, "男");
3
 4
 5
         public static Person getInstance() {
             return InstanceHolder.instatuce;
 6
 7
         private String name = 111;
 8
 9
         private Integer age = 2nul1;
10
         private Gender gender ≡null;
11
         private Person() {
12
             System. out. printl (无参构造");
13
14
        private Person(String.TMme, Integer age, String gender) {
    System.out.println("有参构造");
15
16
             this.name = name:
             this.age = age;
17
             this.gender = gender;
18
19
```

同时要修改MySerilizable 应用,使得能够保存/获取上述单例对象,并进行对象相等性比较,如下代码所示:

```
public class MvSerilizable {
2
       public static void main(String[] args) throws Exception {
           File file = new File("person.txt");
3
           ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(file));
4
           out.writeObject(Person.getInstance()); // 保存单例对象
5
           out.close():
6
           ObjectInputStream in = new ObjectInputStream(new FileInputStream(file));
7
            Object newPerson = in.readObject();
            in.close();
8
           System.out.println(newPerson);
9
           System.out.println(Person.getInstance() == newPerson); // 将获取的对象与Person类中的单例对象进行相等
10
    性比较
11
12
```

打印结果:

九、序列化对象注意事项

- 1. 对象的类名、属性都会被序列化;而方法、static属性(静态属性)、transient属性(即瞬态属性)都不会被序列化(这也就是第4条注意事项的原因)
- 2. 虽然加static也能让某个属性不被序列化,但static不是这么用的
- 3. 要序列化的对象的引用属性也必须是可序列化的,否则该对象不可序列化,除非以transient关键字修饰该属性使其不用序列化。
- 4. 反序列化地象时必须有序列化对象生成的class文件(很多没有被序列化的数据需要从class文件获取)
- 5. 当通过文件、网络来读取序列化后的对象时,必须按实际的写入顺序读取。

⚠
内容举报

命 返回顶部

版权声明:博主地址:http://blog.csdn.net/zcl_love_wx

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

登录 注册 🗙

Q

javac4 (/javac4) 2017-09-12 08:59 (/javas)的真好,帮大忙了! ď

回复 1楼

2

输入输出之对象序列化Serializable

🌑 hanhan1016 2015年09月18日 14:21 🕮 998

1.对象序列化的目标 将对象保存到磁盘中,或允许在网络中直接输出对象。 2.对象序列化的含义 对象的序列化是指将一个Ja va对象写入IO流中。 对象的反序列化是...

(http://blog.csdn.net/hanhan1016/article/details/48545625)

Java中的序列化Serialable高级详解 🧁 jiangwei0910410003 2014年02月08日 17:19 🕮 30572



引言将 Java 对象序列化为二进制文件的 Java 序列化技术是 Java 系列技术中一个较为重要的技术点,在大部分情况下,开 发人员只需要了解被序列化的类需要实现 Serializable 接口...

(http://blog.csdn.net/jiangwei0910410003/article/details/18989711)

Java中如何序列化一个对象 - CSDN博客

Java中如何序列化一个对象我们都知道java 中无法保存一个对象到文本文件中,但是当我们有这种需求的时候,我们可以通过ja va 的序列化功能把...

2018-1-15

(http://blog.csdn.net/wgw335363240/article/details/6443556?>)

对象序列化的几种方式的比较 - CSDN博客

比较常见的做法有两种:一是把对象包装成JSON字符串传输,二是采用java对象的序列化和反序列化。随着Google工具protoB uf的开源,protobuf也是个不错的选择。对JSON,...

2018-1-17

(http://blog.csdn.net/a925907195/article/details/52434265?>)

程序员不会英语怎么行?



老司机教你一个数学公式秒懂天下英语

Java对象序列化与反序列化



BonJean 2016年07月25日 22:59

2359

Java对象转换为字节序列的过程称为序列化,用途是对象永久保存、传输。序列化运行时使用一个称为 serialVersionUID 的 版本号与每个可序列化类相关联,该序列号在反序列化过程中用于验证序列化...

(http://blog.csdn.net/BonJean/article/details/52027656)

Java对象序列化详解 - CSDN博客

 \triangle 内容举报

所有分布式应用常常需要跨平台.跨网络,因此要求所有传的参数、返回值都必须实现序列化。一、定义序列化:把Java对象转 换为字节序列的过程。 反序列化:把字节序列...

TOP 返回顶部

2018-1-18

(http://blog.csdn.net/zcl love wx/article/details/52126876?>)

对象的序列化和反序列化-CSDN下载

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

登录

序列化是将某个对象改写成某种信息格式 然后存储存于某种介质上的过程 当然 某种信息的格式是要能被恢复回来的 这就是 反序列化. 参考博客:http://blog.csdn....

(https://passpo

2018-1-16

(http://download.csdn.net/detail/u010963246/8858295?>)

Java 对象序列化机制详解。



🔇 zuiwuyuan 2014年10月09日 22:24 🕮 1066

对象序列化的目标:将对象保存到磁盘中,或允许在网络中直接传输对象。对象序列化机制允许把内存中的Java对象转换成 平台无关的二进制流,从而允许把这种二进制流持久的保存在磁盘上,通过网络将这种二进制流...

(http://blog.csdn.net/zuiwuyuan/article/details/39940839)

Java序列化与反序列化



娜 wangloveall 2012年09月18日 16:48 🕮 189093

Java序列化与反序列化是什么?为什么需要序列化与反序列化?如何实现Java序列化与反序列化?本文围绕这些问题进行了 探讨。 1.Java序列化与反序列化 Java序列化是指把Java对象转换为...

(http://blog.csdn.net/wangloveall/article/details/7992448)

什么是对象序列化?如何实现?什么情况下使用? - CSDN博客

a)什么是对象序列化: 对象序列化可以将一个对象保存到一个文件,可以将通过流的方式在网络上传输,可以将文件的内容读取转 化为一个对象。所谓对象流也就是将对象的...

2018-1-18

(http://blog.csdn.net/jimbowong/article/details/54909581?>)

Java中,什么是实现对象序列化?如何实现对象序列化? - CSDN博客

对象序列化可以将一个对象保存到一个文件,可以将通过流的方式在网络上传输,可以将文件的内容读取转化为一个对象。所谓 对象流也就是将对象的内容流化,可以对流化后...

2018-1-16

(http://blog.csdn.net/u012868077/article/details/46933395?>)

改变,从讲一口流利英文开始



随时随地学习,把欧美外教装在口袋里!现在注册可领取15分钟免费课程!

深入理解Java对象序列化



_____zero__007 2016年08月19日 22:37 🕮 2336

转载自: http://www.blogjava.net/jiangshachina/archive/2012/02/13/369898.html 1.什么是Java对象序列化 使用Java 对象序列化...

(http://blog.csdn.net/zero__007/article/details/52254308)

对象序列化原因的简单理解 - CSDN博客

序列化和反序列化我们可能经常会听到,其实通俗一点的解释,序列化就是把一个对象保存到一个文件或数据库字段中去,其最终 目的都是将内存中的对象持久化或者是在...

2018-1-17

(http://blog.csdn.net/sundacheng1989/article/details/30101627?>)

\triangle 内容举报



对象的序列化与反序列化---IO学习笔记(四) - CSDN博客

对象的序列化,反序列化对象的序列化:就是将Object转换成byte序列对象的反序列化:将byte序列转换成Object序列化流,反 序列化流序列化流(ObjectOutputStream),...

2018-1-11

加水性的内容整度2岁9000克程序與傾便使37?>)

登录

对Java Serializable (序列化)的理解和总结 @ dreamtdp 2013年11月11日 20:00 @ 45833

(http://blog.csdn.net/dreamtdp/article/details/15378329)

Java的IO流与对象的序列化浅谈

🍘 LULEI1217 2016年01月16日 13:54 🕮 1990

今天再次回顾了一些关于java IO流的知识。虽然以前会点IO的方面的编程,但是还是知其然不知其所以然。首先上一张IO流的主要家族图吧。 该图给出了IO流的具…

(http://blog.csdn.net/LULEI1217/article/details/50527824)

java,什么是序列化,怎么实现序列化 - CSDN博客

序列化就是一种用来处理对象流的机制,所谓对象流也就是将对象的内容进行流化(将对象转换成二进制)。可以对流化后的对象进行读写操作,也可将流化后的对象传输于网...

2018-1-18

(http://blog.csdn.net/zhangzhangjava/article/details/51135413?>)

iOS对象的序列化 - CSDN博客

当你于写数据需要本地存储时,即将你的数据写到硬盘上的时候,你就必须对他进行序列化,转换成二进制文件,从而便于在磁盘上的读写,同理在取出的时候必须将其在反...

2018-1-12

(http://blog.csdn.net/skylin19840101/article/details/50999033?>)

Java基础学习总结——Java对象的序列化和反序列化

序列化。 把字节序列恢复为对象的过程称为对象的反序列化。 对象的序列...

(http://blog.csdn.net/weibin_6388/article/details/50955553)

java提高篇(五)-----使用序列化实现对象的拷贝

(http://blog.csdn.net/chenssy/article/details/12952063)

Java中如何序列化一个对象

● wgw335363240 2011年05月24日 22:11 □ 16121

Java中如何序列化一个对象我们都知道java 中无法保存一个对象到文本文件中,但是当我们有这种需求的时候,我们可以通过java 的序列化功能把当前对象的一些属性以二进制的形式保存到文件中。当我们...

(http://blog.csdn.net/wgw335363240/article/details/6443556)

a)什么是对象序列化: 对象序列化可以将一个对象保存到一个文件,可以将通过流的方式在网络上传输,可以将文件的内容读取转化为一个对象。 所谓对象流也就是将对象的内容流化,可以对流化后的对象进行读写操作,也...

(http://blog.csdn.net/JimboWonG/article/details/54909581)

怎样对带有不可序列化属性的Java对象进行序列化

加法使用默认的表现精湛的内容能适宜中与疑心真程病现法用通话定制的序列化方法,对一...

⚠
内容举报

忘 返回顶部

登录 注册

(http://blog.csdn.net/CSDN_LOR/article/details/51464338)

(https://passpo

Java序列化的几种方式以及序列化的作用



№ u012554102 2016年07月14日 00:00 □ 7676

文章转载自: 本文着重讲解一下Java序列化的相关内容。如果对Java序列化感兴趣的同学可以研究一下。一.Java序列化的作 用有的时候我们想要把一个Java对象变成字节流的形式传出去,有的时候我们想...

(http://blog.csdn.net/u012554102/article/details/51902697)

对象序列化的几种方式的比较



在java中socket传输数据时,数据类等在往比较难选择。可能要考虑带宽、跨语言、版本的兼容等问题。比较常见的做法有两 种:一是把对象包装成JSON字符串传输,二是采用java对象的序列化和反序列化。...

(http://blog.csdn.net/a925907195/article/details/52434265)

···

Java序列化的几种方式



얘 wl6965307 2016年04月18日 11:13 🔘 14206

本文着重讲解一下Java序列化的相关内容。 如果对Java序列化感兴趣的同学可以研究一下。 一.Java序列化的作用 有的时 候我们想要把一个Java对象变成字节流的形式传出去,...

(http://blog.csdn.net/wl6965307/article/details/51178143)

Java序列化对象的一个使用案例-使用Http发送对象

《 Effective Java 》中序列化一节关于java的序列化存在如下说 法:对象序列化(object serialization)API,它提供了一个框架,用 来将对象编码成字节流(seriali...

(http://blog.csdn.net/zmx729618/article/details/52814213)

Java对象的serialVersionUID在序列化和反序列化的用途

本博客主要转自如下链接 http://blog.csdn.net/javazejian/articl 闷 u011734144 2016年09月29日 12:44 🚨 1553 e/details/52665164 这篇文章写的不错,但是有些地方我估计博 主没有亲自测试,所以有...

(http://blog.csdn.net/u011734144/article/details/52700868)

[疯狂Java]I/O:I/O流的最高境界——对象流(序列化:手动序列化、自动序列化、...

1. 什么是对象流:序列化/反序列化的概念 1) 对象流是和字节流/字符流同处于一个概念体系的: a. 这么说字节流是 流动的字节序列,字符流是流动的字符序列,那么对象流就是流动...

C Lirx_Tech 2016年04月28日 13:41 □ 1979

(http://blog.csdn.net/Lirx_Tech/article/details/51273380)

Java 对象序列化机制详解



zuiwuyuan 2014年10月09日 22:24 🕮 1066

对象序列化的目标:将对象保存到磁盘中,或允许在网络中直接传输对象。 对象序列化机制允许把内存中的Java对象转换成 平台无关的二进制流,从而允许把这种二进制流持久的保存在磁盘上,通过网络将这种二进制流...

(http://blog.csdn.net/zuiwuyuan/article/details/39940839)

 \triangle 内容举报

TOP 返回顶部

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

登录