

zeroone

知行合一

昵称: 武胜-阿伟
园龄: 8年4个月
粉丝: 198
关注: 80
[+加关注](#)

< 2012年6月 >						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[C#\(479\)](#)
[winform\(109\)](#)
[devpress\(76\)](#)
[控件\(71\)](#)
[股票\(67\)](#)
[wcf\(61\)](#)
[mysql\(43\)](#)
[生活\(36\)](#)
[dll\(27\)](#)
[tpl\(26\)](#)
[更多](#)

随笔档案(758)

[2016年7月 \(3\)](#)

[博客园](#) [首页](#) [博文](#) [闪存](#) [新随笔](#) [联系](#) [订阅](#) [XML](#) [管理](#)
随笔-758 评论-103 文章-35 trackbacks-0

DataSet用法详细 转

DataSet用法详细

一、特点介绍

- 1、处理脱机数据，在多层应用程序中很有用。
- 2、可以在任何时候查看DataSet中任意行的内容，允许修改查询结果的方法。
- 3、处理分级数据
- 4、缓存更改
- 5、XML的完整性：DataSet对象和XML文档几乎是可互换的。

二、使用介绍

1、创建DataSet对象：

DataSetds = new DataSet("DataSetName");

2、查看调用SqlDataAdapter.Fill创建的结构

da.Fill(ds,"Orders");
DataTbletbl = ds.Table[0];
foreach(DataColumncol in tbl.Columns)
Console.WriteLine(col.ColumnName);

3、查看SqlDataAdapter返回的数据

①DataRow对象

DataTbletbl = ds.Table[0];
DataRowrow = tbl.Row[0];
Console.WriteLine(ros["OrderID"]);

②检查存储在DataRow中的数据

2016年5月 (3)
2016年4月 (3)
2016年3月 (4)
2016年2月 (5)
2016年1月 (19)
2015年12月 (6)
2015年11月 (5)
2015年10月 (33)
2015年9月 (23)
2015年8月 (11)
2015年7月 (7)
2015年6月 (8)
2015年5月 (7)
2015年4月 (18)
2015年3月 (34)
2015年2月 (9)
2015年1月 (6)
2014年12月 (16)
2014年11月 (7)
2014年10月 (5)
2014年9月 (14)
2014年8月 (3)
2014年7月 (8)
2014年6月 (6)
2014年5月 (51)
2014年4月 (22)
2014年3月 (36)
2014年2月 (8)
2014年1月 (18)
2013年12月 (4)
2013年11月 (4)
2013年10月 (14)
2013年9月 (45)
2013年8月 (27)
2013年7月 (15)
2013年6月 (13)
2013年5月 (16)
2013年4月 (16)
2013年3月 (16)
2013年2月 (4)
2013年1月 (6)
2012年12月 (14)

```
DataTable tbl = row.Table;
```

```
foreach(DataColumn col in tbl.Columns)
```

```
Console.WriteLine(row[col]);
```

③检查**DataTable**中的**DataRow**对象

```
foreach(DataRow row in tbl.Rows)
```

```
DisplayRow(row);
```

4、校验**DataSet**中的数据

①校验**DataColumn**的属性：

ReadOnly, AllowDBNull, MaxLength, Unique

②**DataTable**对象的**Constraints**集合：

UniqueConstraints, PrimaryKey, ForeignKeyConstraints

通常不必刻意去创建ForeignKeyConstraints, 因为当在DataSet的两个DataTable对象之间创建关系时会创建一个。

③用**SqlDataAdapter.Fill**模式来检索模式信息

5、编写代码创建**DataTable**对象

①创建**DataTable**对象：

```
DataTable tbl = new DataTable("TableName");
```

②将**DataTable**添加到**DataSet**对象的**Table**集合

```
DataSet ds = new DataSet();
```

```
DataTable tbl = new DataTable("Customers");
```

```
ds.Tables.Add(tbl);
```

```
DataSet ds = new DataSet();
```

```
DataTable tbl = ds.Tables.Add("Customers");
```

DataTable对象只能存在于至多一个DataSet对象中。如果希望将DataTable添加到多个DataSet中, 就必须使用Copy方法或Clone方法。Copy方法创建一个与原DataTable结构相同并且包含相同行的新DataTable; Clone方法创建一个与原DataTable结构相同, 但没有包含任何行的新DataTable。

③为**DataTable**添加列

```
DataTable tbl = ds.Tables.Add("Orders");
```

```
DataColumn col
```

```
=tbl.Columns.Add("OrderID", typeof(int));
```

```
col.AllowDBNull = false;
```

```
col.MaxLength = 5;
```

[2012年11月 \(7\)](#)
[2012年10月 \(28\)](#)
[2012年9月 \(16\)](#)
[2012年8月 \(1\)](#)
[2012年6月 \(10\)](#)
[2012年5月 \(13\)](#)
[2012年4月 \(23\)](#)
[2012年3月 \(2\)](#)
[2012年2月 \(1\)](#)
[2011年12月 \(2\)](#)
[2011年8月 \(3\)](#)
[2011年7月 \(1\)](#)
[2011年6月 \(1\)](#)
[2011年4月 \(2\)](#)
[2011年3月 \(3\)](#)
[2011年1月 \(2\)](#)
[2010年12月 \(1\)](#)
[2010年11月 \(2\)](#)
[2010年5月 \(4\)](#)
[2010年4月 \(16\)](#)
[2010年3月 \(14\)](#)
[2010年2月 \(1\)](#)
[2010年1月 \(6\)](#)
[2009年12月 \(1\)](#)
[2009年11月 \(1\)](#)
[2008年9月 \(1\)](#)
[2008年4月 \(1\)](#)
[2008年3月 \(3\)](#)

积分与排名

积分 - 379653

排名 - 294

最新评论

1. Re:c#写入Mysql中文显示乱码 解决方法 z

问下大神，从MSSQL查出的数据插入到MySql乱码怎么解决试过直接对链接 MySql 字符串加Charset=utf8但是这样程序直接报Incorrect string value: '\xE9\.....

--皑上雪

2. Re:在Dev GridControl中添加颜色可变的ProgressBar z

```
col.Unique= true;
```

```
tbl.PrimaryKey= new DataColumn[]
{tbl.Columns["CustomersID"]};
```

当设置主键时，AllowDBNull自动设置为False;

④处理自动增量列

```
DataSetds = new DataSet();
```

```
DataTabletbl = ds.Tables.Add("Orders");
```

```
DataColumncol =
tbl.Columns.Add("OrderID",typeof(int));
```

```
col.AutoIncrement= true;
```

```
col.AutoIncrementSeed= -1;
```

```
col.AutoIncrementStep= -1;
```

```
col.ReadOnly= true;
```

⑤添加基于表达式的列

```
tbl.Columns.Add("ItemTotal",typeof(Decimal),"Quan
tity*UnitPrice");
```

6、修改DataTable内容

①添加新DataRow

```
DataRowrow = ds.Tables["Customers"].NewRow();
```

```
row["CustomerID"]= "ALFKI";
```

```
ds.Tables["Customers"].Rows.Add(row);
```

```
object[]aValues={"ALFKI","Alfreds","Anders","030-
22222"};
```

```
da.Tables["Customers"].LoadDataRow(aValues,false
);
```

②修改当前行

修改行的内容逼供内不会自动修改数据库中相应的内容，对行所做的修改被视为是随后将使用SqlDataAdapter对象来提交交给数据库的待定的更改。

```
DataRowrowCustomer;
```

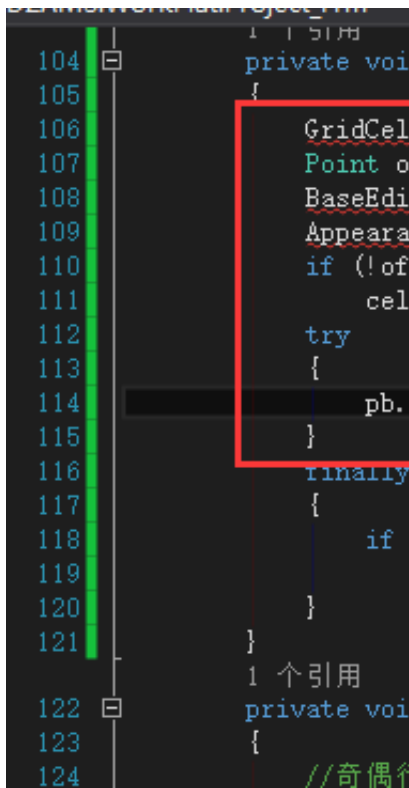
```
rowCustomer=
ds.Tables["Custoemrs"].Rows.Find("ANTON");
```

```
if(rowCustomer== null)
```

```
//没有查找客户
```

```
else
```

```
{
```



--David Gong

3. Re:c#写入Mysql中文显示乱码 解决方法 z

不错，解决了我的问题，非常感谢

--雨夜1

4. Re:tdx api z

请问你们弄到TdxApi.dll了吗？有的话可以发我邮箱哦

--cnpc_hr

5. Re:Vc code

请问你有相关的dll么，有偿哦

270010904

--cnpc_hr

阅读排行榜

1. MySQL日期时间函数大全 转 (363074)
2. mysqldump备份还原和mysqldump导入导出语句大全详解 转(104723)
3. DataSet用法详细 转(28430)
4. DevExpress控件学习总结 z(19797)
5. MySql数据库导出csv(17863)

评论排行榜

1. C#发送Email邮件（实例：QQ邮箱和Gmail邮箱）(5)
2. DevExpress 中根据数据库字典动态生成卡式菜单 z(5)
3. DevExpress控件学习总结 z(5)
4. WCF 获取客户端IP 整理(4)

```

rowCustomer["CompanyName"]="NewCompanyNa
me";

rowCustomer["ContactName"]="NewContactName";
}

//推荐使用这种方式
DataRowrowCustomer;

rowCustomer=
ds.Tables["Custoemrs"].Rows.Find("ANTON");
if(rowCustomer== null)
    //没有查找客户
else
{
    rowCustomer.BeginEdit();

    rowCustomer["CompanyName"]="NewCompanyNa
me";

    rowCustomer["ContactName"]="NewContactName";

    rowCustomer.EndEdit();
}

//null表示不修改该列的数据
obejct[]aCustomer =
{null,"NewCompanyName","NewContactName",null}

DataRowrowCustomer;

rowCustomer=
ds.Tables["Customers"].Rows.Find("ALFKI");

rowCustomer.ItemArray= aCustomer;

③处理DataRow的空值
//查看是否为空

DataRowrowCustomer;

rowCustomer=
ds.Tables["Customers"].Rows.Find("ALFKI");

if(rowCustomer.IsNull("Phone"))
    Console.WriteLine("It"'sNull");

```

5. net 中捕获摄像头视频的方式及对比(How to Capture Camera Video via .Net) (转) (4)

推荐排行榜

1. MySQL日期时间函数大全 转(9)
2. DataSet用法详细 转(5)
3. wireshark tcp 协议分析 z(5)
4. C# Process.Start()方法详解(4)
5. 用C#调用Windows API向指定窗口发送按键消息 z(4)

else

```
Console.WriteLine("It""snot Null");
```

//赋予空值

```
rowCustomer["Phone"]= DBNull.Value;
```

④删除DataRow

```
DataRow rowCustomer;
```

```
rowCustomer=
ds.Tables["Customers"].Rows.Find("ALFKI");
```

```
rowCustomer.Delete();
```

⑤清除DataRow

```
DataRow rowCustomer =
ds.Tables["Customers"].Rows.Find("ALFKI");
```

```
rowCustomer.ItemArray= aCustomer;
```

```
ds.Tables["Customers"].Remove(rowCustomer);
```

或者

```
ds.Tables["Customers"].RemoveAt(intIndex);
```

⑥使用DataRow.RowState属性：

Unchanged,Detached,Added,Modified,Deleted

```
private void DemonstrateRowState()
```

```
{
```

```
    // Run a function to create a DataTable with one
    column.
```

```
    DataTable myTable = MakeTable();
```

```
    DataRow myRow;
```

```
    // Create a new DataRow.
```

```
    myRow = myTable.NewRow();
```

```
    // Detached row.
```

```
    Console.WriteLine("New Row "
+ myRow.RowState);
```

```
    myTable.Rows.Add(myRow);
```

```
    // New row.
```

```
    Console.WriteLine("AddRow "
+ myRow.RowState);
```

```
    myTable.AcceptChanges();
```

```
    // Unchanged row.
```

```
Console.WriteLine("AcceptChanges" +  
myRow.RowState);  
  
myRow["FirstName"] = "Scott";  
  
// Modified row.  
  
Console.WriteLine("Modified "  
+ myRow.RowState);  
  
myRow.Delete();  
  
// Deleted row.  
  
Console.WriteLine("Deleted "  
+ myRow.RowState);  
  
}
```

⑦ 检查 **DataRow** 中的挂起更改

```
DataRow rowCustomer;  
  
rowCustomer =  
ds.Tables["Customers"].Rows.Find("ALFKI");  
  
rowCustomer["CompanyName"] =  
"NewCompanyName";  
  
string strNewCompanyName, strOldCompanyName;  
  
Console.WriteLine(rowCustomer["CompanyName",  
DataRowVersion.Current]);  
  
Console.WriteLine(rowCustomer["CompanyName",  
DataRowVersion.Original]);
```

⑧ 遍历 **DataSet**

```
foreach(DataTable dt in dataSet.Tables)  
  
foreach(DataRow dr in dt.Rows)  
  
foreach(DataColumn dc in dr.Table.Columns)  
  
Console.WriteLine(dr[dc]);
```

三、属性方法事件介绍

1、DataSet

① 属性

CaseSensitive: 用于控制 DataTable 中的字符串比较是否区分大小写。

DataSetName: 当前 DataSet 的名称。如果不指定，则该属性值设置为 "NewDataSet"。如果将 DataSet 内容写入 XML 文件，DataSetName 是 XML 文件的根节点名称。

DesignMode: 如果在设计时使用组件中的 DataSet，DesignMode 返回 True，否则返回 False。

HasErrors:表示DataSet中的DataRow对象是否包含错误。如果将一批更改提交给数据库并将DataAdapter对象的ContinueUpdateOnError属性设置为 True, 则在提交更改后必须检查DataSet的HasErrors属性, 以确定是否有更新失败。

Namespace和**Prefix**:指定XML命名空间和前缀

Relations:返回一个DataRelationCollection对象。

Tables:检查现有的DataTable对象。通过索引访问DataTable有更好的性能。

② 方法

AcceptChanges和**RejectChanges**: 接受或放弃DataSet中所有挂起更改。调用AcceptChanges时, RowState属性值为Added或Modified的所有行的RowState属性都将被设置为UnChanged.任何标记为Deleted的DataRow对象将从DataSet中删除。调用RejectChanges时, 任何标记为Added的DataRow对象将会被从DataSet中删除, 其他修改过的DataRow对象将返回前一状态。

Clear:清除DataSet中所有DataRow对象。该方法比释放一个DataSet然后再创建一个相同结构的新DataSet要快。

Clone和**Copy**:使用Copy方法会创建与原DataSet具有相同结构和相同行的新DataSet。使用Clone方法会创建具有相同结构的新DataSet, 但不包含任何行。

GetChanges:返回与原DataSet对象具有相同结构的新DataSet, 并且还包含原DataSet中所有挂起更改的行。

GetXml和**GetXmlSchema**:使用GetXml方法得到由DataSet的内容与她的架构信息转换为XML格式后的字符串。如果只希望返回架构信息, 可以使用GetXmlSchema。

HasChange:表示DataSet中是否包含挂起更改的DataRow对象。

Merge:从另一个DataSet、DataTable或现有DataSet中的一组DataRow对象载入数据。

ReadXml和**WriteXml**:使用ReadXml方法从文件、TextReader、数据流或者XmlReader中将XML数据载入DataSet中。

Reset:将DataSet返回为未初始化状态。如果想放弃现有DataSet并且开始处理新的DataSet, 使用Reset方法比创建一个DataSet的新实例好。

③ 事件

MergeFailed:在DataSet的Merge方法发生一个异常时触发。

2、DataTable

①属性

②方法

③事件

ColumnChanged:在列的内容被改变之后触发

ColumnChanging:在列的内容被改变之前触发

RowChanged,RowChanging,RowDeleted,RowDeleting。

3、 DataColumn

①属性

4、 DataRow

①属性

HasError:确定行是否包含错误。

Item:通过指定行的列数，列的名称或DataColumn对象本身，访问列的内容。

ItemArray:获取或设置行中所有列的值。

RowError:返回一个包含行错误信息的字符串。

RowState:返回DataRowState枚举中的值来表示行的当前状态。

Table:返回DataRow对象所在的DataTable。

②方法

AcceptChanges和**RejectChanges**:提交和放弃挂起更改。

BeginEdit、CancelEdit、EndEdit

ClearErrors:清除DataRow中所有的错误。

Delete:Delete方法实际上并不从DataRow表的Row集合中删除该DataRow。当调用DataRow对象的Delete方法时，ADO.NET将该行标记为删除，之后调用SqlDataAdapter对象的Update方法来删除其在数据库中对应的行。

如果希望彻底删除DataRow，可以调用Delete方法，接着再调用它的AcceptChanges方法,还可以使用DataRowCollection对象的Remove方法完成相同的任务。

```
datasetds =new datast();
```

```
datatabletel =new datatable();
```

```
ds..tables.add(tel);
```

```
stringcode=ds.tables["tel"].rows[0][0].toString();
```


浅谈DataSet 的用法

DataSet 是ADO.NET开发人员为方便数据处理开发出来的，是数据的集合，是为解决DataReader的缺陷设计的，DataReader数据处理速度快，但它是只读的，而且一旦移到下一行，就不能查看上一行的数据，DataSet则可以自由移动指针。DataSet的数据是与数据库断开的。DataSet 还可用于多层应用程序中，如果应用程序运行在中间层的业务对象中来访问数据库，则业务对象需将脱机数据结构传递给客户应用程序。

DataSet的功能：浏览、排序、搜索、过滤、处理分级数据、缓存更改等。还可以与XML数据互换。DataSet中可包括多个 DataTable，可将多个查询结构存到一个DataSet中，方便操作，而DataTable中又包括多个DataRow、DataColumn，可通过这些 DataRow、DataColumn来查看、操作其中的数据，而需将操作结果返回给数据库的话，则可以调用DataAdapter的Update方法。

DataSet的操作：

```
DataSetds=new DataSet();  
DataTabledt=new DataTable("newTable");  
ds.Tables.Add(dt);  
DataSetds=new DataSet();  
DataTabledt=ds.Tables.Add("newTable");
```

上述两种方法都可以在DataSet中添加一个DataTable，看需要而进行选择。添加DataTable后，需向其中添加行和列。

```
DataSetds=new DataSet();  
DataTabledt=ds.Tables.Add("newTables");  
DataColumncol=dt.Columns.Add("newColumn",type  
of(int));  
col.AllowDBNull=false;  
col.MaxLength=4;  
col.Unique=true;
```

上述代码向DataSet中的DataTable中添加名为“newColumn”，类型为int且不为空，最大长度为4和唯一性为真的列。

```
dt.PrimaryKey=new DataColumn[]  
{dt.Columns["ID"]}
```

这段代码是继续上面代码的，为一个DataTable中添加一个主键列，主键列是一个数据组，如有多个主键，只需在数组中添加一个列即可。如下：

```
dt.PrimaryKey=new DataColumn[]  
{dt.Columns["OrderID"],dt.Columns["ProductID"]}
```

添加外键:

```
ForeignKeyConstraintfk;
```

```
fk=newForeignKeyConstraint(ds.Tables["Customers"]  
.Columns["CustomerID"],ds.Tables["Orders"].Colu  
mns["CustomerID"]);
```

```
ds.Tables["Orders"].Constraints.Add(fk);
```

上述代码假如已经为Customers表和Orders创建了主键，此句为添加外键约束。

上述是根据Customers表和Orders表的CustomerID来创建约束。

下面介绍修改DataRow中的内容:

```
DataRowdr=ds.Tables["Customer"].Rows.Find("ANT  
ON");
```

```
if(dr==null)
```

```
else
```

```
{
```

```
    dr.BeginEdit();
```

```
    dr["CompanyName"]="newValue";
```

```
    dr["ContactName"]="newValue2";
```

```
    dr.EndEdit();
```

```
}
```

上面代码通过Row集合的Find方法来在DataTable中的行进行定位，找到"ANTON"行，再修改"ANTON"行中CompanyName列和ContactName列的值。通过BeginEdit和EndEdit来缓存对行的修改，还可调用CancelEdit为取消修改。

判断某列是否为空值:

```
DataRowdr=ds.Tables["Customers"].Rows.Find("aaa  
");
```

```
if(dr.IsNull("ContactName");
```

```
..
```

```
else
```

```
    dr["ContactName"]=DBNull.Value
```

这里判断ContactName列是否为空，如果不是则为其赋空值，呵，很无厘头的做法，这里只为演示为列赋空值的做法。

删除DataRow:

有两种方法可以删除DataRow,Delete方法和Remove方法和RemoveAt方法。其区别是Delete方法实际上不是从DataTable 中删除掉一行,而是将其标志为删除,仅仅是做个记号,而Remove方法则是真正的从DataRow中删除一行,RemoveAt方法是根本行的索引来删除。列:

```
DataRow dr = ds.Tables["table"].Rows.Find("a");
```

```
ds.Tables["table"].Remove(dr);
```

或

```
ds.Tables["table"].Remove(index);
```

dr 为"a"所在的行,查出后将其删除,index为 "a"所在的索引号。关于DataSet中的其用法,参照MSDN

```
DataRow dr = ds.Tables["Customers"].Rows.Find("aaa");
```

```
if(dr.IsNull("ContactName"));
```

```
..
```

```
else
```

```
dr["ContactName"] = DBNull.Value
```

这里判断ContactName列是否为空,如果不是则为其赋空值,呵,很无厘头的做法,这里只为演示为列赋空值的做法。

```
using System.Data;
```

```
using System;
```

```
using System.Windows.Forms;
```

```
class DataT
```

```
{
```

```
static DataTable dt; // = new DataTable();
```

```
static DataSet ds;
```

```
static void method1()
```

```
{
```

```
dt = new DataTable("Name");
```

```
ds = new DataSet();
```

```
dt.Columns.Add(new DataColumn("ID",  
typeof(Int32)));
```

```
dt.Columns.Add(new DataColumn("Name",  
typeof(string)));
```

```
dt.Columns.Add(new DataColumn("Sex",  
typeof(string)));
```

```
dt.Columns.Add(new DataColumn("Addr",
```

```
typeof(string)));  
  
    }  
  
    static void add(int id,string  
name,stringsex,string addr)  
    {  
        DataRow dr = dt.NewRow();  
        dr["id"] = id;  
        dr["Name"] = name;  
        dr["Sex"] = sex;  
        dr["Addr"] = addr;  
        dt.Rows.Add(dr);  
    }  
  
    static void Main()  
    {  
        DataT DT = new DataT();  
        method1();  
        add(100,"Join","Male","北京");  
        add(101,"Lily","feMale","北京");  
        add(102,"JIM","Male","北京");  
        ds.Tables.Add(dt);  
        foreach(DataRow dr in dt.Rows)  
        {  
            MessageBox.Show(dr["ID"].ToString()+ " "  
+ dr["Name"].ToString() + " "  
+dr["Sex"].ToString() + " "  
+dr["Addr"].ToString(),"Message");  
  
            Console.WriteLine(dr["ID"].ToString()+ " "  
+ dr["Name"].ToString()+ " " +dr["Sex"].ToString()  
+ " " +dr["Addr"].ToString());  
        }  
        Try  
        {  
            foreach(DataTable dt2 in ds.Tables)  
            foreach(DataRow dr in dt2.Rows)  
            Console.WriteLine(dr["ID"].ToString()+ " "  
+ dr["Name"].ToString() + " " +
```

```
dr["Sex"].ToString()+ " " + dr["Addr"].ToString());  
    }  
    catch(Exception ex)  
    {  
        Console.WriteLine("dkfjksdjfk");  
    }  
}  
}
```

标签: C#

好文要顶

关注我

收藏该文

武胜-阿伟
关注 - 80
粉丝 - 198
[+加关注](#)

5 0

(请您对文章做出评价)

- « 上一篇: [AppConfigHelper](#)
- » 下一篇: [解决MySQL CPU占用100%的经验总结](#) 转

posted on 2012-06-08 09:47 武胜-阿伟 阅读(28430)
评论(1) 编辑 收藏

评论:

#1楼 2015-08-20 13:25 | 剑仙
沙发~

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站
首页。

- 最新IT新闻:
- [微软新规定: 五年不登录Xbox用户名就要丢](#)
 - [Ubuntu 16.04.1 LTS 桌面/服务器/云版本发布](#)
 - [庆祝地图服务升级 谷歌周末在加州推免费加油活动](#)
 - [雅虎48亿美元卖身Verizon 品牌将得到保留](#)

- [中国版无人驾驶规范要来了，标准主要有四个部分](#)
- » [更多新闻...](#)

最新知识库文章：

- [可是姑娘，你为什么要编程呢？](#)
- [知其所以然（以算法学习为例）](#)
- [如何给变量取个简短且无歧义的名字](#)
- [编程的智慧](#)
- [写给初学前端工程师的一封信](#)
- » [更多知识库文章...](#)

Powered by: [博客园](#) 模板提供: [沪江博客](#) Copyright ©2016
武胜-阿伟