

# 服务端文档

制作者 Doxygen 1.9.1



<b>1 命名空间索引</b>	<b>1</b>
1.1 命名空间列表	1
<b>2 继承关系索引</b>	<b>3</b>
2.1 类继承关系	3
<b>3 类索引</b>	<b>5</b>
3.1 类列表	5
<b>4 文件索引</b>	<b>7</b>
4.1 文件列表	7
<b>5 命名空间文档</b>	<b>9</b>
5.1 Ui 命名空间参考	9
<b>6 类说明</b>	<b>11</b>
6.1 instruction类 参考	11
6.1.1 详细描述	12
6.1.2 构造及析构造函数说明	12
6.1.2.1 instruction()	12
6.1.3 成员函数说明	12
6.1.3.1 getResult()	12
6.1.3.2 parseCart()	12
6.1.3.3 parseClass()	13
6.1.3.4 parseDiscount()	13
6.1.3.5 parseJson()	13
6.1.3.6 parseQuery()	14
6.1.4 类成员变量说明	14
6.1.4.1 cartMethods	14
6.1.4.2 classes	14
6.1.4.3 discountMethods	14
6.1.4.4 json	15
6.1.4.5 jsonMethods	15
6.1.4.6 queryMethod	15
6.1.4.7 result	15
6.1.4.8 sqlC	15
6.1.4.9 sqlD	15
6.1.4.10 sqlQ	16
6.2 jsonexe类 参考	16
6.2.1 详细描述	16
6.2.2 构造及析构造函数说明	16
6.2.2.1 jsonexe()	17
6.2.2.2 ~jsonexe()	17
6.2.3 成员函数说明	17

6.2.3.1 appendJson()	17
6.2.3.2 editJson()	17
6.2.3.3 getJson()	18
6.2.3.4 judgeJson()	18
6.2.4 类成员变量说明	18
6.2.4.1 file	19
6.2.4.2 jdoc	19
6.2.4.3 path	19
6.2.4.4 users	19
6.3 MainWindow类 参考	19
6.4 processPool类 参考	20
6.4.1 详细描述	20
6.4.2 构造及析构造函数说明	20
6.4.2.1 processPool()	20
6.4.2.2 ~processPool()	21
6.4.3 成员函数说明	21
6.4.3.1 addPool()	21
6.4.3.2 clearPool()	21
6.4.3.3 erasePool	21
6.4.4 类成员变量说明	22
6.4.4.1 maxSize	22
6.4.4.2 nextID	22
6.4.4.3 nowSize	22
6.4.4.4 pool	22
6.5 sqlCart类 参考	22
6.5.1 详细描述	23
6.5.2 构造及析构造函数说明	23
6.5.2.1 sqlCart()	23
6.5.2.2 ~sqlCart()	23
6.5.3 成员函数说明	23
6.5.3.1 deletedb()	23
6.5.3.2 insertdb()	24
6.5.3.3 selectdb()	24
6.5.3.4 updatedb()	25
6.5.4 类成员变量说明	25
6.5.4.1 db	25
6.5.4.2 delete_all.sql	25
6.5.4.3 delete.sql	25
6.5.4.4 insert.sql	26
6.5.4.5 query	26
6.5.4.6 select.sql	26
6.5.4.7 update.sql	26

6.6 sqlDiscount类 参考	26
6.6.1 详细描述	27
6.6.2 构造及析构造函数说明	27
6.6.2.1 sqlDiscount()	27
6.6.2.2 ~sqlDiscount()	27
6.6.3 成员函数说明	27
6.6.3.1 insertdb()	27
6.6.3.2 selectdb()	28
6.6.3.3 updatedb()	28
6.6.4 类成员变量说明	29
6.6.4.1 db	29
6.6.4.2 delete_sql	29
6.6.4.3 insert_sql	29
6.6.4.4 query	29
6.6.4.5 select_sql	29
6.6.4.6 update_sql	30
6.7 sqlQuery类 参考	30
6.7.1 详细描述	31
6.7.2 构造及析构造函数说明	31
6.7.2.1 sqlQuery()	31
6.7.2.2 ~sqlQuery()	31
6.7.3 成员函数说明	31
6.7.3.1 deletedb()	31
6.7.3.2 insertdb()	32
6.7.3.3 outPut()	32
6.7.3.4 searchdb()	32
6.7.3.5 selectdb()	33
6.7.3.6 selectStoredb()	33
6.7.3.7 transform()	33
6.7.3.8 updatedb()	34
6.7.3.9 updateResetDb()	34
6.7.4 类成员变量说明	35
6.7.4.1 create_sql	35
6.7.4.2 db	35
6.7.4.3 delete_sql	35
6.7.4.4 insert_sql	35
6.7.4.5 query	35
6.7.4.6 search_sql	36
6.7.4.7 select_all_sql	36
6.7.4.8 select_all_type	36
6.7.4.9 select_store_sql	36
6.7.4.10 select_store_type	36

6.7.4.11 typeMap	36
6.7.4.12 update_reserved_sql	37
6.7.4.13 update_reset_sql	37
6.7.4.14 update_sql	37
6.8 thread类 参考	37
6.8.1 详细描述	38
6.8.2 构造及析构函数说明	38
6.8.2.1 thread()	38
6.8.3 成员函数说明	38
6.8.3.1 delThread	38
6.8.3.2 disconnect	39
6.8.3.3 getID()	39
6.8.3.4 iniSocket()	39
6.8.3.5 recData	39
6.8.3.6 sendData()	39
6.8.4 类成员变量说明	40
6.8.4.1 id	40
6.8.4.2 in	40
6.8.4.3 ins	40
6.8.4.4 socket	40
7 文件说明	41
7.1 D:/Document/VScode/C++/eCommerce/server/instruction.cpp 文件参考	41
7.2 D:/Document/VScode/C++/eCommerce/server/instruction.h 文件参考	41
7.3 D:/Document/VScode/C++/eCommerce/server/lib/global.h 文件参考	41
7.3.1 宏定义说明	42
7.3.1.1 AppendJson	42
7.3.1.2 Deletedb	42
7.3.1.3 EditJson	42
7.3.1.4 GetJson	43
7.3.1.5 Insertdb	43
7.3.1.6 JsonExe	43
7.3.1.7 JudgeJson	43
7.3.1.8 OutPut	43
7.3.1.9 Searchdb	43
7.3.1.10 Selectdb	44
7.3.1.11 SelectStoredb	44
7.3.1.12 SqlCart	44
7.3.1.13 SqlDiscount	44
7.3.1.14 SqlQuery	44
7.3.1.15 Updatedb	44
7.3.1.16 UpdateResetDb	45

---

7.4 D:/Document/VScode/C++/eCommerce/server/lib/jsonexe.cpp 文件参考 . . . . .	45
7.5 D:/Document/VScode/C++/eCommerce/server/lib/jsonexe.h 文件参考 . . . . .	45
7.6 D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.cpp 文件参考 . . . . .	45
7.7 D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.h 文件参考 . . . . .	45
7.8 D:/Document/VScode/C++/eCommerce/server/lib/sqldiscount.cpp 文件参考 . . . . .	46
7.9 D:/Document/VScode/C++/eCommerce/server/lib/sqldiscount.h 文件参考 . . . . .	46
7.10 D:/Document/VScode/C++/eCommerce/server/lib/sqlquery.cpp 文件参考 . . . . .	46
7.11 D:/Document/VScode/C++/eCommerce/server/lib/sqlquery.h 文件参考 . . . . .	46
7.12 D:/Document/VScode/C++/eCommerce/server/main.cpp 文件参考 . . . . .	47
7.12.1 函数说明 . . . . .	47
7.12.1.1 main() . . . . .	47
7.13 D:/Document/VScode/C++/eCommerce/server/processpool.cpp 文件参考 . . . . .	47
7.14 D:/Document/VScode/C++/eCommerce/server/processpool.h 文件参考 . . . . .	47
7.15 D:/Document/VScode/C++/eCommerce/server/server.cpp 文件参考 . . . . .	48
7.16 D:/Document/VScode/C++/eCommerce/server/server.h 文件参考 . . . . .	48
7.17 D:/Document/VScode/C++/eCommerce/server/thread.cpp 文件参考 . . . . .	48
7.18 D:/Document/VScode/C++/eCommerce/server/thread.h 文件参考 . . . . .	48





## Chapter 1

# 命名空间索引

### 1.1 命名空间列表

这里列出了所有命名空间定义,附带简要说明:

Ui .....	9
----------	---



## Chapter 2

# 继承关系索引

### 2.1 类继承关系

此继承关系列表按字典顺序粗略的排序:

instruction . . . . .	11
jsonexe . . . . .	16
QMainWindow	
MainWindow . . . . .	19
QObject	
processPool . . . . .	20
thread . . . . .	37
sqlCart . . . . .	22
sqlDiscount . . . . .	26
sqlQuery . . . . .	30



## Chapter 3

# 类索引

### 3.1 类列表

这里列出了所有类、结构、联合以及接口定义等，并附带简要说明：

<a href="#">instruction</a>	序列化操作指令类 . . . . .	11
<a href="#">jsonexe</a>	JSON文件的操作类 . . . . .	16
<a href="#">MainWindow</a>	服务端类 . . . . .	19
<a href="#">processPool</a>	进程池 . . . . .	20
<a href="#">sqlCart</a>	购物车数据库类 . . . . .	22
<a href="#">sqlDiscount</a>	折扣数据库类 . . . . .	26
<a href="#">sqlQuery</a>	商品数据库类 . . . . .	30
<a href="#">thread</a>	自定义线程 . . . . .	37



## Chapter 4

# 文件索引

### 4.1 文件列表

这里列出了所有文件，并附带简要说明:

D:/Document/VScode/C++/eCommerce/server/instruction.cpp	41
D:/Document/VScode/C++/eCommerce/server/instruction.h	41
D:/Document/VScode/C++/eCommerce/server/main.cpp	47
D:/Document/VScode/C++/eCommerce/server/processpool.cpp	47
D:/Document/VScode/C++/eCommerce/server/processpool.h	47
D:/Document/VScode/C++/eCommerce/server/server.cpp	48
D:/Document/VScode/C++/eCommerce/server/server.h	48
D:/Document/VScode/C++/eCommerce/server/thread.cpp	48
D:/Document/VScode/C++/eCommerce/server/thread.h	48
D:/Document/VScode/C++/eCommerce/server/lib/global.h	41
D:/Document/VScode/C++/eCommerce/server/lib/jsonexe.cpp	45
D:/Document/VScode/C++/eCommerce/server/lib/jsonexe.h	45
D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.cpp	45
D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.h	45
D:/Document/VScode/C++/eCommerce/server/lib/sqldiscount.cpp	46
D:/Document/VScode/C++/eCommerce/server/lib/sqldiscount.h	46
D:/Document/VScode/C++/eCommerce/server/lib/sqlquery.cpp	46
D:/Document/VScode/C++/eCommerce/server/lib/sqlquery.h	46





## Chapter 5

# 命名空间文档

### 5.1 Ui 命名空间参考



## Chapter 6

# 类说明

### 6.1 instruction类 参考

序列化操作指令类

```
#include <instruction.h>
```

instruction 的协作图:

#### Public 成员函数

- [instruction](#) ()
- QString [getResult](#) ()  
*getResult* 获取服务端传输数据的结果
- void [parseClass](#) (QStringList sL)  
*parseClass* 序列化调用的类名
- void [parseJson](#) (QString method, QStringList params)  
*parseJson* 序列化调用的JSON类
- void [parseQuery](#) (QString method, QStringList params)  
*parseQuery* 序列化调用的sqlQuery类
- void [parseDiscount](#) (QString method, QStringList params)  
*parseDiscount* 序列化调用的sqlDiscount类
- void [parseCart](#) (QString method, QStringList params)  
*parseCart* 序列化调用的sqlCart类

#### Private 属性

- QStringList [classes](#)
- QStringList [jsonMethods](#)
- QStringList [queryMethod](#)
- QStringList [discountMethods](#)
- QStringList [cartMethods](#)
- QByteArray [result](#)
- class [jsonexe](#) \* [json](#)
- class [sqlQuery](#) \* [sqlQ](#)
- class [sqlDiscount](#) \* [sqlD](#)
- class [sqlCart](#) \* [sqlC](#)

### 6.1.1 详细描述

序列化操作指令类

在文件 `instruction.h` 第 14 行定义.

### 6.1.2 构造及析构函数说明

#### 6.1.2.1 `instruction()`

```
instruction::instruction ( )
```

在文件 `instruction.cpp` 第 3 行定义.

### 6.1.3 成员函数说明

#### 6.1.3.1 `getResult()`

```
QString instruction::getResult ( )
```

**getResult** 获取服务端传输数据的结果

返回

在文件 `instruction.cpp` 第 18 行定义.

#### 6.1.3.2 `parseCart()`

```
void instruction::parseCart (
    QString method,
    QStringList params )
```

**parseCart** 序列化调用的`sqlCart`类

参数

<i>method</i>	调用的方法
<i>params</i>	传入的参数

在文件 `instruction.cpp` 第 129 行定义.

### 6.1.3.3 `parseClass()`

```
void instruction::parseClass (
    QStringList sL )
```

`parseClass` 序列化调用的类名

参数

<i>sL</i>	操作指令
-----------	------

在文件 `instruction.cpp` 第 23 行定义.

### 6.1.3.4 `parseDiscount()`

```
void instruction::parseDiscount (
    QString method,
    QStringList params )
```

`parseDiscount` 序列化调用的`sqlDiscount`类

参数

<i>method</i>	调用的方法
<i>params</i>	传入的参数

在文件 `instruction.cpp` 第 109 行定义.

### 6.1.3.5 `parseJson()`

```
void instruction::parseJson (
    QString method,
    QStringList params )
```

`parseJson` 序列化调用的JSON类

参数

<i>method</i>	调用的方法
<i>params</i>	传入的参数

在文件 `instruction.cpp` 第 48 行定义.

#### 6.1.3.6 `parseQuery()`

```
void instruction::parseQuery (
    QString method,
    QStringList params )
```

`parseQuery` 序列化调用的`sqlQuery`类

参数

<i>method</i>	调用的方法
<i>params</i>	传入的参数

在文件 `instruction.cpp` 第 72 行定义.

### 6.1.4 类成员变量说明

#### 6.1.4.1 `cartMethods`

```
QStringList instruction::cartMethods [private]
```

在文件 `instruction.h` 第 57 行定义.

#### 6.1.4.2 `classes`

```
QStringList instruction::classes [private]
```

在文件 `instruction.h` 第 53 行定义.

#### 6.1.4.3 `discountMethods`

```
QStringList instruction::discountMethods [private]
```

在文件 `instruction.h` 第 56 行定义.

#### 6.1.4.4 json

```
class jsonexe* instruction::json [private]
```

在文件 instruction.h 第 60 行定义.

#### 6.1.4.5 jsonMethods

```
QStringList instruction::jsonMethods [private]
```

在文件 instruction.h 第 54 行定义.

#### 6.1.4.6 queryMethod

```
QStringList instruction::queryMethod [private]
```

在文件 instruction.h 第 55 行定义.

#### 6.1.4.7 result

```
QByteArray instruction::result [private]
```

在文件 instruction.h 第 58 行定义.

#### 6.1.4.8 sqlC

```
class sqlCart* instruction::sqlC [private]
```

在文件 instruction.h 第 63 行定义.

#### 6.1.4.9 sqlD

```
class sqlDiscount* instruction::sqlD [private]
```

在文件 instruction.h 第 62 行定义.

#### 6.1.4.10 sqlQ

```
class sqlQuery* instruction::sqlQ [private]
```

在文件 `instruction.h` 第 61 行定义.

该类的文档由以下文件生成:

- D:/Document/VScode/C++/eCommerce/server/[instruction.h](#)
- D:/Document/VScode/C++/eCommerce/server/[instruction.cpp](#)

## 6.2 jsonexe类 参考

JSON文件的操作类

```
#include <jsonexe.h>
```

### Public 成员函数

- [jsonexe](#) ()
- [~jsonexe](#) ()
- QByteArray [getJSON](#) (QStringList sL)  
获取某用户的信息
- bool [judgeJson](#) (QString username)  
*judgeJson* 判断是否存在某用户
- bool [editJson](#) (QStringList sL)  
编辑用户 *uu* 的信息
- bool [appendJson](#) (QStringList sL)  
*appendJson* 添加

### Private 属性

- QString [path](#)
- QFile [file](#)
- QJsonDocument [jdoc](#)
- QJsonArray [users](#)

#### 6.2.1 详细描述

JSON文件的操作类

在文件 `jsonexe.h` 第 14 行定义.

#### 6.2.2 构造及析构函数说明



### 6.2.2.1 jsonexe()

```
jsonexe::jsonexe ( )
```

在文件 jsonexe.cpp 第 4 行定义.

### 6.2.2.2 ~jsonexe()

```
jsonexe::~~jsonexe ( )
```

在文件 jsonexe.cpp 第 11 行定义.

## 6.2.3 成员函数说明

### 6.2.3.1 appendJson()

```
bool jsonexe::appendJson (
    QStringList sL )
```

appendJson 添加

参数

<code>sL[0]-&gt;username;sL[1]-&gt;password;sL[2]-&gt;type</code>
---

返回

在文件 jsonexe.cpp 第 99 行定义.

### 6.2.3.2 editJson()

```
bool jsonexe::editJson (
    QStringList sL )
```

编辑用户uu的信息

参数

<code>sL[0]-&gt;username;sL[1]-&gt;acount;sL[2]score;sL[3]-&gt;password</code>
--

返回

是否编辑成功

在文件 jsonexe.cpp 第 58 行定义.

### 6.2.3.3 getJson()

```
QByteArray jsonexe::getJson (
    QStringList sL )
```

获取某用户的信息

参数

<code>sL[0]-&gt;username;sL[1]-&gt;password</code>	
--	--

返回

某用户的对象字符串

在文件 jsonexe.cpp 第 16 行定义.

### 6.2.3.4 judgeJson()

```
bool jsonexe::judgeJson (
    QString username )
```

judgeJson 判断是否存在某用户

参数

<code>username</code>	用户名
-----------------------	-----

返回

bool是否存在

在文件 jsonexe.cpp 第 38 行定义.

## 6.2.4 类成员变量说明

#### 6.2.4.1 file

```
QFile jsonexe::file [private]
```

在文件 jsonexe.h 第 49 行定义.

#### 6.2.4.2 jdoc

```
QJsonDocument jsonexe::jdoc [private]
```

在文件 jsonexe.h 第 50 行定义.

#### 6.2.4.3 path

```
QString jsonexe::path [private]
```

在文件 jsonexe.h 第 48 行定义.

#### 6.2.4.4 users

```
QJsonArray jsonexe::users [private]
```

在文件 jsonexe.h 第 51 行定义.

该类的文档由以下文件生成:

- [D:/Document/Vscode/C++/eCommerce/server/lib/jsonexe.h](#)
- [D:/Document/Vscode/C++/eCommerce/server/lib/jsonexe.cpp](#)

## 6.3 MainWindow类 参考

服务端类

```
#include <server.h>
```

类 MainWindow 继承关系图:

## 6.4 processPool类 参考

进程池

```
#include <processpool.h>
```

类 processPool 继承关系图:

processPool 的协作图:

### Public 成员函数

- `processPool` (int max=50)
- `~processPool` ()
- bool `addPool` (QTcpSocket \*socket)  
*addPool* 向进程池中添加某个 *socket* 进程
- void `clearPool` ()  
*clearPool* 清空进程池

### Private 槽

- void `erasePool` (int)  
*erasePool* 获取线程的信号，根据进程ID，删除某个 *socket*

### Private 属性

- QVector< class `thread` \* > `pool`
- int `nowSize`
- int `maxSize`
- int `nextID`

#### 6.4.1 详细描述

进程池

在文件 processpool.h 第 11 行定义.

#### 6.4.2 构造及析构函数说明

##### 6.4.2.1 processPool()

```
processPool::processPool (  
    int max = 50 )
```

在文件 processpool.cpp 第 3 行定义.

### 6.4.2.2 ~processPool()

```
processPool::~~processPool ( )
```

在文件 processpool.cpp 第 10 行定义.

## 6.4.3 成员函数说明

### 6.4.3.1 addPool()

```
bool processPool::addPool (
    QTcpSocket * socket )
```

addPool 向进程池中添加某个socket进程

参数

<i>socket</i>	客户端的套接字
---------------	---------

返回

是否添加成功

在文件 processpool.cpp 第 28 行定义.

### 6.4.3.2 clearPool()

```
void processPool::clearPool ( )
```

clearPool 清空进程池

在文件 processpool.cpp 第 47 行定义.

### 6.4.3.3 erasePool

```
void processPool::erasePool (
    int processId ) [private], [slot]
```

erasePool 获取线程的信号，根据进程ID，删除某个socket

在文件 processpool.cpp 第 16 行定义.

### 6.4.4 类成员变量说明

#### 6.4.4.1 maxSize

```
int processPool::maxSize [private]
```

在文件 `processpool.h` 第 16 行定义.

#### 6.4.4.2 nextID

```
int processPool::nextID [private]
```

在文件 `processpool.h` 第 16 行定义.

#### 6.4.4.3 nowSize

```
int processPool::nowSize [private]
```

在文件 `processpool.h` 第 16 行定义.

#### 6.4.4.4 pool

```
QVector<class thread*> processPool::pool [private]
```

在文件 `processpool.h` 第 15 行定义.

该类的文档由以下文件生成:

- [D:/Document/VScode/C++/eCommerce/server/processpool.h](#)
- [D:/Document/VScode/C++/eCommerce/server/processpool.cpp](#)

## 6.5 sqlCart类 参考

购物车数据库类

```
#include <sqlcart.h>
```

## Public 成员函数

- `sqlCart ()`
- `~sqlCart ()`
- `QByteArray selectdb (QString username)`  
*selectdb* 获取某个用户下的购物车商品
- `bool insertdb (QStringList sL)`  
*insertdb* 向购物车数据库添加某个商品
- `bool updatedb (QStringList sL)`  
*updatedb* 更新某个用户的某个商品的购买量
- `bool deletedb (QStringList sL)`  
*deletedb*

## Private 属性

- `QSqlDatabase db`
- `QSqlQuery query`
- `QString insert_sql`
- `QString update_sql`
- `QString delete_sql`
- `QString delete_all_sql`
- `QString select_sql`

### 6.5.1 详细描述

购物车数据库类

在文件 `sqlcart.h` 第 18 行定义.

### 6.5.2 构造及析构函数说明

#### 6.5.2.1 sqlCart()

```
sqlCart::sqlCart ( )
```

在文件 `sqlcart.cpp` 第 3 行定义.

#### 6.5.2.2 ~sqlCart()

```
sqlCart::~sqlCart ( )
```

### 6.5.3 成员函数说明

#### 6.5.3.1 deletedb()

```
bool sqlCart::deletedb (
    QStringList sL )
```

`deletedb`

参数

两种情况一个是全删除，一个是删除某个商品	
----------------------	--

返回

是否删除成功

在文件 `sqlcart.cpp` 第 103 行定义.

### 6.5.3.2 insertdb()

```
bool sqlCart::insertdb (  
    QStringList sL )
```

**insertdb** 向购物车数据库添加某个商品

参数

<code>(sL[0]~sL[4])</code>	=> (name,price,number,store,user)
----------------------------	-----------------------------------

返回

是否添加成功

在文件 `sqlcart.cpp` 第 51 行定义.

### 6.5.3.3 selectdb()

```
QByteArray sqlCart::selectdb (  
    QString username )
```

**selectdb** 获取某个用户下的购物车商品

参数

<code>username</code>	用户名
-----------------------	-----

返回

商品列表Json信息

在文件 `sqlcart.cpp` 第 19 行定义.



#### 6.5.3.4 updatedb()

```
bool sqlCart::updatedb (
    QStringList sL )
```

**updatedb** 更新某个用户的某个商品的购买量

参数

<code>(sL[0]~sL[2])</code>	<code>=&gt; (number,name,user)</code>
----------------------------	---------------------------------------

返回

是否更新成功

在文件 `sqlcart.cpp` 第 80 行定义.

#### 6.5.4 类成员变量说明

##### 6.5.4.1 db

```
QSqlDatabase sqlCart::db [private]
```

在文件 `sqlcart.h` 第 49 行定义.

##### 6.5.4.2 delete\_all\_sql

```
QString sqlCart::delete_all_sql [private]
```

在文件 `sqlcart.h` 第 53 行定义.

##### 6.5.4.3 delete\_sql

```
QString sqlCart::delete_sql [private]
```

在文件 `sqlcart.h` 第 53 行定义.

#### 6.5.4.4 insert\_sql

```
QString sqlCart::insert_sql [private]
```

在文件 `sqlcart.h` 第 51 行定义.

#### 6.5.4.5 query

```
QSqlQuery sqlCart::query [private]
```

在文件 `sqlcart.h` 第 50 行定义.

#### 6.5.4.6 select\_sql

```
QString sqlCart::select_sql [private]
```

在文件 `sqlcart.h` 第 54 行定义.

#### 6.5.4.7 update\_sql

```
QString sqlCart::update_sql [private]
```

在文件 `sqlcart.h` 第 52 行定义.

该类的文档由以下文件生成:

- [D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.h](#)
- [D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.cpp](#)

## 6.6 sqlDiscount类 参考

折扣数据库类

```
#include <sqldiscount.h>
```

### Public 成员函数

- [sqlDiscount \(\)](#)
- [~sqlDiscount \(\)](#)
- int [selectdb](#) (QString name, QString store)  
*selectdb* 获取某个店某个商品的折扣
- bool [insertdb](#) (QString proType, QString discount, QString store)  
*insertdb* 插入某个商品的折扣
- bool [updatedb](#) (QString proType, QString discount, QString store)  
*updatedb* 更新某个商品的折扣

## Private 属性

- QSqlDatabase [db](#)
- QSqlQuery [query](#)
- QString [insert\\_sql](#)
- QString [update\\_sql](#)
- QString [delete\\_sql](#)
- QString [select\\_sql](#)

### 6.6.1 详细描述

折扣数据库类

在文件 `sqldiscount.h` 第 14 行定义.

### 6.6.2 构造及析构函数说明

#### 6.6.2.1 QSqlDiscount()

```
sqlDiscount::sqlDiscount ( )
```

在文件 `sqldiscount.cpp` 第 3 行定义.

#### 6.6.2.2 ~sqlDiscount()

```
sqlDiscount::~~sqlDiscount ( )
```

### 6.6.3 成员函数说明

#### 6.6.3.1 insertdb()

```
bool sqlDiscount::insertdb (
    QString proType,
    QString discount,
    QString store )
```

`insertdb` 插入某个商品的折扣

参数

<i>proType</i>	商品类型
<i>discount</i>	折扣数
<i>store</i>	店名

返回

是否插入成功

在文件 `sqldiscount.cpp` 第 42 行定义.

### 6.6.3.2 selectdb()

```
int sqlDiscount::selectdb (
    QString name,
    QString store )
```

`selectdb` 获取某个店某个商品的折扣

参数

<i>name</i>	商品名
<i>store</i>	点名

返回

折扣1-10

在文件 `sqldiscount.cpp` 第 18 行定义.

### 6.6.3.3 updatedb()

```
bool sqlDiscount::updatedb (
    QString proType,
    QString discount,
    QString store )
```

`updatedb` 更新某个商品的折扣

参数

<i>proType</i>	商品类型
<i>discount</i>	折扣数
<i>store</i>	店名

返回

是否更新成功

在文件 `sqldiscount.cpp` 第 71 行定义.

## 6.6.4 类成员变量说明

### 6.6.4.1 db

```
QSqlDatabase sqlDiscount::db [private]
```

在文件 `sqldiscount.h` 第 43 行定义.

### 6.6.4.2 delete\_sql

```
QString sqlDiscount::delete_sql [private]
```

在文件 `sqldiscount.h` 第 47 行定义.

### 6.6.4.3 insert\_sql

```
QString sqlDiscount::insert_sql [private]
```

在文件 `sqldiscount.h` 第 45 行定义.

### 6.6.4.4 query

```
QSqlQuery sqlDiscount::query [private]
```

在文件 `sqldiscount.h` 第 44 行定义.

### 6.6.4.5 select\_sql

```
QString sqlDiscount::select_sql [private]
```

在文件 `sqldiscount.h` 第 48 行定义.

#### 6.6.4.6 update\_sql

QString sqlDiscount::update\_sql [private]

在文件 sqldiscount.h 第 46 行定义.

该类的文档由以下文件生成:

- D:/Document/Vscode/C++/eCommerce/server/lib/sqldiscount.h
- D:/Document/Vscode/C++/eCommerce/server/lib/sqldiscount.cpp

## 6.7 sqlQuery类 参考

商品数据库类

```
#include <sqlquery.h>
```

### Public 成员函数

- [sqlQuery \(\)](#)
- [~sqlQuery \(\)](#)
- QByteArray [selectdb](#) (QString proType)  
*selectdb* 获取某个类型的商品列表
- QByteArray [selectStoredb](#) (QString proType, QString store)  
*selectStoredb* 获取某个商品某个类型的商品列表
- QByteArray [searchdb](#) (QString searchContent)  
*searchdb* 根据关键词模糊搜索得到的商品列表
- QByteArray [transform](#) (QSqlQuery query)  
*transform* 将 *QsqlQuery* 的内容转换为 *QByte* 类型
- void [outPut](#) ()  
*outPut* 输入数据库内容
- bool [insertdb](#) (QStringList sL)  
*insertdb* 插入数据库内容
- bool [updatedb](#) (QStringList sL)  
*updatedb* 更新某个商品列表
- bool [updateResetDb](#) (QString name, QString store)  
*updateResetDb* 重置某商店某商品的预定量为 0
- bool [deletedb](#) (QString name, QString store)  
*deletedb* 删除某个商店的某个商品

### Private 属性

- QStringList [typeMap](#)
- QSqlDatabase [db](#)
- QSqlQuery [query](#)
- QString [create\\_sql](#)
- QString [insert\\_sql](#)
- QString [update\\_sql](#)
- QString [update\\_reserved\\_sql](#)
- QString [update\\_reset\\_sql](#)
- QString [delete\\_sql](#)
- QString [select\\_all\\_sql](#)
- QString [select\\_all\\_type](#)
- QString [select\\_store\\_sql](#)
- QString [select\\_store\\_type](#)
- QString [search\\_sql](#)

### 6.7.1 详细描述

商品数据库类

在文件 `sqlquery.h` 第 17 行定义.

### 6.7.2 构造及析构函数说明

#### 6.7.2.1 sqlQuery()

```
sqlQuery::sqlQuery ( )
```

在文件 `sqlquery.cpp` 第 4 行定义.

#### 6.7.2.2 ~sqlQuery()

```
sqlQuery::~sqlQuery ( )
```

在文件 `sqlquery.cpp` 第 27 行定义.

### 6.7.3 成员函数说明

#### 6.7.3.1 deletedb()

```
bool sqlQuery::deletedb (
    QString name,
    QString store )
```

`deletedb` 删除某个商店的某个商品

参数

<i>name</i>	商品名
<i>store</i>	店名

返回

是否删除成功

在文件 `sqlquery.cpp` 第 226 行定义.

### 6.7.3.2 insertdb()

```
bool sqlQuery::insertdb (
    QStringList sL )
```

`insertdb` 插入数据库内容

参数

<code>(sL[0]~sL[6])</code>	=> (name,intro,price,rest,type,store,reserved)
----------------------------	--

返回

是否插入成功

在文件 `sqlquery.cpp` 第 141 行定义.

### 6.7.3.3 outPut()

```
void sqlQuery::outPut ( )
```

`outPut` 输入数据库内容

在文件 `sqlquery.cpp` 第 113 行定义.

### 6.7.3.4 searchdb()

```
QByteArray sqlQuery::searchdb (
    QString searchContent )
```

`searchdb` 根据关键词模糊搜索得到的商品列表

参数

<code>searchContent</code>	商品关键词
----------------------------	-------

返回

商品列表的JSON



在文件 `sqlquery.cpp` 第 76 行定义.

### 6.7.3.5 selectdb()

```
QByteArray sqlQuery::selectdb (
    QString proType )
```

`selectdb` 获取某个类型的商品列表

参数

<i>proType</i>	商品类型
----------------	------

返回

商品列表的JSON

在文件 `sqlquery.cpp` 第 32 行定义.

### 6.7.3.6 selectStoredb()

```
QByteArray sqlQuery::selectStoredb (
    QString proType,
    QString store )
```

`selectStoredb` 获取某个商品某个类型的商品列表

参数

<i>proType</i>	商品类型
<i>store</i>	商家名

返回

商品列表的JSON

在文件 `sqlquery.cpp` 第 53 行定义.

### 6.7.3.7 transform()

```
QByteArray sqlQuery::transform (
    QSqlQuery query )
```

`transform` 将`QsqlQuery`的内容转换为`QByte`类型

参数

<i>query</i>	sql的结果
--------------	--------

返回

商品列表的JSON

在文件 `sqlquery.cpp` 第 94 行定义.

#### 6.7.3.8 updatedb()

```
bool sqlQuery::updatedb (
    QStringList sL )
```

`updatedb` 更新某个商品列表

参数

两种类型，一种是更新展示的信息，一种是更新预定量和库存	
-----------------------------	--

返回

是否更新成功

在文件 `sqlquery.cpp` 第 168 行定义.

#### 6.7.3.9 updateResetDb()

```
bool sqlQuery::updateResetDb (
    QString name,
    QString store )
```

`updateResetDb` 重置某商店某商品的预定量为0

参数

<i>name</i>	商品名
<i>store</i>	预定量

返回

是否更新成功

在文件 `sqlquery.cpp` 第 203 行定义.

## 6.7.4 类成员变量说明

### 6.7.4.1 `create_sql`

```
QString sqlQuery::create_sql [private]
```

在文件 `sqlquery.h` 第 81 行定义.

### 6.7.4.2 `db`

```
QSqlDatabase sqlQuery::db [private]
```

在文件 `sqlquery.h` 第 79 行定义.

### 6.7.4.3 `delete_sql`

```
QString sqlQuery::delete_sql [private]
```

在文件 `sqlquery.h` 第 84 行定义.

### 6.7.4.4 `insert_sql`

```
QString sqlQuery::insert_sql [private]
```

在文件 `sqlquery.h` 第 82 行定义.

### 6.7.4.5 `query`

```
QSqlQuery sqlQuery::query [private]
```

在文件 `sqlquery.h` 第 80 行定义.

#### 6.7.4.6 search\_sql

```
QString sqlQuery::search_sql [private]
```

在文件 sqlquery.h 第 87 行定义.

#### 6.7.4.7 select\_all\_sql

```
QString sqlQuery::select_all_sql [private]
```

在文件 sqlquery.h 第 85 行定义.

#### 6.7.4.8 select\_all\_type

```
QString sqlQuery::select_all_type [private]
```

在文件 sqlquery.h 第 85 行定义.

#### 6.7.4.9 select\_store\_sql

```
QString sqlQuery::select_store_sql [private]
```

在文件 sqlquery.h 第 86 行定义.

#### 6.7.4.10 select\_store\_type

```
QString sqlQuery::select_store_type [private]
```

在文件 sqlquery.h 第 86 行定义.

#### 6.7.4.11 typeMap

```
QStringList sqlQuery::typeMap [private]
```

在文件 sqlquery.h 第 78 行定义.

#### 6.7.4.12 update\_reserved\_sql

```
QString sqlQuery::update_reserved_sql [private]
```

在文件 `sqlquery.h` 第 83 行定义.

#### 6.7.4.13 update\_reset\_sql

```
QString sqlQuery::update_reset_sql [private]
```

在文件 `sqlquery.h` 第 83 行定义.

#### 6.7.4.14 update\_sql

```
QString sqlQuery::update_sql [private]
```

在文件 `sqlquery.h` 第 83 行定义.

该类的文档由以下文件生成:

- D:/Document/Vscode/C++/eCommerce/server/lib/[sqlquery.h](#)
- D:/Document/Vscode/C++/eCommerce/server/lib/[sqlquery.cpp](#)

## 6.8 thread类 参考

自定义线程

```
#include <thread.h>
```

类 `thread` 继承关系图:

`thread` 的协作图:

信号

- void [delThread](#) (int)  
*delThread* 信号: 触发进程池删除进程

### Public 成员函数

- [thread](#) (QTcpSocket \*newSocket, int id)
- void [iniSocket](#) ()  
*iniSocket* 初始化线程
- int [getID](#) ()  
*getID* 获取线程ID
- void [sendData](#) (QString sendMsg)  
*sendData* 向客户端发送数据

## Private 槽

- void `recData` ()  
*recData* 监听是否接收客户端发来的数据
- void `disconnect` ()  
*disconnect* 断开客户端的连接

## Private 属性

- int `id`
- QDataStream `in`
- QTcpSocket \* `socket`
- class `instruction` \* `ins`

### 6.8.1 详细描述

自定义线程

在文件 `thread.h` 第 11 行定义.

### 6.8.2 构造及析构函数说明

#### 6.8.2.1 `thread()`

```
thread::thread (
    QTcpSocket * newSocket,
    int id )
```

在文件 `thread.cpp` 第 3 行定义.

### 6.8.3 成员函数说明

#### 6.8.3.1 `delThread`

```
void thread::delThread (
    int ) [signal]
```

`delThread` 信号: 触发进程池删除进程

### 6.8.3.2 disconnect

```
void thread::disconnect ( ) [private], [slot]
```

disconnect 断开客户端的连接

在文件 thread.cpp 第 61 行定义.

### 6.8.3.3 getID()

```
int thread::getID ( )
```

getID 获取线程ID

返回

线程ID

在文件 thread.cpp 第 24 行定义.

### 6.8.3.4 iniSocket()

```
void thread::iniSocket ( )
```

iniSocket 初始化线程

在文件 thread.cpp 第 11 行定义.

### 6.8.3.5 recData

```
void thread::recData ( ) [private], [slot]
```

recData 监听是否接收客户端发来的数据

在文件 thread.cpp 第 45 行定义.

### 6.8.3.6 sendData()

```
void thread::sendData (
    QString sendMsg )
```

sendData 向客户端发送数据

参数

<code>sendMsg</code>	
----------------------	--

在文件 `thread.cpp` 第 29 行定义.

## 6.8.4 类成员变量说明

### 6.8.4.1 id

```
int thread::id [private]
```

在文件 `thread.h` 第 15 行定义.

### 6.8.4.2 in

```
QDataStream thread::in [private]
```

在文件 `thread.h` 第 16 行定义.

### 6.8.4.3 ins

```
class instruction* thread::ins [private]
```

在文件 `thread.h` 第 18 行定义.

### 6.8.4.4 socket

```
QTcpSocket* thread::socket [private]
```

在文件 `thread.h` 第 17 行定义.

该类的文档由以下文件生成:

- [D:/Document/VScode/C++/eCommerce/server/thread.h](#)
- [D:/Document/VScode/C++/eCommerce/server/thread.cpp](#)



## Chapter 7

# 文件说明

### 7.1 D:/Document/VScode/C++/eCommerce/server/instruction.cpp 文件参考

`#include "instruction.h"`  
instruction.cpp 的引用(Include)关系图:

### 7.2 D:/Document/VScode/C++/eCommerce/server/instruction.h 文件参考

```
#include <QStringList>
#include "lib/global.h"
#include "lib/jsonexe.h"
#include "lib/sqldiscount.h"
#include "lib/sqlquery.h"
#include "lib/sqlcart.h"
```

instruction.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

#### 类

- class [instruction](#)  
序列化操作指令类

### 7.3 D:/Document/VScode/C++/eCommerce/server/lib/global.h 文件参考

此图展示该文件直接或间接的被哪些文件引用了:

## 宏定义

- `#define JsonExe 0`
- `#define SqlQuery 1`
- `#define SqlDiscount 2`
- `#define SqlCart 3`
- `#define GetJson 0`
- `#define JudgeJson 1`
- `#define EditJson 2`
- `#define AppendJson 3`
- `#define Selectdb 0`
- `#define Insertdb 1`
- `#define Updatedb 2`
- `#define Deletedb 3`
- `#define SelectStoredb 4`
- `#define Searchdb 5`
- `#define UpdateResetDb 6`
- `#define OutPut 7`

### 7.3.1 宏定义说明

#### 7.3.1.1 AppendJson

```
#define AppendJson 3
```

在文件 `global.h` 第 14 行定义.

#### 7.3.1.2 Deletedb

```
#define Deletedb 3
```

在文件 `global.h` 第 20 行定义.

#### 7.3.1.3 EditJson

```
#define EditJson 2
```

在文件 `global.h` 第 13 行定义.

#### 7.3.1.4 GetJson

```
#define GetJson 0
```

在文件 global.h 第 11 行定义.

#### 7.3.1.5 Insertdb

```
#define Insertdb 1
```

在文件 global.h 第 18 行定义.

#### 7.3.1.6 JsonExe

```
#define JsonExe 0
```

在文件 global.h 第 5 行定义.

#### 7.3.1.7 JudgeJson

```
#define JudgeJson 1
```

在文件 global.h 第 12 行定义.

#### 7.3.1.8 OutPut

```
#define OutPut 7
```

在文件 global.h 第 26 行定义.

#### 7.3.1.9 Searchdb

```
#define Searchdb 5
```

在文件 global.h 第 24 行定义.

#### 7.3.1.10 Selectdb

```
#define Selectdb 0
```

在文件 global.h 第 17 行定义.

#### 7.3.1.11 SelectStoredb

```
#define SelectStoredb 4
```

在文件 global.h 第 23 行定义.

#### 7.3.1.12 SqlCart

```
#define SqlCart 3
```

在文件 global.h 第 8 行定义.

#### 7.3.1.13 SqlDiscount

```
#define SqlDiscount 2
```

在文件 global.h 第 7 行定义.

#### 7.3.1.14 SqlQuery

```
#define SqlQuery 1
```

在文件 global.h 第 6 行定义.

#### 7.3.1.15 Updatedb

```
#define Updatedb 2
```

在文件 global.h 第 19 行定义.

### 7.3.1.16 UpdateResetDb

```
#define UpdateResetDb 6
```

在文件 global.h 第 25 行定义.

## 7.4 D:/Document/VScode/C++/eCommerce/server/lib/jsonexe.cpp 文件参考

```
#include "jsonexe.h"
#include <QDebug>
```

jsonexe.cpp 的引用(Include)关系图:

## 7.5 D:/Document/VScode/C++/eCommerce/server/lib/jsonexe.h 文件参考

```
#include <QString>
#include <QFile>
#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonArray>
#include <QDir>
```

jsonexe.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

### 类

- class `jsonexe`  
JSON文件的操作类

## 7.6 D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.cpp 文件参考

```
#include "sqlcart.h"
```

sqlcart.cpp 的引用(Include)关系图:

## 7.7 D:/Document/VScode/C++/eCommerce/server/lib/sqlcart.h 文件参考

```
#include <QJsonDocument>
#include <QJsonArray>
#include <QJsonObject>
#include <QSqlDatabase>
#include <QSqlQuery>
#include <QSql>
#include <QSqlError>
#include <QString>
#include <QDebug>
#include <QDir>
```

sqlcart.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

## 类

- class [sqlCart](#)  
购物车数据库类

## 7.8 D:/Document/VScode/C++/eCommerce/server/lib/sqldiscount.cpp 文件参考

```
#include "sqldiscount.h"
```

sqldiscount.cpp 的引用(Include)关系图:

## 7.9 D:/Document/VScode/C++/eCommerce/server/lib/sqldiscount.h 文件参考

```
#include <QSqlDatabase>
#include <QSqlQuery>
#include <QSql>
#include <QSqlError>
#include <QString>
#include <QDebug>
#include <QDir>
```

sqldiscount.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

## 类

- class [sqlDiscount](#)  
折扣数据库类

## 7.10 D:/Document/VScode/C++/eCommerce/server/lib/sqlquery.cpp 文件参考

```
#include "sqlquery.h"
```

sqlquery.cpp 的引用(Include)关系图:

## 7.11 D:/Document/VScode/C++/eCommerce/server/lib/sqlquery.h 文件参考

```
#include <QJsonDocument>
#include <QJsonArray>
#include <QJsonObject>
#include <QSqlDatabase>
#include <QSqlQuery>
#include <QSql>
#include <QSqlError>
#include <QString>
#include <QDebug>
#include <QDir>
```

sqlquery.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

## 类

- class `sqlQuery`  
商品数据库类

## 7.12 D:/Document/VScode/C++/eCommerce/server/main.cpp 文件参考

```
#include "server.h"
#include <QApplication>
main.cpp 的引用(Include)关系图:
```

## 函数

- int `main` (int argc, char \*argv[])

### 7.12.1 函数说明

#### 7.12.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

在文件 main.cpp 第 5 行定义.

## 7.13 D:/Document/VScode/C++/eCommerce/server/processpool.cpp 文件参考

```
#include "processpool.h"
processpool.cpp 的引用(Include)关系图:
```

## 7.14 D:/Document/VScode/C++/eCommerce/server/processpool.h 文件参考

```
#include <QVector>
#include <QtAlgorithms>
#include "thread.h"
processpool.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:
```

## 类

- class `processPool`  
进程池

## 7.15 D:/Document/VScode/C++/eCommerce/server/server.cpp 文件参考

```
#include "server.h"
#include "ui_server.h"
#include <QMessageBox>
#include <QString>
server.cpp 的引用(Include)关系图:
```

## 7.16 D:/Document/VScode/C++/eCommerce/server/server.h 文件参考

```
#include <QMainWindow>
#include <QTcpServer>
#include "processpool.h"
#include "lib/sqlquery.h"
server.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:
```

## 类

- class `MainWindow`  
服务端类

## 命名空间

- `Ui`

## 7.17 D:/Document/VScode/C++/eCommerce/server/thread.cpp 文件参考

```
#include "thread.h"
thread.cpp 的引用(Include)关系图:
```

## 7.18 D:/Document/VScode/C++/eCommerce/server/thread.h 文件参考

```
#include <QTcpSocket>
#include <QDataStream>
#include "instruction.h"
thread.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:
```

## 类

- class `thread`  
自定义线程