

QtDocumentCN

2021 年 2 月 13 日

目录

第一章 QSql	3
第二章 QSqlDatabase	5

第一章 QSql

QSql 命名空间

QSql 命名空间里的各种各样的标识符, 已经被运用在 Qt SQL 各个模块中。[更多](#)

属性	方法
头文件	#include <QSql>
qmake	QT += sql

类型

enum	Location { BeforeFirstRow, AfterLastRow }
enum	NumericalPrecisionPolicy { LowPrecisionInt32, LowPrecisionInt64, LowPrecisionInt128 }
flags	ParamType
enum	ParamTypeFlag { In, Out, InOut, Binary }
enum	TableType { Tables, SystemTables, Views, AllTables }

细节的介绍

[查看 Qt SQL](#)

类型文档

enum QSql::Location

此枚举类型描述特殊的 sql 导航位置

常量	值	介绍
QSql::BeforeFirstRow	-1	在第一个记录之前
QSql::AfterLastRow	-2	在最后一个记录之后

另请参阅 `QSqlQuery::at()`

```
enum QSql::NumericalPrecisionPolicy
```

数据库中的数值可以比它们对应的 C++ 类型更精确。此枚举列出在应用程序中表示此类值的策略。

常量	值	介绍
<code>QSql::LowPrecisionInt32</code>	<code>0x01</code>	对于 32 位的整形数值。在浮点数的情况下，小数部分将会被舍去
<code>QSql::LowPrecisionInt64</code>	<code>0x02</code>	对于 64 位的整形数值。在浮点数的情况下，小数部分将会被舍去
<code>QSql::LowPrecisionDouble</code>	<code>0x04</code>	强制双精度值。这个默认的规则
<code>QSql::HighPrecision</code>	<code>0</code>	字符串将会维持精度

注意：如果特定的驱动发生溢出，这是一个真实行为。像 Oracle 数据库在这种情形下，就会返回一个错误。

```
enum QSql::ParamTypeFlag
```

```
flags QSql::ParamType
```

这个枚举用于指定绑定参数的类型

常量	值	介绍
<code>QSql::In</code>	<code>0x00000001</code>	这个参数被用于向数据库里写入数据
<code>QSql::Out</code>	<code>0x00000002</code>	这个参数被用于向数据库里获得数据
<code>QSql::InOut</code>	<code>In Out</code>	这个参数被用于向数据库里写入数据；使用查询来向数据库里，重写数据
<code>QSql::Binary</code>	<code>0x00000004</code>	如果您想显示数据为原始的二进制数据，那么必须是 OR'd 和其他的标志

类型参数类型定义为 `QFlags`。它被存放在一个 OR 与类型参数标志的值的组合。

第二章 QSqlDatabase

QSqlDatabase 类用于处理数据库的连接

属性	方法
头文件	<code>#include <QSql></code>
qmake	<code>QT += sql</code>

类型

enum	Location { BeforeFirstRow, AfterLastRow }
enum	NumericalPrecisionPolicy { LowPrecisionInt32, LowPrecisionInt64, LowPrecisionInt128 }
flags	ParamType
enum	ParamTypeFlag { In, Out, InOut, Binary }
enum	TableType { Tables, SystemTables, Views, AllTables }

细节的介绍

查看 [Qt SQL](#)

类型文档

enum QSql::Location

此枚举类型描述特殊的 sql 导航位置

常量	值	介绍
<code>QSql::BeforeFirstRow</code>	-1	在第一个记录之前
<code>QSql::AfterLastRow</code>	-2	在最后一个记录之后

另请参阅 [QSqlQuery::at\(\)](#)

enum QSql::NumericalPrecisionPolicy

数据库中的数值可以比它们对应的 C++ 类型更精确。此枚举列出在应用程序中表示此类值的策略。

常量	值	介绍
QSql::LowPrecisionInt32	0x01	对于 32 位的整形数值。在浮点数的情况下，小数部分将会被舍去
QSql::LowPrecisionInt64	0x02	对于 64 位的整形数值。在浮点数的情况下，小数部分将会被舍去
QSql::LowPrecisionDouble	0x04	强制双精度值。这个默认的规则
QSql::HighPrecision	0	字符串将会维持精度

注意：如果特定的驱动发生溢出，这是一个真实行为。像 Oracle 数据库在这种情形下，就会返回一个错误。

```
enum QSql::ParamTypeFlag
```

```
flags QSql::ParamType
```

这个枚举用于指定绑定参数的类型

常量	值	介绍
QSql::In	0x00000001	这个参数被用于向数据库里写入数据
QSql::Out	0x00000002	这个参数被用于向数据库里获得数据
QSql::InOut	In Out	这个参数被用于向数据库里写入数据；使用查询来向数据库里，重写数据
QSql::Binary	0x00000004	如果您想显示数据为原始的二进制数据，那么必须是 OR'd 和其他的标志

类型参数类型定义为 `QFlags`。它被存放在一个 OR 与类型参数标志的值的组合。