50 QStringListModel

在这个案例中,借助QStringListModel类的对象,表示计算机内存中的字符串表,将其关联于QListView 视图。只要模型中的数据发生变化,视图就会立即同步所发生的改变。用户在视图中对数据所做的修改,也会自动反映到模型中。视图与模型在任何时候,都会自动保持一致。

50.1 创建项目

通过QtCreator,在C:\Users\Minwei\Projects\Qt路径下,创建名为StringList的项目。

50.2 设计界面



对象查看	器 - Qt Designer		x
对象		类	^
✓ 333 St	tringListDialog	QDialog	
~ =	m_layoutLeft	QVBoxLayout	
	m_btnAppend	QPushButton	
	m_btnClear	QPushButton	
	m_btnDelete	QPushButton	
	m_btnInsert	QPushButton	
	m_btnReset	QPushButton	
	m_spacerLeftBottom	Spacer	
	m_spacerLeftTop	Spacer	
× 0] m_layoutHor	QHBoxLayout	
~	📃 m_layoutRight	QVBoxLayout	
	m_btnEdit2List	QPushButton	
	m_btnList2Edit	QPushButton	
	m_spacerRightBottom	Spacer	
	m_spacerRightTop	Spacer	
	m_edit	QPlainTextEdit	
	m_list	QListView	
m	label	QLabel	\lor

 $\label{eq:c:Users} C: \label{eq:c:Users} C$

1	xml version="1.0" encoding="UTF-8"?		
2	<ui version="4.0"></ui>		
3	<class>StringListDialog</class>		
4	<pre><widget class="QDialog" name="StringListDialog"></widget></pre>		
5	<property name="geometry"></property>		
6	<rect></rect>		
7	<x>0</x>		
8	<y>0</y>		
9	<pre><width>700</width></pre>		
10	<height>400</height>		
11			
12			
13	<property name="windowTitle"></property>		
14	<pre><string>字符串表</string></pre>		
15			
16	<layout class="QGridLayout" name="m_layoutGrid"></layout>		
17	<item column="0" row="0"></item>		
18	<layout class="QVBoxLayout" name="m_layoutLeft"></layout>		
19	<item></item>		
20	<spacer name="m_spacerLeftTop"/>		
21	<property name="orientation"></property>		
22	<enum>Qt::Vertical</enum>		
23			
24	<property name="sizeHint" stdset="0"></property>		
25	<size></size>		
26	<width>20</width>		
27	<height>40</height>		
28			
29			
30			
31			
32	<item></item>		
33	<widget class="QPushButton" name="m_btnReset"></widget>		
34	<property name="text"></property>		
35	<string>重置</string>		
36			
37			
38			
39	<item></item>		
40	<pre><widget class="QPushButton" name="m_btnAppend"></widget></pre>		
41	<property name="text"></property>		
42	<pre><string>添加</string></pre>		
43			
44			
45	1tem		
46	<1tem>		
47	<pre><widget class="QPushButton" name="m_btnInsert"></widget></pre>		
48	<property name="text"></property>		
49	<string>抽入</string>		
50			
51			
52	I Lem		
53	<1tem>		

54	<pre><widget class="QPushButton" name="m_btnDelete"></widget></pre>
55	<property name="text"></property>
56	<string>删除</string>
57	
58	
59	
60	<item></item>
61	<pre><widget class="OPushButton" name="m btnClear"></widget></pre>
62	<property name="text"></property>
63	<pre>string> imaxima></pre>
64	
65	
66	//itam>
67	zitem
68	<pre><recompany="m_spacer! aftpottom"=""></recompany="m_spacer!></pre>
60	<pre>cspace: name="m_space:Lercbottom"> cspace: name="m_space:Lercbottom"> cspace:Lercbottom: name="m_space:Lercbo</pre>
70	<pre><pre>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre></pre>
70	<enum>qt::vertical</enum>
71	
72	<property name="sizeHint" stdset="0"></property>
73	<size></size>
74	<wrdth>20</wrdth>
75	<height>40</height>
76	
77	
78	
79	
80	
81	
82	<item column="1" row="0"></item>
83	<layout class="QHBoxLayout" name="m_layoutHor"></layout>
84	<item></item>
85	<widget class="QListView" name="m_list"></widget>
86	<property name="frameShape"></property>
87	<enum>QFrame::WinPanel</enum>
88	
89	
90	
91	<item></item>
92	<layout class="QVBoxLayout" name="m_layoutRight"></layout>
93	<item></item>
94	<spacer name="m_spacerRightTop"/>
95	<property name="orientation"></property>
96	<pre><enum>Ot::Vertical</enum></pre>
97	
98	<property_name="sizehint" stdset="0"></property_name="sizehint">
99	<size></size>
100	<pre><width>20</width></pre>
101	<height>40</height>
102	
102	
104	
105	/itam>
105	
107	<pre>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>
100	<pre><wruget class="QPUSHBULLOH" name="m_DTHL1ST2E01T"> </wruget></pre>
100	<property name="text"></property>
T03	<string>>></string>

110	
111	
112	
113	<item></item>
114	<widget class="QPushButton" name="m_btnEdit2List"></widget>
115	<property name="text"></property>
116	<pre><string><<</string></pre>
117	
118	
119	
120	<item></item>
121	<pre><spacer name="m spacerRightBottom"/></pre>
122	<property name="orientation"></property>
123	<pre><enum>Ot::Vertical</enum></pre>
124	
125	<pre><pre>chronerty_name="sizeHint" stdset="0"</pre></pre>
126	
127	<pre><width>20//width></width></pre>
128	<pre>chaight</pre>
120	
130	
121	
122	
122	
137	
135	
126	<pre>duidant class="OPlainTextEdit" name="m edit"></pre>
137	<pre></pre>
138	<pre><pre>conum>OErame::WinPanel</pre></pre>
130	
140	
141	
142	
143	
144	<pre><item column="1" row="1"></item></pre>
145	<pre><widget class="0.abel" name="m label"></widget></pre>
146	<pre>chroperty_name="nalette"></pre>
147	
148	<active></active>
149	<pre><colorrole role="WindowText"></colorrole></pre>
150	<pre><brush brushstyle="SolidPattern"></brush></pre>
151	<color alpha="255"></color>
152	<red>255</red>
153	<areen>0</areen>
154	<pre><hlueson< pre="">/hlueson</hlueson<></pre>
155	
156	
157	
158	
159	<inactive></inactive>
160	<pre><colorrole_role="windowtext"></colorrole_role="windowtext"></pre>
161	<pre></pre>
162	<color alpha="255"></color>
163	<red>255</red>
164	<areen>0</areen>
165	<pre> <br <="" td=""/></pre>
-05	

100	
T00	
167	
168	
169	
170	<disabled></disabled>
171	<colorrole role="WindowText"></colorrole>
172	<brush brushstyle="SolidPattern"></brush>
173	<color alpha="255"></color>
174	<red>120</red>
175	<pre><green>120</green></pre>
176	<blue>120</blue>
177	
178	
179	
180	
181	
182	
183	
184	
185	
186	
187	<resources></resources>
188	<connections></connections>
189	

50.3 实现功能

 $\label{eq:c:UsersMinweilProjectsQtStringListStringlistdialog.h: \\$

```
1 #ifndef STRINGLISTDIALOG_H
 2
    #define STRINGLISTDIALOG_H
 3
   #include <QDialog>
 4
    #include <QStringListModel>
 5
 6
 7
    QT_BEGIN_NAMESPACE
    namespace Ui { class StringListDialog; }
 8
9
    QT_END_NAMESPACE
10
11
    class StringListDialog : public QDialog
12
    {
13
        Q_OBJECT
14
15
    public:
16
        StringListDialog(QWidget *parent = nullptr);
        ~StringListDialog();
17
18
19
    private slots:
20
        void on_m_btnReset_clicked();
        void on_m_btnAppend_clicked();
21
        void on_m_btnInsert_clicked();
22
        void on_m_btnDelete_clicked();
23
        void on_m_btnClear_clicked();
24
25
26
        void on_m_btnList2Edit_clicked();
```

```
27
        void on_m_btnEdit2List_clicked();
28
29
        void on_m_list_clicked(const QModelIndex &index);
30
31
    private:
32
        Ui::StringListDialog *ui;
        QStringListModel* m_model;
33
34
   };
35
    #endif // STRINGLISTDIALOG_H
36
```

 $\label{eq:c:UsersMinweilProjectsQtStringListStringlistdialog.cpp: \end{tabular}$

```
1
    #include <QTextBlock>
2
    #include "stringlistdialog.h"
 3
4
    #include "ui_stringlistdialog.h"
 5
    StringListDialog::StringListDialog(QWidget *parent)
6
 7
        : QDialog(parent)
8
        , ui(new Ui::StringListDialog)
        , m_model(new QStringListModel(this))
9
10
    {
11
        ui->setupUi(this);
12
13
        ui->m_list->setModel(m_model);
14
        on_m_btnReset_clicked();
15
    }
16
17
    StringListDialog::~StringListDialog()
18
    {
        delete ui:
19
    }
20
21
    void StringListDialog::on_m_btnReset_clicked()
22
23
    {
24
        QStringList strings;
        strings << "北京" << "上海" << "广州" << "深圳";
25
        m_model->setStringList(strings);
26
27
    }
28
    void StringListDialog::on_m_btnAppend_clicked()
29
30
    {
        m_model->insertRow(m_model->rowCount());
31
32
        QModelIndex index = m_model->index(m_model->rowCount() - 1, 0);
        m_model->setData(index, "添加数据", Qt::DisplayRole);
33
        ui->m_list->setCurrentIndex(index);
34
35
    }
36
    void StringListDialog::on_m_btnInsert_clicked()
37
38
    {
39
        QModelIndex index = ui->m_list->currentIndex();
40
        m_model->insertRow(index.row());
        m_model->setData(index, "插入数据", Qt::DisplayRole);
41
        ui->m_list->setCurrentIndex(index);
42
```

```
43
    }
44
45
    void StringListDialog::on_m_btnDelete_clicked()
46
    {
        m_model->removeRow(ui->m_list->currentIndex().row());
47
48
    }
49
    void StringListDialog::on_m_btnClear_clicked()
50
51
    {
52
        m_model->removeRows(0, m_model->rowCount());
53
    }
54
55
    void StringListDialog::on_m_btnList2Edit_clicked()
56
    {
        ui->m_edit->clear();
57
58
59
        for (QString const& string : m_model->stringList())
            ui->m_edit->appendPlainText(string);
60
61
    }
62
63
    void StringListDialog::on_m_btnEdit2List_clicked()
64
    {
65
        on_m_btnClear_clicked();
66
67
        QStringList strings;
68
        QTextDocument* doc = ui->m_edit->document();
69
70
        int blockCount = doc->blockCount();
71
        for(int blockNumber = 0; blockNumber < blockCount; ++blockNumber)</pre>
72
73
            strings << doc->findBlockByNumber(blockNumber).text();
74
75
        m_model->setStringList(strings);
76
    }
77
    void StringListDialog::on_m_list_clicked(const QModelIndex &index)
78
79
    {
        ui->m_label->setText(QString("第%1行第%2列的项被点击:%3")
80
            .arg(index.row()).arg(index.column()).arg(index.data().toString()));
81
82
    }
```

50.4 测试验证

运行效果如图所示:

■ 字符串表				?	×
重置 添加 插入 删除 清空	北京 插入数据 上海 插入数据 广州 插入数据 深圳 插入数据 武汉 添加数据 添加数据 添加数据 添加数据	>>> <<<	北京 插入数据 上海 插入数据 定州 插入数据 深圳 插入数据 武汉 添加数据 添加数据 添加数据 添加数据		
	第7行第0列的项被点击				

50.5 案例对比

将这个案例与先前的List案例做一个对比。在MV结构中,QStringListModel模型和QListView视图是分离 且同步的。将数据加入QStringListModel,QListView即可显示,通过QListView修改数据, QStringListModel也会因之而变。List案例里的QListWidget并没有模型与之关联,数据都是以 QListWidgetItem对象的形式直接存放在QListWidget内部的。数据和显示是一体的。这就是视图组件和 项集组件的主要区别。