<http://blog.sina.com.cn/s/blog_a6fb6cc90102uw3d.html>

博客好久没更新了，连续加班几个月，一直没时间写，现在项目终于发版了，先来写篇博客庆祝一下，也希望能有时间好好休息一下。。。

    今天介绍一下一个小东西 — 如何让QComboBox实现复选功能？

需求：

1. 下拉列表有复选功能
2. 不可编辑
3. 显示所有选中项

关于QComboBox的复选功能有几种方案：

1. QStandardItemModel + QStandardItem
2. QListWidget + QListWidgetItem
3. Model/View + QItemDelegate

当然，还有其它更好的方式，这里就不再过多介绍了，下面介绍一种比较简单的：

QListWidget + QListWidgetItem + QCheckBox

    pListWidget = new QListWidget(this); pLineEdit = new QLineEdit(this); for (int i = 0; i < 5; ++i) { QListWidgetItem \*pItem = new QListWidgetItem(pListWidget); pListWidget->addItem(pItem); pItem->setData(Qt::UserRole, i); QCheckBox \*pCheckBox = new QCheckBox(this); pCheckBox->setText(QStringLiteral("Qter%1").arg(i)); pListWidget->addItem(pItem); pListWidget->setItemWidget(pItem, pCheckBox); connect(pCheckBox, SIGNAL(stateChanged(int)), this, SLOT(stateChanged(int))); } ui.comboBox->setModel(pListWidget->model()); ui.comboBox->setView(pListWidget); ui.comboBox->setLineEdit(pLineEdit); pLineEdit->setReadOnly(true); //ui.comboBox->setEditable(true); connect(pLineEdit, SIGNAL(textChanged(const QString &)), this, SLOT(textChanged(const QString &)));

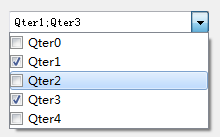
void MultiComboBox::stateChanged(int state) { bSelected = true; QString strSelectedData(""); strSelectedText.clear(); QObject \*object = QObject::sender(); QCheckBox \*pSenderCheckBox = static\_cast(object); int nCount = pListWidget->count(); for (int i = 0; i < nCount; ++i) { QListWidgetItem \*pItem = pListWidget->item(i); QWidget \*pWidget = pListWidget->itemWidget(pItem); QCheckBox \*pCheckBox = (QCheckBox \*)pWidget; if (pCheckBox->isChecked()) { QString strText = pCheckBox->text(); strSelectedData.append(strText).append(";"); } //所点击的复选框 if (pSenderCheckBox == pCheckBox) { int nData = pItem->data(Qt::UserRole).toInt(); qDebug() << QString("I am sender...id : %1").arg(nData); } } if (strSelectedData.endsWith(";")) strSelectedData.remove(strSelectedData.count() - 1, 1); if (!strSelectedData.isEmpty()) { //ui.comboBox->setEditText(strSelectedData); strSelectedText = strSelectedData; pLineEdit->setText(strSelectedData); pLineEdit->setToolTip(strSelectedData); } else { pLineEdit->clear(); //ui.comboBox->setEditText(""); } bSelected = false; }

void MultiComboBox::textChanged(const QString &text) { if (!bSelected) pLineEdit->setText(strSelectedText); }

  当点击的复选框状态改变时候，会发送stateChanged信号，槽中通过sender来获取信号的发送者，然后可以获取所需要的数据（比如：可以通过setData保存一些自定义的数据），遍历所有的复选框，获取选中项的数据进行显示。

  中间遇到一个小问题，当点击空白处，下来列表会进行收回，此时QLineEdit的数据将会被自动清空，所以此处判断QLineEdit的textChanged来进行恢复。

效果：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=a6fb6cc90102uw3d&url=http://album.sina.com.cn/pic/0033AMT7zy6K6qAuAYzcb)

**注：**

    技术在于交流、沟通，转载请注明出处并保持作品的完整性。

    作者：[╰☆奋斗ing❤孩子`](http://blog.sina.com.cn/liang19890820)  原文：<http://blog.sina.com.cn/s/blog_a6fb6cc90102uw3d.html>。