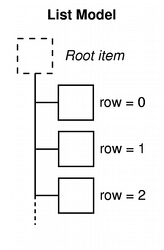
## QTreeWidget用法

1、简介

QTreeWidget的初始化是使用一个QTreeWidgetItem 的list填充到QTreeWidget中。

QTreeWidget的setColumnCount ( int )方法可以设置QTreeWidget显示的列数。



2、代码

int CEnergyquery::InitLeftTree()

{

//初始化左边作树形结构

QList<QTreeWidgetItem \*> TreeWigItms; //使用一个QTreeWidgetItem 的list填充到//QTreeWidget中

int ID = 0;

QString Name;

// QTreeWidgetItem的list中插入N个元素

for(int m=0;m<N;m++)

{

QTreeWidgetItem \*PareNodeItm = new QTreeWidgetItem;

PareNodeItm->setData(0,Qt::UserRole,NodeID); //存储节点相关数据

PareNodeItm->setText(0, Name); //设置节点的名称

TreeWigItms.push\_back(PareNodeItm);

//为 list中插入的元素添加M个子节点

for(int k=0;k<M;k++)

{

QTreeWidgetItem \*ChidNodeItm = new QTreeWidgetItem(PareNodeItm);

ChidNodeItm->setData(0, Qt::UserRole,ID); //节点相关数据

ChidNodeItm->setText(0, Name); //节点名称

}

}

//在QTreeWidget上显示树形结构

ui->treeWidget->insertTopLevelItems(0,TreeWigItms);

return 1;

}

3、常用函数：

1）void QTreeWidget::insertTopLevelItems ( int index, const QList<QTreeWidgetItem \*> & items )

从QTreeWidget的第Index元素处，开始插入列表QList<QTreeWidgetItem \*> & items

2）QList<QTreeWidgetItem \*> QTreeWidget::findItems ( const QString & text, Qt::MatchFlags flags, int column = 0 ) const

在QTreeWidget的Item节点中检索包含text字符串的节点，并返回这些节点的列表。

有用初始化QTreeWidget的时候使用的是QList<QTreeWidgetItem \*>列表，检索时只从QList<QTreeWidgetItem \*>列表的元素中检索，检索元素下的子节点。需要自己实现代码检索QList<QTreeWidgetItem \*>列表元素以及元素下的子节点。

for(int m=0;m< RowNum; m++) //遍历所有父节点

{

CurTopLevItm = ui->treeWidget->topLevelItem(m); //显示与关键字匹配的馈线

TopNodText=CurTopLevItm->text(0);

IsContain=TopNodText.contains(KeyWord,Qt::CaseInsensitive); //节点名字是否包含关键字

if(IsContain)

{

if(CurTopLevItm->isHidden())

CurTopLevItm->setHidden(false); //显示此节点

}

else

CurTopLevItm->setHidden(true);

//遍历该父节点下的子节点

for(int n =0;n<CurTopLevItm->childCount();n++)

{

CurChidItm = CurTopLevItm->child(n);

ChilNodText=CurChidItm->text(0);

IsContain = ChilNodText.contains(KeyWord,Qt::CaseInsensitive); //节点名字是否包含关键字

if(!IsContain) //与关键字不匹配

{

CurChidItm->setHidden(true);

continue;

}

CurChidItm->setHidden(false); //与关键字匹配

if(CurTopLevItm->isHidden()) //显示其父节点

CurTopLevItm->setHidden(false);

if(!CurTopLevItm->isExpanded())

CurTopLevItm->setExpanded(true); //展开父节点

}

}