

QTableView Class

QTableView 类提供表视图的默认模型/视图实现。[更多的...](#)

Header:	#include < QTableView>
CMake:	find_package(Qt6 REQUIRED COMPONENTS Widgets) target_link_libraries(mytarget PRIVATE Qt6::Widgets)
qmake:	QT += widgets
Inherits:	QAbstractItemView
Inherited By:	QTableWidget

- [所有成员的列表，包括继承的成员](#)

特性

cornerButtonEnabled : bool	gridStyle : Qt::PenStyle	showGrid : bool	sortingEnabled : bool	wordWrap : bool
--	--	---------------------------------	---------------------------------------	---------------------------------

公共职能

	QTableView (QWidget *parent = nullptr)
virtual	~QTableView ()
void	clearSpans ()
int	columnAt (int x) const
int	columnSpan (int row, int column) const
int	columnViewportPosition (int column) const
int	columnWidth (int column) const
Qt::PenStyle	gridStyle () const
QHeaderView *	horizontalHeader () const
bool	isColumnHidden (int column) const

bool	QTableView(QWidget *parent = nullptr) isCornerButtonEnabled() const
bool	isRowHidden (int row) const
bool	isSortingEnabled () const
int	rowAt (int y) const
int	rowHeight (int row) const
int	rowSpan (int row, int column) const
int	rowViewportPosition (int row) const
void	setColumnHidden (int column, bool hide)
void	setColumnWidth (int column, int width)
void	setCornerButtonEnabled (bool enable)
void	setGridStyle (Qt::PenStyle style)
void	setHorizontalHeader (QHeaderView *header)
void	setRowHeight (int row, int height)
void	setRowHidden (int row, bool hide)
void	setSortingEnabled (bool enable)
void	setSpan (int row, int column, int rowSpanCount, int columnSpanCount)
void	setVerticalHeader (QHeaderView *header)
void	setWordWrap (bool on)
bool	showGrid () const
QHeaderView *	verticalHeader () const
bool	wordWrap () const

重新实施公共职能

virtual QModelIndex	indexAt (const QPoint &pos) const override
virtual void	scrollTo (const QModelIndex &index, QAbstractItemView::ScrollHint hint = EnsureVisible) override
virtual void	setModel (QAbstractItemModel *model) override
virtual void	setRootIndex (const QModelIndex &index) override

virtual QModelIndex	indexAt (const QPoint & <i>pos</i>) const override
virtual void	setSelectionModel (QItemSelectionModel * <i>selectionModel</i>) override
virtual QRect	visualRect (const QModelIndex & <i>index</i>) const override

公共槽

void	hideColumn (int <i>column</i>)
void	hideRow (int <i>row</i>)
void	resizeColumnToContents (int <i>column</i>)
void	resizeColumnsToContents ()
void	resizeRowToContents (int <i>row</i>)
void	resizeRowsToContents ()
void	selectColumn (int <i>column</i>)
void	selectRow (int <i>row</i>)
void	setShowGrid (bool <i>show</i>)
void	showColumn (int <i>column</i>)
void	showRow (int <i>row</i>)
void	sortByColumn (int <i>column</i> , Qt::SortOrder <i>order</i>)

重载的protected function

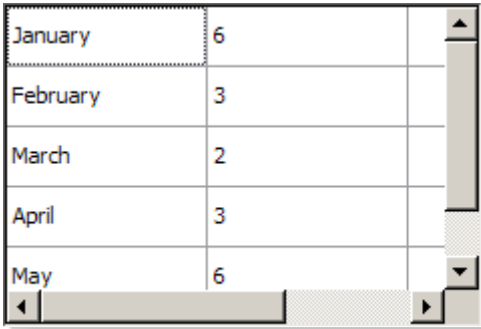
virtual void	currentChanged (const QModelIndex & <i>current</i> , const QModelIndex & <i>previous</i>) override
virtual int	horizontalOffset () const override
virtual void	initViewItemOption (QStyleOptionViewItem * <i>option</i>) const override
virtual bool	isIndexHidden (const QModelIndex & <i>index</i>) const override
virtual QModelIndex	moveCursor (QAbstractItemView::CursorAction <i>cursorAction</i> , Qt::KeyboardModifiers <i>modifiers</i>) override
virtual void	paintEvent (QPaintEvent * <i>event</i>) override
virtual void	scrollContentsBy (int <i>dx</i> , int <i>dy</i>) override
virtual QModelIndexList	selectedIndexes () const override

virtual void	currentChanged (const QModelIndex & <i>current</i> , const QModelIndex & <i>previous</i>) override
virtual void	selectionChanged (const QItemSelection & <i>selected</i> , const QItemSelection & <i>deselected</i>) override
virtual void	setSelection (const QRect & <i>rect</i> , QItemSelectionModel::SelectionFlags <i>flags</i>) override
virtual int	sizeHintForColumn (int <i>column</i>) const override
virtual int	sizeHintForRow (int <i>row</i>) const override
virtual void	timerEvent (QTimerEvent * <i>event</i>) override
virtual void	updateGeometries () override
virtual int	verticalOffset () const override
virtual QSize	viewportSizeHint () const override
virtual QRegion	visualRegionForSelection (const QItemSelection & <i>selection</i>) const override

继承的槽

void	columnCountChanged (int <i>oldCount</i> , int <i>newCount</i>)
void	columnMoved (int <i>column</i> , int <i>oldIndex</i> , int <i>newIndex</i>)
void	columnResized (int <i>column</i> , int <i>oldWidth</i> , int <i>newWidth</i>)
void	rowCountChanged (int <i>oldCount</i> , int <i>newCount</i>)
void	rowMoved (int <i>row</i> , int <i>oldIndex</i> , int <i>newIndex</i>)
void	rowResized (int <i>row</i> , int <i>oldHeight</i> , int <i>newHeight</i>)

详细说明



QTableView 实现了一个显示模型中项目的表视图。该类用于提供以前由 QTable 类提供的标准表，但使用 Qt 模型/视图架构提供的更灵活的方法。

QTableView 类是其中之一[Model/View Classes](#)是 Qt 的一部分[model/view framework](#)。

QTableView 实现了定义的接口 [QAbstractItemView](#) 类，允许它显示由派生的模型提供的数据 [QAbstractItemModel](#) 班级。

导航


您可以通过用鼠标单击单元格或使用箭头键来导航表中的单元格。因为 QTableView 启用 [tabKeyNavigation](#) 默认情况下，您还可以点击 Tab 和 Backtab 在单元格之间移动。

视觉外观

该表有一个垂直标题，可以使用以下命令获得 [verticalHeader\(\)](#) 函数，以及可通过以下方式获得的水平标题 [horizontalHeader\(\)](#) 功能。表中每行的高度可以通过使用找到 [rowHeight\(\)](#)；类似地，可以使用以下方法找到列的宽度 [columnWidth\(\)](#)。由于这两个都是普通的小部件，因此您可以使用它们来隐藏它们中的任何一个 [hide\(\)](#) 功能。每个标头都配置有其 [highlightSections](#) 和 [sectionsClickable](#) 属性设置为 true。

行和列可以隐藏和显示 [hideRow\(\)](#), [hideColumn\(\)](#), [showRow\(\)](#)，和 [showColumn\(\)](#)。可以选择它们 [selectRow\(\)](#) 和 [selectColumn\(\)](#)。该表将显示一个网格，具体取决于 [showGrid](#) 财产。

表视图中显示的项目与其他项目视图中的项目一样，都是使用标准进行渲染和编辑的 [delegates](#)。但是，对于某些任务，有时能够在表中插入小部件会很有用。小部件是为特定索引设置的 [setIndexWidget\(\)](#) 函数，稍后用 [indexWidget\(\)](#)。



ID	NAME	ADDRESS	COUNTRY
1	John Doe	123 Main Street, New York	USA
2	Jane Smith	456 Elm Street, London	UK
3	Bob Johnson	789 Oak Street, Tokyo	Japan
4	Alice Brown	101 Pine Street, Sydney	Australia
5	Charlie Wilson	202 Cedar Street, Paris	France
6	Diana Prince	303 Birch Street, Berlin	Germany
7	Frank Miller	404 Spruce Street, Rome	Italy
8	Grace Lee	505 Willow Street, Seoul	South Korea
9	Henry Kim	606 Ash Street, Manila	Philippines
10	Ivy Green	707 Hickory Street, Jakarta	Indonesia

默认情况下，表中的单元格不会扩展以填充可用空间。您可以通过拉伸最后一个标题部分来使单元格填充可用空间。使用访问相关标头 [horizontalHeader\(\)](#) 或者 [verticalHeader\(\)](#) 并设置标题 [stretchLastSection](#) 财产。要根据每列或行的空间需求分配可用空间，请调用视图的 [resizeColumnsToContents\(\)](#) 或者 [resizeRowsToContents\(\)](#) 功能。

坐标系

对于某些特殊形式的表格，能够在行索引和列索引以及小部件坐标之间进行转换非常有用。这 [rowAt\(\)](#) 函数提供指定行的视图内的 y 坐标；行索引可用于获取相应的 y 坐标 [rowViewportPosition\(\)](#)。这 [columnAt\(\)](#) 和 [columnViewportPosition\(\)](#) 函数提供 x 坐标和列索引之间的等效转换操作。

也可以看看 [QTableWidget](#), [View Classes](#), [QAbstractItemModel](#), [QAbstractItemView](#), [Chart Example](#), [Pixelator Example](#)，和 [Table Model Example](#)。

财产文件

cornerButtonEnabled : bool

该属性保存左上角的按钮是否启用

如果此属性为true，则启用表视图左上角的按钮。单击此按钮将选择表视图中的所有单元格。

该属性是true默认的。

访问功能：

bool	isCornerButtonEnabled() const
void	setCornerButtonEnabled(bool enable)

gridStyle : Qt::PenStyle

该属性保存用于绘制网格的笔样式。

该属性保存绘制网格时使用的样式（请参见[showGrid](#)）。

访问功能：

Qt::PenStyle	gridStyle() const
void	setGridStyle(Qt::PenStyle style)

showGrid : bool

该属性保存是否显示网格

如果该属性是true为表格绘制网格；如果属性为false，则不绘制网格。默认值是true。

访问功能：

bool	showGrid() const
void	setShowGrid(bool show)

sortingEnabled : bool

该属性保存是否启用排序

如果此属性为true，则为表启用排序。如果此属性为false，则不启用排序。默认值为 false。

注： 将属性设置为 true时 `setSortingEnabled()` 立即触发调用 `sortByColumn()` 与当前排序部分和顺序。

访问功能：

bool	isSortingEnabled() const
void	setSortingEnabled (bool <i>enable</i>)

也可以看看[sortByColumn\(\)](#)。

wordWrap : bool

该属性保存项目文本自动换行策略

如果此属性为true，则项目文本将在必要的断字处换行；否则它根本没有被包裹。该属性是true默认的。

请注意，即使启用了换行，单元格也不会扩展以适合所有文本。省略号将根据当前插入[textElideMode](#)。

访问功能：

bool	wordWrap() const
void	setWordWrap (bool <i>on</i>)

成员函数文档

*[explicit] QTableView::QTableView([QWidget](#) *parent = nullptr)*

构造一个表视图*parent*来表示数据。

也可以看看[QAbstractItemModel](#)。

[virtual] QTableView::~~QTableView()

破坏表视图。

void QTableView::clearSpans()

删除表视图中的所有行和列跨度。

也可以看看[setSpan\(\)](#)。

int QTableView::columnAt(int x) const

返回给定 x 坐标所在的列，x，在内容坐标中位于。

注意：如果给定坐标无效（没有列），则该函数返回 -1。

也可以看看[rowAt\(\)](#)。

*[protected slot]void QTableView::columnCountChanged(int oldCount, int
newCount)*

每当添加或删除列时都会调用此槽。先前的列数由以下方式指定oldCount，新的列数由下式指定newCount。

*[protected slot]void QTableView::columnMoved(int column, int oldIndex,
int newIndex)*

调用此槽来更改给定的索引column在表视图中。旧索引由以下方式指定oldIndex，新索引为newIndex。

也可以看看[rowMoved\(\)](#)。

*[protected slot]void QTableView::columnResized(int column, int
oldWidth, int newWidth)*

调用此槽来更改给定的宽度column。旧宽度指定为oldWidth，新宽度为newWidth。

也可以看看[rowResized\(\)](#)。

int QTableView::columnSpan(int row, int column) const

返回（处的表元素的列跨度row,column）。默认值为 1。

也可以看看[setSpan \(\)](#) 和[rowSpan\(\)](#)。

int QTableView::columnViewportPosition(int column) const

返回给定内容坐标中的 x 坐标column。

int QTableView::columnWidth(int column) const

返回给定的宽度`column`。

也可以看看[setColumnWidth\(\)](#),[resizeColumnToContents \(\)](#) , 和[rowHeight\(\)](#)。

[override virtual protected]void QTableView::currentChanged(const QModelIndex ¤t, const QModelIndex &previous)

重新实现: [QAbstractItemView::currentChanged](#) (常量 `QModelIndex` 和当前, 常量 `QModelIndex` 和上一个) 。

[slot]void QTableView::hideColumn(int column)

隐藏给定的`column`。

也可以看看[showColumn \(\)](#) 和[hideRow\(\)](#)。

[slot]void QTableView::hideRow(int row)

隐藏给定的`row`。

也可以看看[showRow \(\)](#) 和[hideColumn\(\)](#)。

*QHeaderView *QTableView::horizontalHeader() const*

返回表视图的水平标题。

也可以看看[setHorizontalHeader\(\)](#),[verticalHeader \(\)](#) , 和[QAbstractItemModel::headerData\(\)](#)。

[override virtual protected]int QTableView::horizontalOffset() const

重新实现: [QAbstractItemView::horizontalOffset\(\) const](#)。

返回表视图中项目的水平偏移量。

请注意, 表视图使用水平标题部分位置来确定视图中列的位置。

也可以看看[verticalOffset\(\)](#)。

*[override virtual] QModelIndex QTableView::indexAt(const QPoint
&pos) const*

重新实现: [QAbstractItemView::indexAt\(const QPoint &point\) const](#)。

返回位于位置的表项对应的模型项的索引位置 pos 在内容坐标中。

*[override virtual protected]void
QTableView::initViewItemOption(QStyleOptionViewItem *option) const*

重新实现: [QAbstractItemView::initViewItemOption\(QStyleOptionViewItem *option\) const](#)。

bool QTableView::isColumnHidden(int column) const

true如果给定则返回 $column$ 被隐藏; 否则返回false.

也可以看看[isRowHidden\(\)](#)。

*[override virtual protected]bool QTableView::isIndexHidden(const
QModelIndex &index) const*

重新实现: [QAbstractItemView::isIndexHidden\(const QModelIndex &index\) const](#)。

bool QTableView::isRowHidden(int row) const

true如果给定则返回 row 被隐藏; 否则返回false.

也可以看看[isColumnHidden\(\)](#)。

*[override virtual protected] QModelIndex
QTableView::moveCursor(QAbstractItemView::CursorAction cursorAction,
Qt::KeyboardModifiers modifiers)*

重新实现: [QAbstractItemView::moveCursor](#) ([QAbstractItemView::CursorAction](#) 光标操作、[Qt::KeyboardModifiers](#) 修饰符) 。

按照给定的方向移动光标 $cursorAction$, 使用提供的信息 $modifiers$ 。

也可以看看[QAbstractItemView::CursorAction](#)。

*[override virtual protected]void QTableView::paintEvent(QPaintEvent *event)*

重新实现：QAbstractScrollArea::paintEvent (QPaintEvent *事件)。

在收到给定的绘制事件后绘制表格event。

[slot]void QTableView::resizeColumnToContents(int column)

调整给定的大小column基于用于呈现列中每个项目的委托的大小提示。

注意：只有可见列的大小才会调整。重新实现sizeHintForColumn() 也可以调整隐藏列的大小。

也可以看看resizeColumnsToContents(),sizeHintForColumn () , 和QHeaderView::resizeContentsPrecision()。

[slot]void QTableView::resizeColumnsToContents()

根据用于呈现列中每个项目的委托的大小提示来调整所有列的大小。

也可以看看resizeColumnToContents(),sizeHintForColumn () , 和QHeaderView::resizeContentsPrecision()。

[slot]void QTableView::resizeRowToContents(int row)

调整给定的大小row基于用于呈现行中每个项目的委托的大小提示。

也可以看看resizeRowsToContents(),sizeHintForRow () , 和QHeaderView::resizeContentsPrecision()。

[slot]void QTableView::resizeRowsToContents()

根据用于呈现行中每个项目的委托的大小提示来调整所有行的大小。

也可以看看resizeRowToContents(),sizeHintForRow () , 和QHeaderView::resizeContentsPrecision()。

int QTableView::rowAt(int y) const

返回给定 y 坐标所在的行, y, 在内容坐标中位于。

注意：如果给定坐标无效（没有行），则该函数返回 -1。

也可以看看columnAt()。

[protected slot]void QTableView::rowCountChanged(int oldCount, int newCount)

每当添加或删除行时都会调用此槽。先前的行数由以下方式指定`oldCount`，新的行数由下式指定`newCount`。

int QTableView::rowHeight(int row) const

返回给定的高度`row`。

也可以看看[setRowHeight\(\)](#),[resizeRowToContents \(\)](#) , 和[columnWidth\(\)](#)。

[protected slot]void QTableView::rowMoved(int row, int oldIndex, int newIndex)

调用此槽来更改给定的索引`row`在表视图中。旧索引由以下方式指定`oldIndex`，新索引为`newIndex`。

也可以看看[columnMoved\(\)](#)。

[protected slot]void QTableView::rowResized(int row, int oldHeight, int newHeight)

调用此槽来更改给定的高度`row`。旧高度由下式指定`oldHeight`，新高度为`newHeight`。

也可以看看[columnResized\(\)](#)。

int QTableView::rowSpan(int row, int column) const

返回 (处的表元素的行跨度`row,column`) 。默认值为 1。

也可以看看[setSpan \(\)](#) 和[columnSpan\(\)](#)。

int QTableView::rowViewportPosition(int row) const

返回给定内容坐标中的 y 坐标`row`。

[override virtual protected]void QTableView::scrollContentsBy(int dx, int dy)

重新实现： [QAbstractScrollArea::scrollContentsBy](#) (int dx, int dy) 。

将表视图的内容滚动 (dx,dy) 。

[override virtual]void QTableView::scrollTo(const [QModelIndex](#) &index, [QAbstractItemView::ScrollHint](#) hint = EnsureVisible)

重新实现： [QAbstractItemView::scrollTo](#) (const QModelIndex &index, QAbstractItemView::ScrollHint 提示) 。

确保给定的index在表视图中可见，必要时可滚动。

[slot]void QTableView::selectColumn(int column)

选择给定的column如果当前 SelectionMode 和 SelectionBehavior 允许选择列，则在表视图中。

也可以看看[selectRow\(\)](#)。

[slot]void QTableView::selectRow(int row)

选择给定的row如果当前 SelectionMode 和 SelectionBehavior 允许选择行，则在表视图中。

也可以看看[selectColumn\(\)](#)。

*[override virtual protected][QModelIndexList](#)
QTableView::selectedIndexes() const*

重新实现： [QAbstractItemView::selectedIndexes\(\)](#) const。

[override virtual protected]void QTableView::selectionChanged(const [QItemSelection](#) &selected, const [QItemSelection](#) &deselected)

重新实现： [QAbstractItemView::selectionChanged](#) (常量 QItemSelection 和已选择，常量 QItemSelection 和取消选择) 。

void QTableView::setColumnHidden(int column, bool hide)

如果`hide`给定的为真`column`将被隐藏；否则将会显示。

也可以看看[isColumnHidden \(\)](#) 和[setRowHidden\(\)](#)。

void QTableView::setColumnWidth(int column, int width)

设置给定的宽度`column`成为`width`。

也可以看看[columnWidth\(\)](#)。

*void QTableView::setHorizontalHeader(QHeaderView *header)*

将用于水平标题的小部件设置为`header`。

也可以看看[horizontalHeader \(\)](#) 和[setVerticalHeader\(\)](#)。

*[override virtual]void QTableView::setModel(QAbstractItemModel
model)

重新实现：[QAbstractItemView::setModel](#) ([QAbstractItemModel *模型](#)) 。

*[override virtual]void QTableView::setRootIndex(const QModelIndex
&index)*

重新实现：[QAbstractItemView::setRootIndex](#) (常量 [QModelIndex &index](#)) 。

void QTableView::setRowHeight(int row, int height)

设置给定的高度`row`成为`height`。

也可以看看[rowHeight\(\)](#)。

void QTableView::setRowHidden(int row, bool hide)

如果`hide`是真的`row`将被隐藏，否则将显示。

也可以看看[isRowHidden \(\)](#) 和[setColumnHidden\(\)](#)。

*[override virtual protected]void QTableView::setSelection(const QRect
&rect, QItemSelectionModel::SelectionFlags flags)*

重新实现：[QAbstractItemView::setSelection](#) (const QRect &rect, QItemSelectionModel::SelectionFlags 标志) 。

选择给定范围内的项目`rect`并按照指定的选择`flags`。

*[override virtual]void
QTableView::setSelectionModel(QItemSelectionModel *selectionModel)*

重新实现：[QAbstractItemView::setSelectionModel](#)(QItemSelectionModel *selectionModel)。

void QTableView::setSortingEnabled(bool enable)

如果`enable`为 true，启用表排序并立即触发调用[sortByColumn\(\)](#) 与当前排序部分和顺序

注意：属性的 Setter 函数[sortingEnabled](#)。

也可以看看[isSortingEnabled\(\)](#)。

*void QTableView::setSpan(int row, int column, int rowSpanCount, int
columnSpanCount)*

将表格元素的跨度设置为 (row,column) 到 (指定的行数和列数rowSpanCount,columnSpanCount) 。

也可以看看[rowSpan \(\)](#) 和[columnSpan\(\)](#)。

*void QTableView::setVerticalHeader(QHeaderView *header)*

将用于垂直标题的小部件设置为`header`。

也可以看看[verticalHeader \(\)](#) 和[setHorizontalHeader\(\)](#)。

[slot]void QTableView::showColumn(int column)

显示给定的`column`。

也可以看看[hideColumn \(\)](#) 和[showRow\(\)](#)。

[slot]void QTableView::showRow(int row)

显示给定的`row`。

也可以看看[hideRow \(\)](#) 和[showColumn\(\)](#)。

[override virtual protected]int QTableView::sizeHintForColumn(int column) const

重新实现: [QAbstractItemView::sizeHintForColumn\(int column\) const](#)。

返回给定的大小提示`column`的宽度，如果没有模型，则为 -1。

如果需要将给定列的宽度设置为固定值，请调用[QHeaderView::resizeSection\(\)](#) 在表格的水平标题上。

如果您在子类中重新实现此函数，请注意，您返回的值将在以下情况下使用: [resizeColumnToContents \(\)](#) 或者 [QHeaderView::resizeSections \(\)](#) 叫做。如果水平标题或项目委托需要更大的列宽，则将使用更大的宽度。

也可以看看[QWidget::sizeHint, horizontalHeader \(\)](#) , 和[QHeaderView::resizeContentsPrecision\(\)](#)。

[override virtual protected]int QTableView::sizeHintForRow(int row) const

重新实现: [QAbstractItemView::sizeHintForRow\(int row\) const](#)。

返回给定的大小提示`row`的高度，如果没有模型，则为 -1。

如果需要将给定行的高度设置为固定值，请调用[QHeaderView::resizeSection\(\)](#) 位于表格的垂直标题上。

如果您在子类中重新实现此函数，请注意您返回的值仅在以下情况下使用: [resizeRowToContents \(\)](#) 叫做。在这种情况下，如果垂直标题或项目委托需要更大的行高，则将使用该宽度。

也可以看看[QWidget::sizeHint, verticalHeader \(\)](#) , 和[QHeaderView::resizeContentsPrecision\(\)](#)。

[slot]void QTableView::sortByColumn(int column, [Qt::SortOrder](#) order)

按给定值对模型进行排序`column`和`order`。

`column`可能是 -1，在这种情况下，不会显示排序指示符，并且模型将返回到其自然的、未排序的顺序。请注意，并非所有型号都支持此功能，在这种情况下甚至可能会崩溃。

也可以看看[sortingEnabled](#)。

*[override virtual protected]void QTableView::timerEvent([QTimerEvent](#) *event)*

重新实现：[QAbstractItemView::timerEvent](#) ([QTimerEvent](#) *事件)。

[override virtual protected]void QTableView::updateGeometries()

重新实现：[QAbstractItemView::updateGeometries\(\)](#)。

*[QHeaderView](#) *QTableView::verticalHeader() const*

返回表视图的垂直标题。

也可以看看[setVerticalHeader\(\)](#),[horizontalHeader](#) () , 和[QAbstractItemModel::headerData\(\)](#)。

[override virtual protected]int QTableView::verticalOffset() const

重新实现：[QAbstractItemView::verticalOffset\(\) const](#)。

返回表视图中项目的垂直偏移量。

请注意，表视图使用垂直标题部分位置来确定视图中行的位置。

也可以看看[horizontalOffset\(\)](#)。

*[override virtual protected] QSize QTableView::viewportSizeHint()
const*

重新实现: `QAbstractItemView::viewportSizeHint() const`。

*[override virtual] QRect QTableView::visualRect(const QModelIndex
&index) const*

重新实现: `QAbstractItemView::visualRect(const QModelIndex &index) const`。

返回给定的视口上占据的矩形`index`。如果索引隐藏在视图中，它将返回 `nullQRect`。

*[override virtual protected] QRegion
QTableView::visualRegionForSelection(const QItemSelection &selection)
const*

重新实现: `QAbstractItemView::visualRegionForSelection(const QItemSelection &selection) const`。

从给定项目的视口返回矩形`selection`。

从 4.7 开始，返回的区域仅包含与视口相交（或包含在视口中）的矩形。