#### CATableViewDataSource

###### 类说明

[CATableView](../CCObject/CAView/CAScrollView/CATableView.docx)的数据代理类，定义tableView的数据接口。

###### 方法

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 方法名 | 说明 |
| public | [tableCellAtIndex](#tableCellAtIndex) | 获取指定cell |
| public | [numberOfRowsInSection](#numberOfRowsInSection) | 获取对应的section所包含的cell个数 |
| public | [numberOfSections](#numberOfSections) | 获取tableview包含的section个数 |
| public | [tableViewHeightForRowAtIndexPath](#tableViewHeightForRowAtIndexPath) | 获取指定的cell高度 |
| public | [tableViewHeightForHeaderInSection](#tableViewHeightForHeaderInSection) | 获取指定的section的header view的高度 |
| public | [tableViewHeightForFooterInSection](#tableViewHeightForFooterInSection) | 获取指定的section的footer view的高度 |

###### 方法介绍

**virtual CATableViewCell\*** **tableCellAtIndex(CATableView\* table, const CCSize& cellSize，unsigned int section, unsigned int row)**

返回值：CATableViewCell\*

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CATableView\* | table | 当前tableView |
| const CCSize | cellSize | cell的size |
| unsigned int | section | cell所属的区域 |
| unsigned int | row | cell所在行数 |

解释：为tableView创建cell，这里采用复用的方式来创建cell。复用即当我们cell滑出界面时，对于新滑入的cell，我们不再创建，而是重复利用已经创建且滑出界面的cell。

示例：

|  |
| --- |
| CATableViewCell\* TableViewTest::tableCellAtIndex(CATableView\* table, const CCSize& cellSize, unsigned int section, unsigned int row)  {  CATableViewCell\* cell = table->dequeueReusableCellWithIdentifier("CrossApp");  if (cell == NULL)  {  cell = CATableViewCell::create("CrossApp");  CALabel\* cellText = CALabel::createWithCenter(CCRect(cellSize.width\*0.1, cellSize.height\*0.5, cellSize.width\*0.3, cellSize.height\*0.8));  cellText->setTag(100);  cellText->setFontSize(30 \* CROSSAPP\_ADPTATION\_RATIO);  cellText->setColor(CAColor\_blueStyle);  cellText->setTextAlignment(CATextAlignmentCenter);  cellText->setVerticalTextAlignmet(CAVerticalTextAlignmentCenter);  cell->addSubview(cellText);  CAButton\* cellBtn = CAButton::createWithCenter(CCRect(cellSize.width\*0.8,cellSize.height\*0.5,cellSize.width\*0.2,cellSize.height\*0.5),CAButtonTypeRoundedRect);  cellBtn->setTag(102);  cellBtn->setTitleForState(CAControlStateAll,"Touch");  cellBtn->addTarget(this, CAControl\_selector(TableViewTest::cellBtnCallback), CAControlEventTouchUpInSide);  cell->addSubview(cellBtn);  }  char order[20] = "";  sprintf(order, "cell-%d", row);  CALabel\* cellText = (CALabel\*)cell->getSubviewByTag(100);  cellText->setText(order);  return cell;  } |

从以上创建cell的方式可以看出，在生成cell时，我们首先从复用队列中根据指定的复用标示符去寻找是否存在这样的cell，如果不存在，则创建一个cell，并指定复用标识符。

通常情况下cell本身的界面是不会发生变化，但cell所展现的数据往往在变化。所以根据实际情况，对于cell来说，界面的定制一般在创建新的cell时规定，而对于cell展现的数据，不能在创建cell时固定。

**virtual unsigned int** **numberOfRowsInSection(CATableView \*table, unsigned int**

**section)**

返回值：unsigned int

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CATableView\* | table | 当前tableView |
| unsigned int | section | cell所属的区域 |

**virtual unsigned int** **numberOfSections(CATableView \*table)**

返回值：unsigned int

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CATableView\* | table | 当前tableView |

**virtual unsigned int** **tableViewHeightForRowAtIndexPath(CATableView\* table,**

**unsigned int section, unsigned int row)**

返回值：unsigned int

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CATableView\* | table | 当前tableView |
| unsigned int | section | cell所属的区域 |
| unsigned int | row | cell所在行数 |

**virtual unsigned int** **tableViewHeightForHeaderInSection(CATableView\* table,**

**unsigned int section)**

返回值：unsigned int

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CATableView\* | table | 当前tableView |
| unsigned int | section | cell所属的区域 |

**virtual unsigned int** **tableViewHeightForFooterInSection(CATableView\* table,**

**unsigned int section)**

返回值：unsigned int

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CATableView\* | table | 当前tableView |
| unsigned int | section | cell所属的区域 |