# 1.在学习ListView扩展之前先了解一下SwipeView

实例

|  |
| --- |
| 项目结构 |

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  Window {  width: 640  height: 480  visible: true  title: *qsTr*("ListView扩展")  SwipeView {  id: *view*  currentIndex: 1  anchors.fill: *parent*  Item {  id: *firstPage*  // Rectangle{  // anchors.fill: parent  // color: "lime"  // }  Image{  source: "img/congq.jpg"  anchors.fill: *parent*  }  }  Item {  id: *secondPage*  // Rectangle{  // anchors.fill: parent  // color: "blue"  // }  Image{  source: "img/jcc.jpg"  anchors.fill: *parent*  }  }  Item {  id: *thirdPage*  // Rectangle{  // anchors.fill: parent  // color: "orange"  // }  Image{  source: "img/ctwheatfield.jpg"  anchors.fill: *parent*  }  }  }  PageIndicator {  id: *indicator*  count: *view*.count  currentIndex: *view*.currentIndex  anchors.bottom: *view*.bottom  anchors.horizontalCenter: *parent*.horizontalCenter  }  } | 默认显示第二页因为设置了index=1也就是第二个  向右移动显示第一个页面    以此类推。。。 |

## 注意：SwipeView有一个不好的地方就是如果直接给它设置宽高，它就会失效！！也就是它只支持anchors.xx的设置方法

# 2.还需要学习ObjectModel。

ListView除了可以使用ListModel填充外，还可以使用ObjectModel来填充

## 简单使用，

注意ObjectModel需要导入QtQml.Models 2.1

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQml.Models 2.1  Window {  width: 640  height: 480  visible: true  title: *qsTr*("ListView扩展")  Rectangle {  width: 100  height: 200  ObjectModel {  id: *itemModel*  Rectangle {  height: 30;  width: 80;  color: "red"  }  Rectangle {  height: 30;  width: 80;  color: "green"  }  Rectangle {  height: 30;  width: 80;  color: "blue"  }  }  ListView {  anchors.fill: *parent*  model: *itemModel*  }  }  } |  |

## 原来ListView可以修改item的滑动方向为横向orientation: ListView.Horizontal

，这样子就和SwipeView有类似的效果

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQml.Models 2.1  Window {  width: 640  height: 480  visible: true  title: *qsTr*("ListView扩展")  //利用ListView实现类似SwipeView的效果  ObjectModel{  id:*objmodel*  Image{  source: "img/congq.jpg"  width: 600  height: 400  }  Image{  source: "img/ctwheatfield.jpg"  width: 600  height: 400  }  Image{  source: "img/jcc.jpg"  width: 600  height: 400  }  }  ListView{  width: 600  height: 400  spacing: 10  model:*objmodel*  orientation: ListView.Horizontal  }  } | 初始状态，可以左右滑动    。。。 |

## 如果只需要滑动屏幕的一部分，另外一部分不能滑动怎么办？

使用ListView的snapMode属性，还需要修改不能滑动部分的z值为1，或者将ListView的clip属性修改为true（较好）

例子：和上面修改一样

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQml.Models 2.1  Window {  width: 640  height: 480  visible: true  title: *qsTr*("ListView扩展")  //利用ListView实现类似SwipeView的效果  Button{  id: *btn*  width: 150  height: 480  //z:1  }  ObjectModel{  id:*objmodel*  Image{  source: "img/congq.jpg"  width: 480  height: 480  }  Image{  source: "img/ctwheatfield.jpg"  width: 480  height: 480  }  Image{  source: "img/jcc.jpg"  width: 480  height: 480  }  }  ListView{  anchors.left : *btn*.right  anchors.right :*parent*.right  height: 480  spacing: 10  model:*objmodel*  orientation: ListView.Horizontal  snapMode: ListView.SnapOneItem  clip: true  }  } |  |

# 3. XmlListModel，用的较多

好处，可以使用外部的xml文件来配置ListView的控件属性

需要：**import QtQuick.XmlListModel 2.0**

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQml.Models 2.1  import QtQuick.XmlListModel 2.0  Window {  width: 640  height: 480  visible: true  title: *qsTr*("ListView扩展")  //利用XmlListModel来填充ListView  XmlListModel{  id:*xmlMd*  source: "./feed.xml"  query:"/rss/channel/item"  XmlRole{  name:"title"  query:"title/string()"  }  XmlRole{  name:"pubDate"  query: "pubDate/string()" //获取xml中的字符串  }  XmlRole{  name:"score"  query: "score/number()" //获取xml的数值  }  }  ListView{  model:*xmlMd*  width: 180; height: 300  delegate: Text {  text: *title* +" : " +pubDate + " : "+score  }  }  } |  |