

Qt Developer Conference 2011





QtQuick 培训课程



第三部分



目标

1 关于lists

什么是model?

QtQuick中的Models

Listview

GridView

动态创建model

2 加载外部数据

从网络上加载到XML model

SQLite 数据库



第三部分

主题

1 关于lists

2 加载外部数据

3 问题解答

4 练习



关于 lists

Models 和 Lists

QtQuick中的Models

ListView

GridView

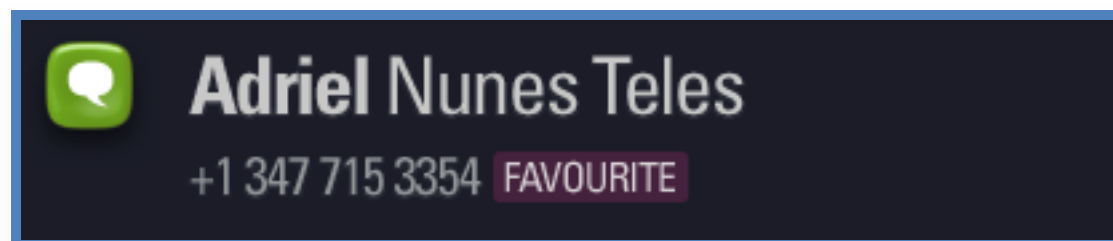
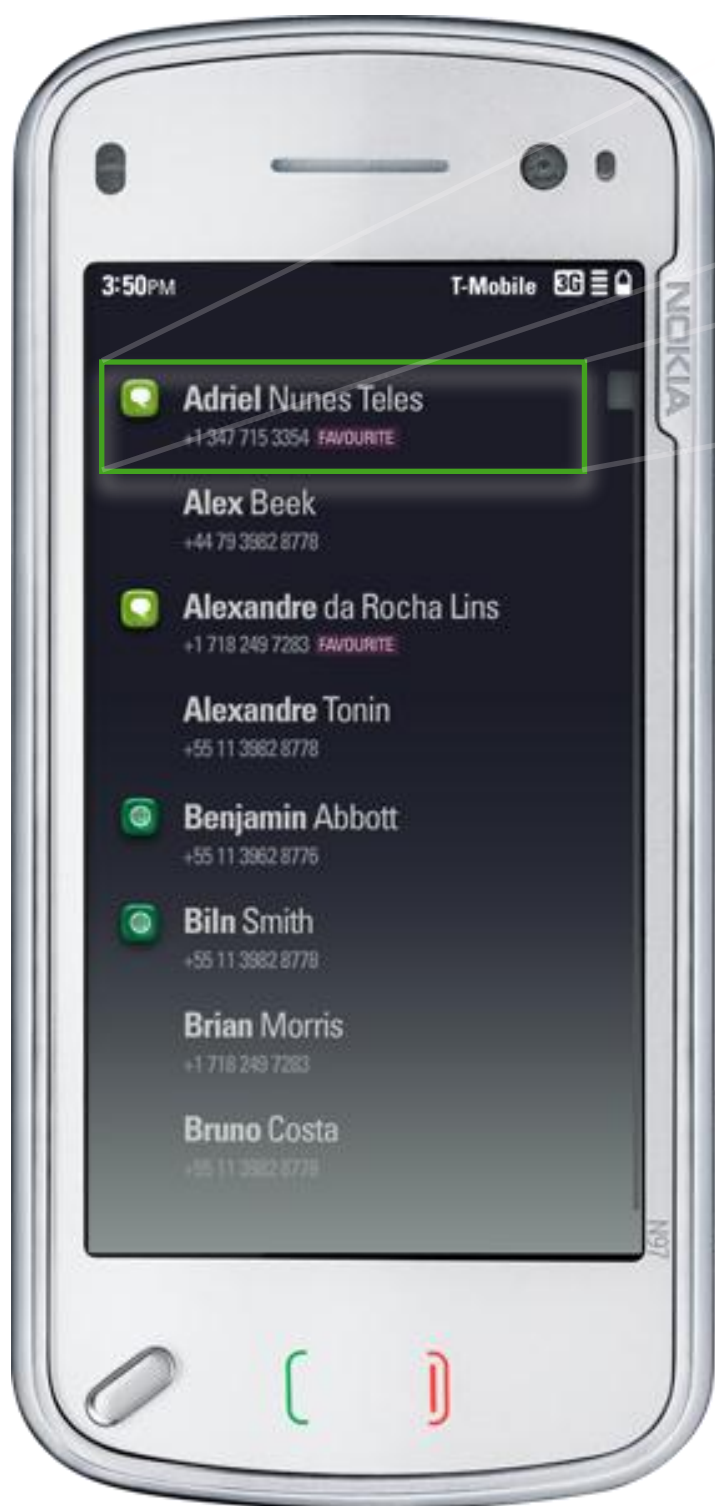
Javascript 和 ListModel

list items的动画效果



关于lists

什么是model?



一个item的全部数据段

名

姓

电话号码

是否为常用联系人

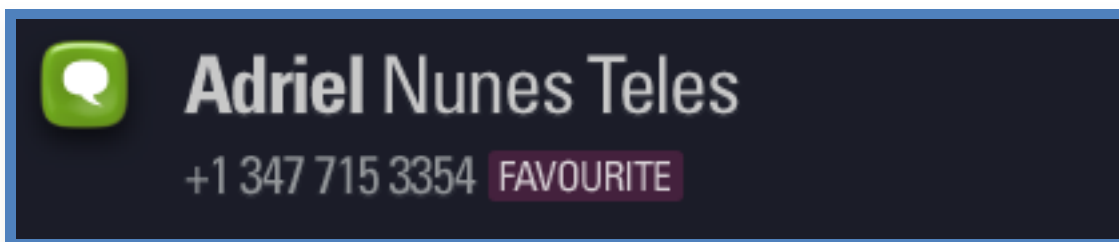
图标



关于lists

QtQuick中的Models

下面代码说明: 在QtQuick model中, 单个item如何组织的. ListModel是一个存放ListElement的数组



```
ListModel {  
    ListElement {  
        picture: "images/user_1.png"  
        name: "Adriel"  
        surname: "Nunes Teles"  
        telephone: "+1 347 715 3354"  
        favorite: true  
        icon: "sms"  
    }  
}
```

查看示例: [addon/module-003/examples/ListModelExample.qml](#)



关于lists

Models 和 Lists

QtQuick 中的Models

ListView

GridView

Javascript 和 ListModel

list items 的动画效果



关于lists

Listview

在QtQuick中创建lists, 首先要组织好各个文件, 这里至少需要三个文件。

描述单个item视图的文件

该文件描述每个list item中各个可视化的元素及它们是如何显现的

描述ListModel的文件

该文件描述要被显示的数据, 所有数据以ListElement格式存放在数组中, 每个ListElement包含单个item中全部信息。

主文件

该文件包含ListView的调用, ListModel和多个单独item视图



关于lists

ListView

首先要创建一个单独item布局的文件...



```
...  
Text {  
    id: nameTxt  
    font.family: "Univers LT Std"  
    color: "#c8c8c8"  
    font.pixelSize: 22  
    text: "<b>Adriel</b> Nunes Teles"  
    anchors.left: icon.right  
    anchors.top: icon.top  
    anchors.topMargin: 8  
}  
...
```

查看示例: [addon/module-003/examples/ItemRenderDelegate.qml](#)



关于lists

Listview

... 然后使item足够灵活可以接受必要的变化

```
...
property string intern_name
property string intern_surname
property string intern_telephone
...

Text {
    id: nameTxt
    font.family: "Univers LT Std"
    color: "#c8c8c8"
    font.pixelSize: 22
    text: "<b>" + intern_name + "</b> " + intern_surname
    anchors.left: icon.right
    anchors.top: icon.top
    anchors.topMargin: 8
}
...
```

查看示例: [addon/module-003/examples/ListItemDelegate.qml](#)



关于lists

ListView

完成了上面的步骤，你只需要在主文件中设置ListView即可

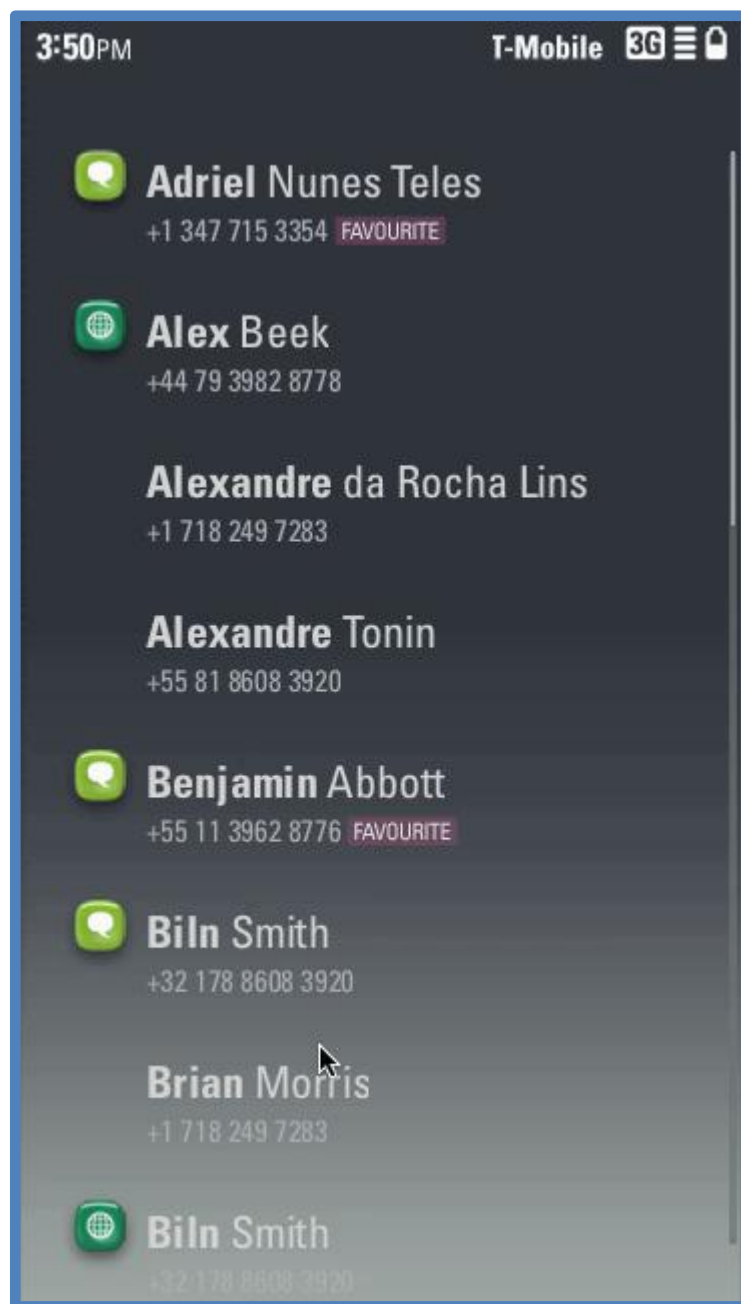
```
ListView {  
    id: listExample  
    orientation: "Vertical"  
    model: ListModelExample {}  
    delegate: itemComponent  
}  
Component {  
    id: itemComponent  
    ListItemExample {  
        intern_name: model.name  
        intern_surname: model.surname  
        intern_telephone: model.telephone  
        intern_icon: model.icon  
    }  
}  
...
```

查看示例: [addon/module-003/examples/ListExample.qml](#)



关于lists

ListView



查看视频: [addon/module-003/videos/list-view.mov](#)

查看示例: [addon/module-003/examples/ListExample.qml](#)



关于lists

Models 和 Lists

QtQuick中的Models

ListView

GridView

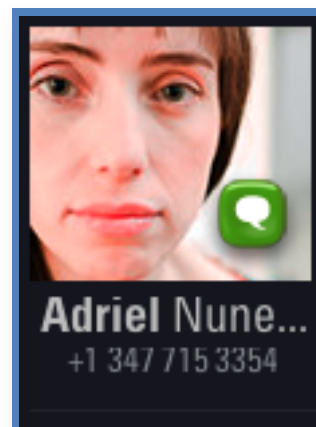
Javascript 和ListModel

list items的动画效果



关于lists

GridView

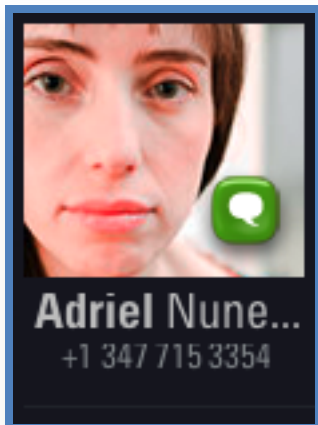


已经有了model, 只需要创建不同的视图显示就可以了



关于lists

GridView



我们要做的 就是重新组织这些元素并
把其他视图中没有使用的picture 元素
显示出来

```
...  
property string intern_name  
property string intern_surname  
property string intern_picture  
property string intern_icon  
property string intern_telephone  
...  
Image {  
    id: picture  
    anchors.top: parent.top  
    anchors.topMargin: 15  
    source: intern_picture  
}  
...
```

查看示例: <addon/module-003/examples/GridItemDelegate.qml>



关于lists

GridView

通过调用GridView的元素并设置它特定的参数，新的视图就完成了

```
...
GridView {
    id: listExample
    cellWidth: 110
    cellHeight: 160
    anchors.fill: parent
    model: ListModelExample {}
    delegate: itemComponent
}
Component {
    id: itemComponent
    GridItemDelegate {
        intern_name: model.name
        intern_surname: model.surname
        intern_picture: model.picture
    }
}
...
```

查看示例: [addon/module-003/examples/GridExample.qml](#)



关于lists

GridView



查看视频: [addon/module-003/videos/grid-view.mov](#)

查看示例: [addon/module-003/examples/GridExample.qml](#)



关于lists

Models 和Lists

QtQuick中的Models

ListView

GridView

Javascript和ListModel

list items的动画效果



使用Javascript 变更列表的元素

通过ListModel 的属性和javascript, 能方便地添加或更改列表的元素.
主要的ListModel属性:

append – 在列表的末端添加一个新的item

get – 返回在特定索引位置Item的属性值个

set or **setProperty** - 改变特定索引位置的item值或该item的属性值

insert – 在特定索引位置添加一个新的

remove – 删除一个特定索引位置上的item

clear – 清除model中的所有数据

你能够从下面的例子中看到上面这些操作的作用

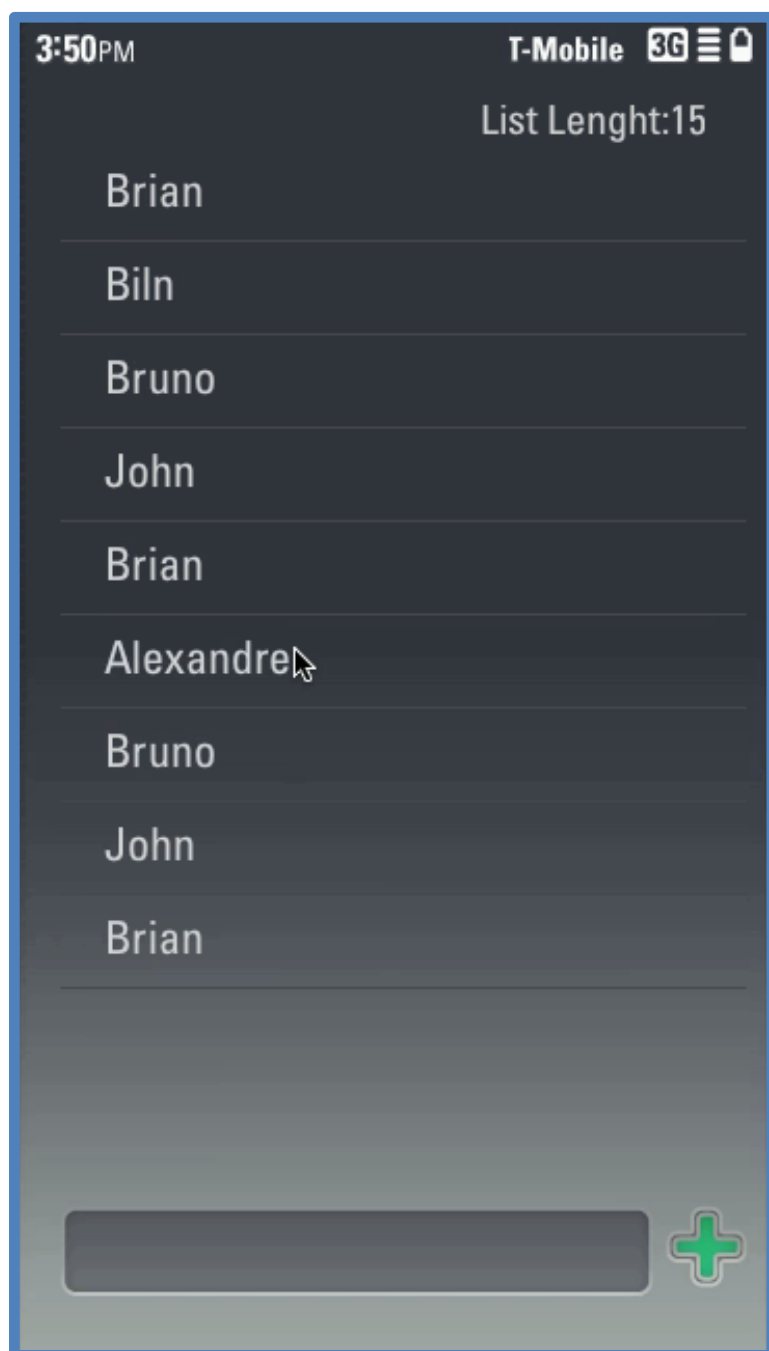
了解更多ListModel 的属性:

<http://doc.qt.nokia.com/4.7-snapshot/qml-listmodel.html>



关于lists

使用 Javascript 添加列表元素



查看视频: <addon/module-003/videos/add.mov>

查看示例: <addon/module-003/examples/ListProperties.qml>

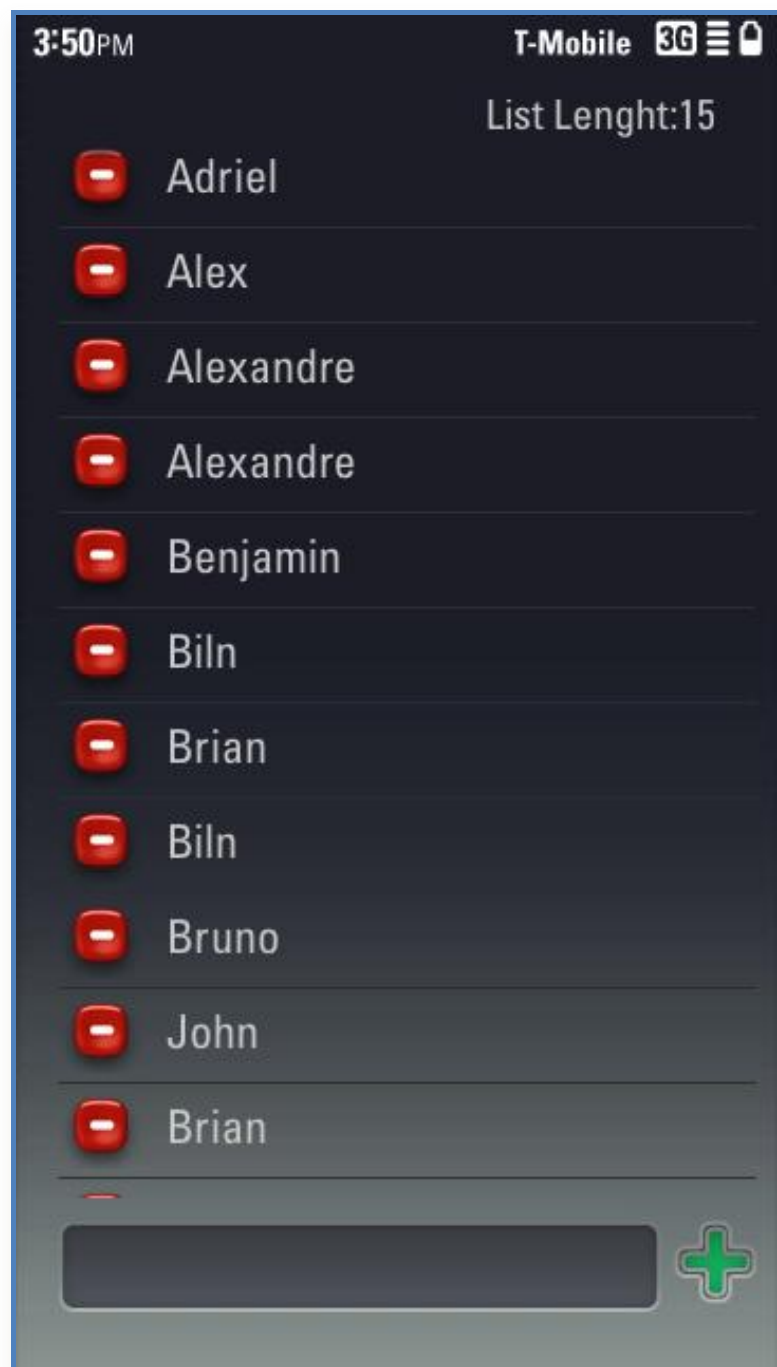
通过通讯录名字和添加一个输入框对列表做稍许改变,说明在列表上添加元素是比较简单的

```
...  
MouseArea{  
    anchors.fill: parent  
    onClicked: {  
        if(inputText.text != "") {  
            listModel.insert(0, {"name": inputText.text});  
            inputText.text = "";  
        }  
    }  
}  
...  
insert
```



关于lists

使用Javascript to 删除列表元素



查看视频: [addon/module-003/videos/remove.mov](#)

在列表item上添加一个按钮和简单的操作, 可以看出remove属性的作用.

```
...
Image {
    id: deleteButton
    source: "images/delete.png"
    anchors.left: parent.left
    remove

    MouseArea {
        anchors.fill: parent
        onClicked: listModel.remove(index)
    }
}
...
```

查看示例: [addon/module-003/examples/ListPropertiesDelete.qml](#)



关于lists

Models 和 Lists

QtQuick中的Models

ListView

GridView

Javascript 和 ListModel

list items的动画效果



关于lists

采用动画如何？

对插入和删除items添加动画是比较简单的. ListView 元素有其属性可以处理这类问题.

ListView.onAdd / ListView.onRemove

```
...
ListView.onAdd: SequentialAnimation {
    PropertyAction { target: listItem; property: "height"; value: 0; }
    PropertyAction { target: listItem; property: "opacity"; value: 0.0; }
    NumberAnimation { target: listItem; property: "height"; to: 45; duration: 300; }
    NumberAnimation { target: listItem; property: "opacity"; to: 1.0; duration: 300; }
}

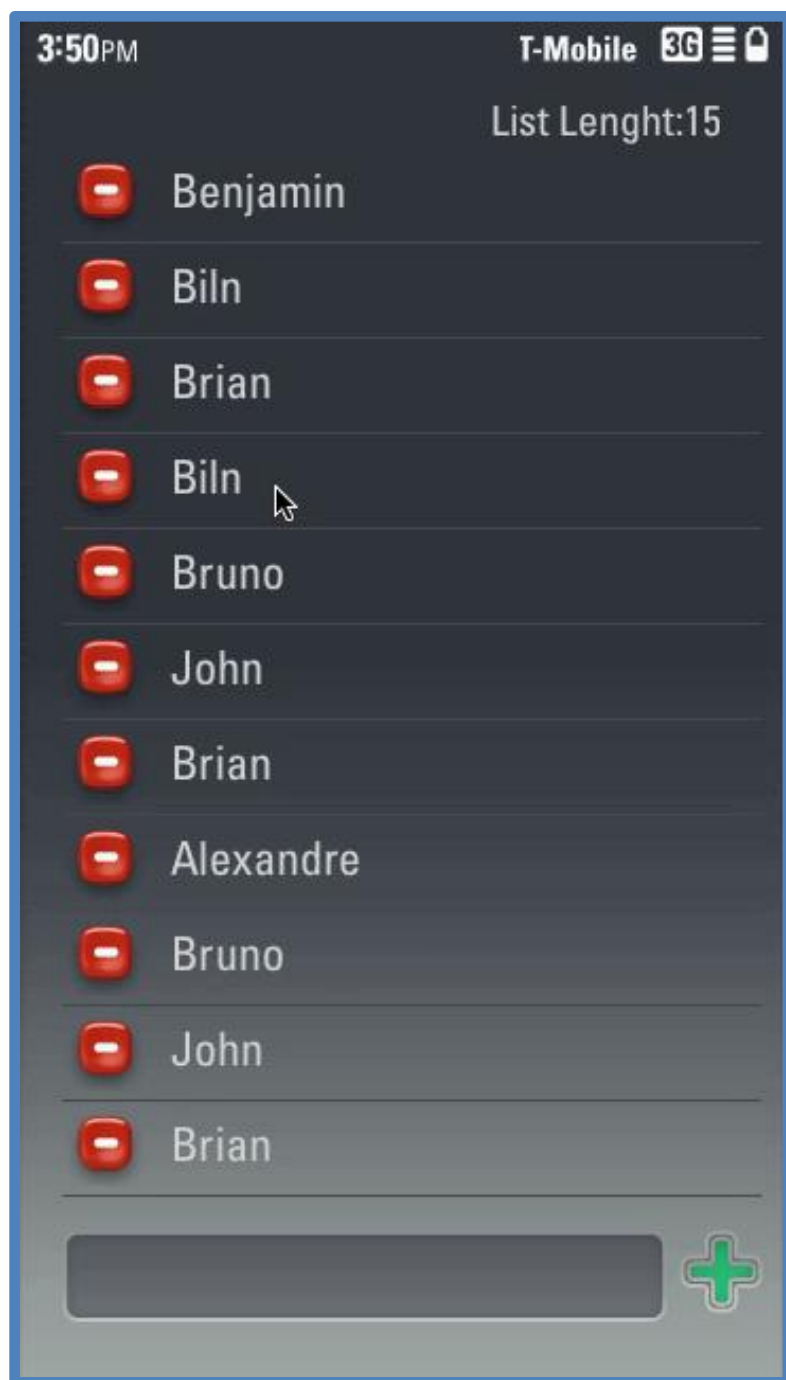
ListView.onRemove: SequentialAnimation {
    PropertyAction { target: listItem; property: "ListView.delayRemove"; value: true; }
    NumberAnimation { target: listItem; property: "opacity"; to: 0.0; duration: 300; }
    NumberAnimation { target: listItem; property: "height"; to: 0; duration: 300; }
    PropertyAction { target: listItem; property: "ListView.delayRemove"; value: false; }
}
...
```

查看示例: [addon/module-003/examples/ListPropertiesMotion.qml](#)



关于lists

采用动画如何？



查看视频: [addon/module-003/videos/list-motion.mov](#)

查看示例: [addon/module-003/examples/ListPropertiesMotion.qml](#)

下一节我们会讨论更多ListView的功能，但是上一个的示例是进行其他开发的一个基础

了解更多关于ListView:

<http://doc.qt.nokia.com/4.7-snapshot/qml-listview.html>



第三部分

主题

1 关于lists

2 加载外部数据

3 问题解答

4 练习



加载外部数据

XmlListModel



Using the same view from the previous examples and just changing how the models are acquired 使用上一个示例相同的视图, 只是改变model的获取方法, 你可以使用 Twitter public timeline feed 创建一个列表

```
...
XmlListModel {
    id: xmlModel
    source: "http://twitter.com/favorites/113425681.rss"
    query: "/rss/channel/item"

    XmlRole { name: "pubDate"; query: "pubDate/string()" }
    XmlRole { name: "desc"; query: "description/string()" }
}
...
```

查看视频: [addon/module-003/videos/xmlmodel.mov](#)

查看示例: [addon/module-003/examples/ListXMLExample.qml](#)



加载外部数据

XmlListModel 属性

QtQuick可以解析XML内容到ListModel元素中. 这是一个与上一个示例中相同的RSS文件

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:atom="http://www.w3.org/2005/Atom" version="2.0" >
  <channel>
    <title>Twitter / Favorites from toptweets</title>
    <link>http://twitter.com/toptweets/favorites</link>
    ...
  <item>
    ...
    <description>planetjedward: Edward has torn his knee ligaments and will have to get surgery. He is going to make a quick recovery and we are (cont)</description>
    <pubDate>Mon, 05 Jul 2010 13:03:11 +0000</pubDate>
    ...
  </item>
  ...
</rss>
```

在上一个示例中, 来自 “<item>”子元素 “<description>” 和 “<pubDate>”的信息是必须的.



XmlListModel 属性

为了明确哪些是建立ListModel必须的, 下面利用从上一个简化来的RSS文件说明XmlListModel如何工作

```
<rss>
  <channel>
    <item>
      <description></description>
      <pubDate></pubDate>
    </item>
  </channel>
</rss>
```

RSS

```
XmlListModel {
  id: xmlModel
  source: "http://twitter.com/favorites/113425681.rss"
  query: "/rss/channel/item"

  XmlRole { name: "pubDate"; query: "pubDate/string()" }
  XmlRole { name: "desc"; query: "description/string()" }
}
```

XmlListModel

query: "/rss/channel/item" – 设定model创建内容的范围。这里只创建 “<item>” 的子元素

XmlRole { name: "pubDate"; query: "pubDate/string()" } - XmlRole就像是 ListElement中值. 所有在XmlRoles 中的数据都会以ListElement的内容呈现。



加载外部数据

SQLite 数据库



这个示例演示如何在QtQuick应用程序中创建持久化数据. 如果你在SQLite中创建过数据库, 这里的示例就是你熟悉的...

```
...
db = openDatabaseSync("Contacts", "1.0", "Contacts SQL",
1000000);
db.transaction(function(tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS
ContactsData(id INT, name STRING)');
    resultSet = tx.executeSql('SELECT * FROM ContactsData
ORDER BY id DESC');
});
...
```

查看视频: <addon/module-003/videos/sqlite-example.mov>

查看示例: <addon/module-003/examples/ListDBExample.qml>



加载外部数据

SQLite 数据库

...你就知道了如何使用SQLite数据库中的数据构成一个ListModel

```
...  
function populateModel(){  
    for(var i = 0; i<=resultSet.rows.length-1; i++){  
        var nameTxt = resultSet.rows.item(i).name;  
        var indexNum = resultSet.rows.item(i).id;  
        listModel.append({id:indexNum, name: nameTxt });  
    }  
}  
...
```

你只需要用“append” 添加一个ListElement 在ListModel的末端即可。



第三部分

主题

1 关于lists

2 加载外部数据

3 问题解答

4 练习



什么是Model?

什么是ListModel?

创建一个ListView需要什么?

如何添加或删除list items?

如何实现在list中添加item时带有动画的效果?

什么是XmlRole?

哪个ListModel方法可以在list末端添加一个Item?



第三部分

主题

1 关于lists

2 加载外部数据

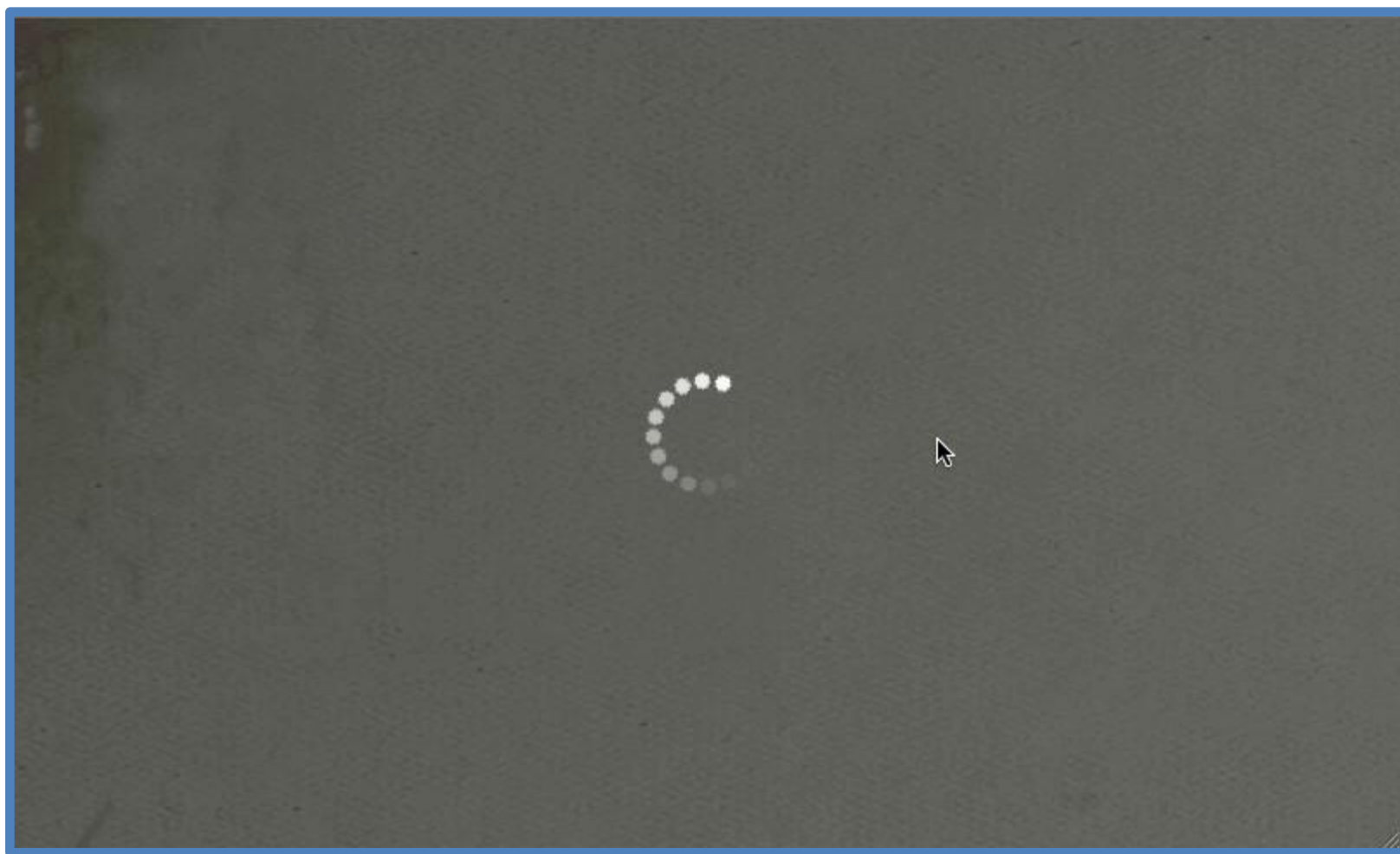
3 问题解答

4 练习



练习

使用XmlModelList和twitter public timeline XML创建一个水平列表.
尝试使用ListView的snapMode属性



查看视频: [addon/module-003/videos/twitter_app.mov](#)

选作部分: 创建一个动态加载过程

查看练习: [labs/lab-three/lab-twitter/labThree.qmlproject](#)



版权 © (2011)属于Nokia公司及其附属公司，Nokia公司及其附属公司保留全部权利。

包含附带的Qt 培训材料都是根据Creative Commons Attribution ShareAlike 2.5 License Agreement发布.



完整的证书参见:<http://creativecommons.org/licenses/by-sa/2.5/legalcode>

Nokia, Qt 和 所有基于 Nokia 和 Qt 的标志是Nokia公司在芬兰及全球其他国家的商标或注册商标.