QML TabBar with accessible off-screen tabs

This article demonstrates how to create a kinetic tab bar which can effectively scale to many tabs, using `TabBarLayout`.

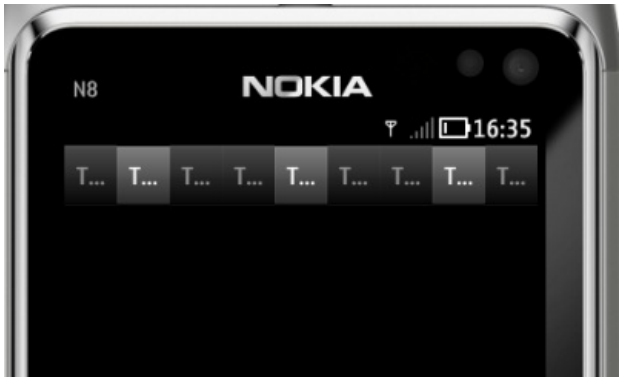
Introduction

Tab bars are a good choice for showing your content categorised into separate sections. In the Qt Quick Components documents, the QML `TabGroup` Element allows developer to create a simple `TabGroup` with three tab page. The example does not scale well to more tabs.

First, take a look at original code sample from the document

```
TabBarLayout {
         id: tabBarLayout
         anchors { left: parent.left; right: parent.right; top: parent.top }
         TabButton { tab: tab1content; text: "Tab 1" }
         TabButton { tab: tab2content; text: "Tab 2" }
         TabButton { tab: tab3content; text: "Tab 3" }
     }
```

What will happen if I have more content tabs, the frame is not enough space to show tab's title clearly.



This article shows how you can create a tab bar with off screen tabs that you can navigate to.

Implementation

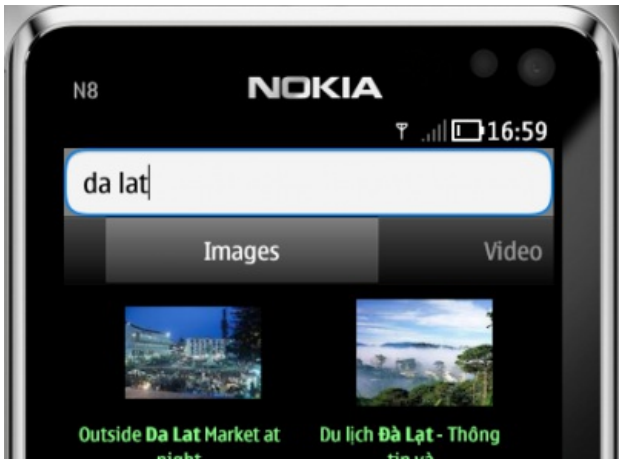
In this case, I will implement a `ListView` inside the `TabBarLayout` like the code below:

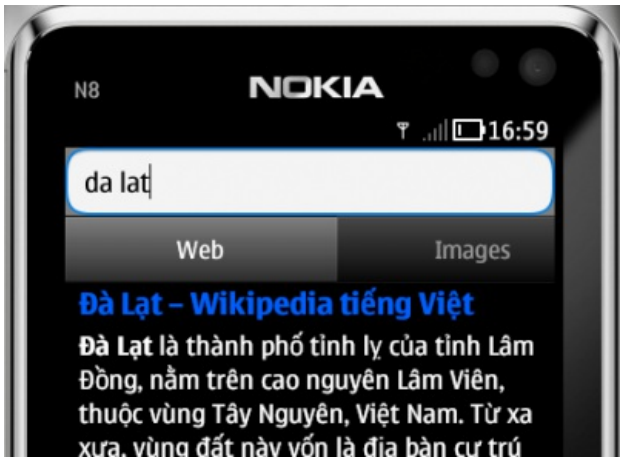
```
TabBarLayout {
         id: tabBarLayout
         anchors { left: parent.left; right: parent.right; top: parent.top }

ListView {
                ListModel {
                    id: searchModel
                    ListElement { type: "Web"; }
                    ListElement { type: "Images"; }
                    ListElement { type: "Video"; }
                    ListElement { type: "Blog"; }
                    ListElement { type: "Book"; }
                    //... more category as you want
                }
                anchors.fill: parent
                model: searchModel
                orientation: ListView.Horizontal
                delegate:TabButton  {
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    width: 200
                    height: 50
                    tab: {
                        if(index == 0){
                              return tab1content
                        }else if(index == 1){
                              return tab2content
                        }
                        //... set up the page content or you can let a delegate model
                    }
                    text: type
                }
        }
}
```





Ok, now I'm going to next step: set up content in each page. Like the sample from the document, TabGroup contains the pages's content. Depending on the purpose of your application, you should choose for your own strategy. For example, if my application display content in various kind of view (ListView and GridView), I will choose to separate TabGroup to many different pages. This method allow you to cache data in each page but it is less flexible

```
TabGroup {
        id: tabGroup
        anchors {
            left: parent.left;
            right: parent.right;
            top: upperRec.bottom;
            bottom: parent.bottom
        }

        Page {
            id: tab1content
            Component {
                id: tab1Delegate
                Item {
                    anchors {
                        left: parent.left;   leftMargin: 10
                        right: parent.right;   rightMargin: 10
                    }
                    height: contentCol.height + 20

                    Column {
                        spacing: 5
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    id: contentCol
                    width: parent.width;
                    Text {
                        text: title //...dataModel's key
                        width: parent.width
                        color: "#005fff"
                        wrapMode: Text.WordWrap
                        font.pixelSize: 22
                        font.bold: true
                    }

                    Text {
                        text: maintext //...dataModel's key
                        width: parent.width
                        color: "white"
                        wrapMode: Text.WordWrap
                        font.pixelSize: 19
                    }

                    Text {
                        text: linkurl //...dataModel's key
                        width: parent.width
                        color: "#87ff87"
                        wrapMode: Text.WordWrap
                        font.pixelSize: 16
                    }
                }

            }
        }

        ListView {
            id: tab1List

            anchors.fill: parent
            delegate:  tab1Delegate
            cacheBuffer: 8000

            model: tab1Data //... modelData here

        }

        ScrollDecorator {
            flickableItem: tab1List
        }
    }

    Page {
        id: tab2content
        Component {
            id: tab2Delegate
            Item {
                id: wrapper
                width: tab2List.cellHeight
                height: tab2List.cellHeight
                Item {
                    anchors.fill: parent
```


(C) Copyright Nokia 2014. All rights reserved.

```
                        anchors.centerIn: parent
                        scale: 0.0
                        Behavior on scale { NumberAnimation { easing.type:
Easing.InOutQuad} }
                        id: scaleMe

                        Item {
                            width: parent.width
                            height: parent.height
                            anchors.centerIn: parent
                            Item {
                                id: whiteRect
                                width: parent.width
                                height: parent.height - 70
                                smooth: true

                                Image {
                                    id: thumb
                                    width: parent.width - 70
                                    height: parent.height
                                    source: imgurl //...dataModel's key
                                    smooth: true
                                    fillMode: Image.PreserveAspectFit
                                    anchors.horizontalCenter: parent.horizontalCenter
                                }
                                Text {
                                    text: maintext + "<br/>" + linkurl + "<br/>" +
imgsize //...dataModel's key
                                    width: parent.width - 20
                                    color: "#87ff87"
                                    wrapMode: Text.WordWrap
                                    font.pixelSize: 14
                                    anchors.top: thumb.bottom
                                    verticalAlignment: Text.AlignVCenter
                                    horizontalAlignment: Text.AlignHCenter
                                }
                            }
                        }
                        states: [
                            State {
                                name: "Show"
                                when: thumb.status == Image.Ready
                                PropertyChanges { target: scaleMe; scale: 1 }
                            }
                        ]
                    }
                }
            }

            GridView {

                id: tab2List
                anchors.fill: parent
                anchors {
                    left: parent.left;  leftMargin: 10
                    right: parent.right;  rightMargin: 10
```


(C) Copyright Nokia 2014. All rights reserved.

```
                }
                delegate:  tab2Delegate
                cacheBuffer: 8000
                model: tab2Data //... modelData here

                cellWidth: (parent.width-20)/2
                cellHeight: cellWidth
            }

            ScrollDecorator {
                flickableItem: tab2List
            }
        }
}
```

Let's make everything be simple and flexible.

```
Component {
        id: tabNewsDelegate

        Item {
            anchors {
                left: parent.left; leftMargin: 10
                right: parent.right;  rightMargin: 10
            }
            height: contentCol.height + 20

            Column {
                spacing: 5
                id: contentCol
                width: parent.width;
                Text {
                    id: newsTitle
                    text: title //...dataModel's key
                    width: parent.width
                    color: "#005fff"
                    wrapMode: Text.WordWrap
                    font.pixelSize: 22
                    font.bold: true
                }

                Text {
                    id: newsMainText
                    text: maintext //...dataModel's key
                    width: parent.width
                    color: "white"
                    wrapMode: Text.WordWrap
                    textFormat: Text.RichText
                    font.pixelSize: 18
                }

                Text {
                    text: source + " - " + time //...dataModel's key
                    width: parent.width
                    color: "#87ff87"
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    wrapMode: Text.WordWrap
                    font.pixelSize: 16
                }

                Button {
                    anchors.right: parent.right
                    width: parent.width / 3
                    text: "Read it"
                    onClicked: {
                        //...
                    }
                }
            }

        }
    }
```

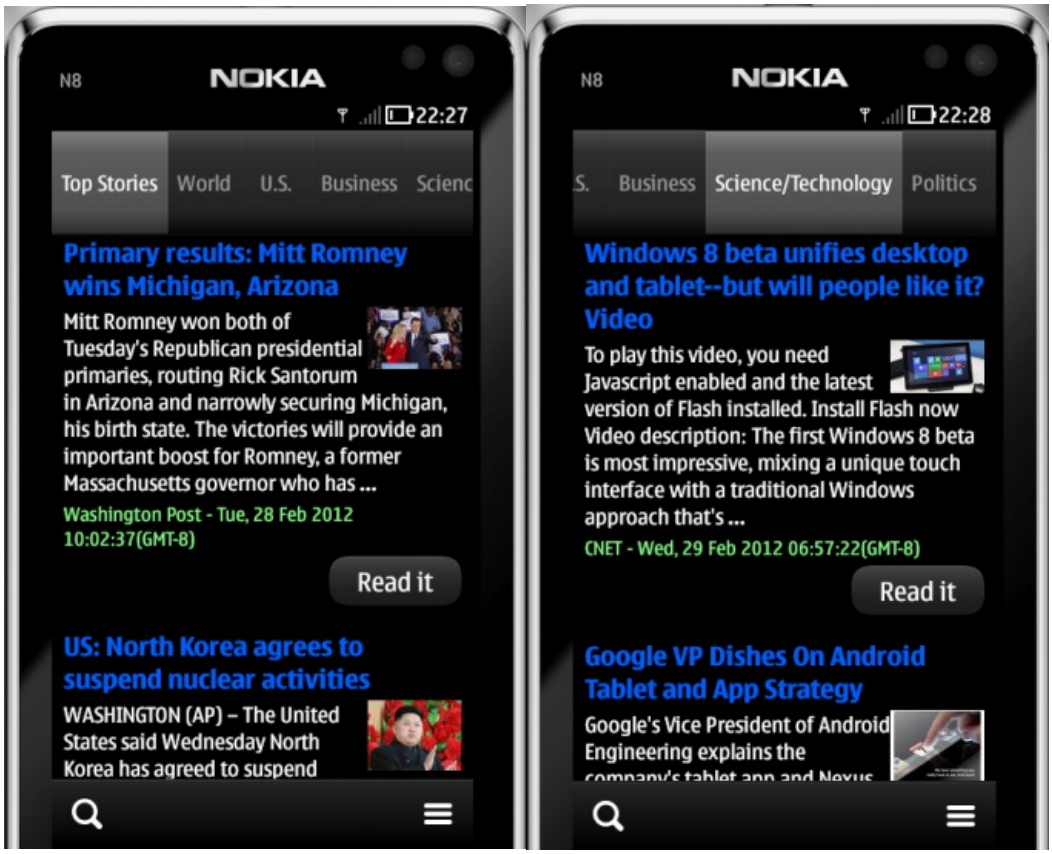
In this code, I set up a delegate component and only one page in TabGroup.

```
TabGroup {
        id: tabGroup
        anchors {
            left: parent.left;
            right: parent.right;
            top: upperRec.bottom;
            bottom: parent.bottom
        }
        Page {
            id: tabcontent
            ListView {
                id: tabList
                anchors.fill: parent
                delegate:  tabNewsDelegate
                cacheBuffer: 18000

               //...Here is the key point of our flexibility model
              //Just set up modelData based on the TabButton's index
              //topicsList in this example is same with searchModel in the previous
example
                model: {
                    if(topicsList.currentIndex == 0){
                        return topstoriesData
                    }else if(topicsList.currentIndex == 1){
                        return worldData
                    }//...more data here
                }
            }
            ScrollDecorator {
                flickableItem: tabList
            }
        }
}
```

Here is what I'll have after put all together

(C) Copyright Nokia 2014. All rights reserved.



App's download

http://store.ovi.com/content/257579

Note: This is an entry in the Symbian Qt Quick Components Competition 2012Q1

(C) Copyright Nokia 2014. All rights reserved.