How to calculate hash for a text using QCryptographicHash in Qt

This code snippet shows how to calculate hash for a text string using QCryptographicHash.

Source Code

Main.cpp

```
#include <QtGui/QApplication>
#include "qmlapplicationviewer.h"
#include <QCryptographicHash>
#include "QDeclarativeContext"
#include "hashcalculate.h"

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    HashCalculate*ihashcalc =  new HashCalculate();
    QmlApplicationViewer viewer;
    viewer.setOrientation(QmlApplicationViewer::ScreenOrientationAuto);
    QDeclarativeContext *ctxt = viewer.rootContext();
    ctxt->setContextProperty("myContextProperty",ihashcalc);
    viewer.setMainQmlFile(QLatin1String("qml/MD4HashSample/main.qml"));
    viewer.showExpanded();
    return app.exec();
}
```

hashcalculate.cpp

```
#include "hashcalculate.h"
#include <QCryptographicHash>
#include "QDebug"
HashCalculate::HashCalculate(QObject *parent) :
    QObject(parent)
{
    iHashValue = "";
}

void HashCalculate::calculateHash(const QString& aOriginalText )
{
 QCryptographicHash hash(QCryptographicHash::Md4);
 hash.addData(aOriginalText.toUtf8());
 SetHash(QString(hash.result().toHex()));
 //qDebug() << QString(hash.result().toHex());
}

void HashCalculate::SetHash(const QString& aHashValue)
{
 iHashValue = aHashValue;
}

QString HashCalculate::getHash()
{
 return iHashValue;
}
```

hashcalculate.h

```
#ifndef HASHCALCULATE_H
#define HASHCALCULATE_H

#include <QObject>

class HashCalculate : public QObject
{
    Q_OBJECT
public:
    explicit HashCalculate(QObject *parent = 0);
    Q_INVOKABLE void calculateHash(const QString& aOriginalText );

signals:

public slots:
        QString getHash();
        void SetHash(const QString& aHashValue);
public:
        QString iHashValue;

};

#endif // HASHCALCULATE_H
```

Main.qml

```
import QtQuick 1.0
import com.nokia.symbian 1.0

Window {
```


(C) Copyright Nokia 2013. All rights reserved.

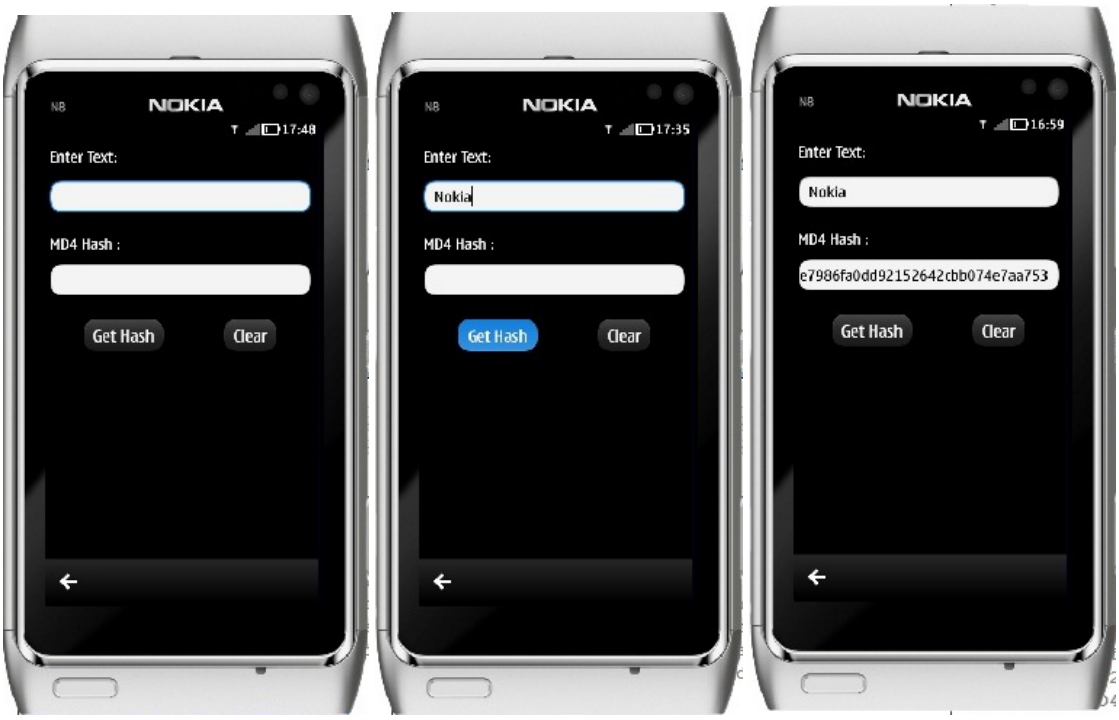
```
    id: window

    StatusBar {
        id: statusBar
        anchors.top: window.top
    }

    PageStack {
        id: pageStack
        anchors { left: parent.left; right: parent.right; top: statusBar.bottom; bottom: toolBar.top }
    }

    ToolBar {
        id: toolBar
        anchors.bottom: window.bottom
        tools: ToolBarLayout {
            id: toolBarLayout
            ToolButton {
                flat: true
                iconSource: "toolbar-back"
                onClicked: pageStack.depth <= 1 ? Qt.quit() : pageStack.pop()
            }
        }
    }

    Component.onCompleted: {
        pageStack.push(Qt.resolvedUrl("MainPage.qml"))
    }
}
```

MainPage.qml

```
import QtQuick 1.0
import com.nokia.symbian 1.0

Page {
    id: mainPage

    Text {
        text: qsTr("Enter Text:")
        id:textorg
        color: platformStyle.colorNormalLight
        font.pixelSize: 20
        x:text_originaltxt.x
        y:text_originaltxt.y-45
    }

    Text {
        text: qsTr("MD4 Hash :")
        id: txthshd
        color: platformStyle.colorNormalLight
        font.pixelSize: 20
        x:text_hashedtxt.x
        y:text_hashedtxt.y-40

    }

    TextField {
        id: text_originaltxt
        x: parent.x+5
        y:parent.y+55
        width: parent.width-15
        text : "Nokia"

    }
    TextField {
        id: text_hashedtxt
        x: text_originaltxt.x
        y:text_originaltxt.y+110
        width: parent.width-15
        readOnly: true
        text : ""

    }

    Button {
            id: btnhash
            x:parent.x+50
            y:text_hashedtxt.y+text_hashedtxt.height+txthshd.height+5
            text: "Get Hash"
            onClicked: {
                myContextProperty.calculateHash(text_originaltxt.text)
                text_hashedtxt.text = myContextProperty.getHash()
            }
         }
    Button {
            id: btnclr
            x:parent.x+btnhash.width+130
            y:text_hashedtxt.y+text_hashedtxt.height+txthshd.height+5
            text: "Clear"
            onClicked: {
                text_originaltxt.text = ""
                text_hashedtxt.text = ""
            }
         }

}
```

ScreenShots

(C) Copyright Nokia 2013. All rights reserved.



About the Update

The implementation has now been converted using the Qt Quick components.As compared with the previous functionality in which a hard-coded text was getting converted into a hash, now the user has flexibility to enter any text & convert it into a hash code as visible in the screenshots.

Source Code

Full source code can be downloaded from here : Media:MD4HashSample.zip

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.