Capturing image using QML Camera and uploading to Facebook

This article demonstrates how to capture an image using the QML Camera Element and share it to Facebook.

Note: This is an entry in the PureView Imaging Competition 2012Q2

Introduction

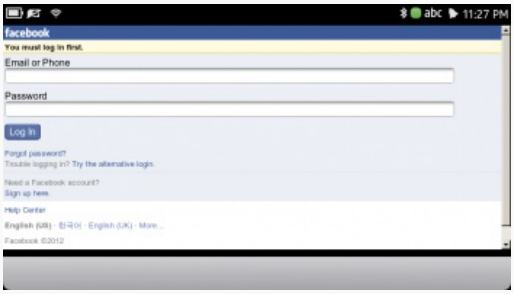
This article explains how to use QML Camera element to capture an image and then how to upload and share captured image to Facebook. In more detail, it shows how to:

1. Use QML Camera element to capture an image
2. Login to facebook using QNetworkAccessManager
3. Use HTTP Post method to upload capture image to facebook album

Following are snaps from my sample application running on N9.

Take picture

Picture



Facebook login

Picture uploaded

Using the QML Camera Element

QML Camera is part of Qt Mobility API, so you need to configure Qt's Project file to to include Qt Mobility package. We also need to use Network, Webkit and Declarative Module. For Symbian we also need to specify capabilities to be able to connect to network and access the camera.

```
QT += network webkit declarative

CONFIG += mobility
MOBILITY = multimedia

symbian {
       TARGET.CAPABILITY += NetworkServices \
         ReadUserData \
         WriteUserData \
         LocalServices \
         UserEnvironment
}
```

Below is QML code for **main.qml** which I launch from **main.cpp** file. The codes uses QML Camera to display camera view and capture image with it. Once image is captured the Camera element is hidden and the preview view is displayed.

`PreviewImage` view shows captured image and it offers the option to share it on Facebook. To avoid QML component dependency I have create a simple custom button element that will work on all Qt platforms.

(C) Copyright Nokia 2013. All rights reserved.

```
import QtQuick 1.1
import QtMultimediaKit 1.1

Rectangle {
    width:640;height:360
    Camera {
        id:camera
        width:parent.width;height:parent.height
        focus : visible // to receive focus and capture key events when visible
        opacity: 1

        flashMode: Camera.FlashRedEyeReduction
        whiteBalanceMode: Camera.WhiteBalanceFlash
        exposureCompensation: -1.0

        onImageCaptured : {
            console.debug("Captured Image:" +preview);
            preview.opacity=1;
            camera.opacity=0;
            preview.source = preview;
        }
        onImageSaved: {
            console.log("Image saved:"+path);
            preview.opacity=1;
            camera.opacity=0;
        }

        MyButton{
            id:captureBtn
            anchors.verticalCenter: parent.verticalCenter
            anchors.right:parent.right
            text:"Capture"

            onClicked: {
                camera.captureImage();
            }
        }
    }

    PreviewImage{
        id:preview
        width:parent.width;height:parent.height
        opacity: 0
        camera:camera

        onHide: {
            preview.opacity=0;
            camera.opacity=1;
        }
    }
}
```

Following is PreviewImage.qml file. PreviewImage view is enabled when Image is captured, it show captured image and by pressing share button it upload the captured image on Facebook account.

```
import QtQuick 1.1
import QtMultimediaKit 1.1

Item {
    id:preview

    property alias source:previewImage.source
    property Camera camera;
    property variant currentImage;

    signal hide();

    Image {
        width:parent.width;height:parent.height
        id:previewImage
    }

    MyButton{
        id:shareBtn
        anchors.verticalCenter: parent.verticalCenter
        anchors.right:parent.right
        text:"Share"
        enabled: false

        onClicked: {
            console.log("share image"+currentImage);
            if( FbHelper.isAuthorized() ) {
                FbHelper.uploadPicture(currentImage,"Testing");
            } else {
                FbHelper.login();
            }
        }
    }

    Connections{
        target: camera
        onImageSaved: {
            currentImage = path;
            preview.source="file://"+path
            shareBtn.enabled=true
        }
    }
}
```

The image is now captured. The following code allows the user to login to their Facebook account and upload a photo.

Facebook helper - log in to facebook and upload photos

I created a blog here some time ago, explaining how to login to facebook and how to post comment on wall. I will use same code here to login to facebook account using `QNetworkAccessManager`.

First of all to login on facebook account and posting on wall or uploading photo, we need to request an application ID from Facebook here. With this id you

 (C) Copyright Nokia 2013. All rights reserved.

can request an authorisation token from the Facebook API when you login, and use it to upload photos.

Login method will launch facebook login webpage in QWebView. Once a user enters a valid username and password, it will redirects to https://www.facebook.com/connect/login_success.html url with access token mentioned in query string.

```
void FacebookHelper::login( const QString& scope ){
    QUrl url("https://www.facebook.com/dialog/oauth");
    url.addQueryItem("client_id",YOUR_APPP_ID);
    url.addQueryItem("redirect_uri",
        "https://www.facebook.com/connect/login_success.html");
    url.addQueryItem("response_type","token");
    url.addQueryItem("scope","read_stream,publish_stream");

    //view = new QWebView();
    view->load( url );
    view->show();
    connect(view,SIGNAL(loadFinished(bool)),this,SLOT(loginResponse(bool)));
}

void FacebookHelper::loginResponse( bool status ){
    QUrl url= view->url();
    QString strUrl = url.toString();

    int sIndex = strUrl.indexOf("access_token=");
    int eIndex = strUrl.indexOf("&expires_in");
    if( sIndex != -1 && eIndex != -1 ){
        mAccessToken= strUrl.mid(sIndex, eIndex - sIndex);
        mAccessToken = mAccessToken.remove("access_token=");
        emit authStatus( mAccessToken);
    }
}
```

Now we have access token, we are ready to upload photo to facebook. Following code describe the process. uploadPicture method takes absolute path to picture to upload and comment for photo, then it creates form-data request and send that to facebook using HTTP Post method.

```
void FacebookHelper::uploadPicture(const QString& picLocation, const QString& comment) {

    QString uploadUrl = "https://graph.facebook.com/me/photos?access_token="+ mAccessToken;

    QFileInfo fileInfo(picLocation);
    QFile file(picLocation);
    if (!file.open(QIODevice::ReadWrite)) {
        qDebug() << "Can not open file:" << picLocation;
        return;
    }

    QString bound="---------------------------723690991551375881941828858";
    QByteArray data(QString("--"+bound+"\r\n").toAscii());
    data += "Content-Disposition: form-data; name=\"action\"\r\n\r\n";
    data += "\r\n";
    data += QString("--" + bound + "\r\n").toAscii();
    data += "Content-Disposition: form-data; name=\"source\"; filename=\""+file.fileName()+"\"\r\n";
    data += "Content-Type: image/"+fileInfo.suffix().toLower()+"\r\n\r\n";
    data += file.readAll();
    data += "\r\n";
    data += QString("--" + bound + "\r\n").toAscii();
    data += QString("--" + bound + "\r\n").toAscii();
    data += "Content-Disposition: form-data; name=\"message\"\r\n\r\n";
    data += comment.toAscii();
    data += "\r\n";
    data += "\r\n";

    QNetworkRequest request(uploadUrl);
    request.setRawHeader(QByteArray("Content-Type"),QString("multipart/form-data; boundary=" + bound).toAscii());
    request.setRawHeader(QByteArray("Content-Length"), QString::number(data.length()).toAscii());
    mCurrentRequest = mNetManager.post(request,data);
    connect(mCurrentRequest,SIGNAL(finished()),this,SLOT(messageResponse()));
}
```

Once we received finished signal for this request, Photo should appear on user's account.

Downloads

The project source is here: Media:FacebookHelper.zip. In order to test this you will need to remember to put your Facebook app id, in **main.cpp** file.

Summary

By using QML Camera and `QNetwokAccessManager` we can easily capture and share image to facebook network.

Hope you enjoy the article.

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.