Redirecting JavaScript console messages in a Qt hybrid application

Overview

One of the first things that you encounter when creating a hybrid application on top of `QtWebKit` is lack of **JavaScript** debug information. While it is easy to use Web Inspector to debug applications, on some occasions it is enough to get debug messages from the hybrid application for later inspection. This article shows how to redirect JavaScript console messages into a file or your own message handler.

Redirecting JavaScript console messages can be easily done by sub classing QWebPage and overriding virtual function called `javaScriptConsoleMessage()`. Writing to file can be omitted, if you have already implemented your own message handler that redirects qDebug –messages into a file

mypage.h

```
#ifndef MYPAGE_H
#define MYPAGE_H
#include <QObject>
#include <QWebPage>

class MyPage : public QWebPage
{
    Q_OBJECT
public:
    MyPage(QObject * parent = 0 );
protected:
    void javaScriptConsoleMessage(const QString& message, int lineNumber, const QString& sourceID);
};
#endif // MYPAGE_H
```

mypage.cpp

```
#include "mypage.h"

#include <QDebug>
#include <QFile>00
#include <QTextStream>

MyPage::MyPage(QObject *parent): QWebPage(parent){
}

void MyPage::javaScriptConsoleMessage(const QString& message, int lineNumber, const QString& sourceID){
   QString logEntry = message +" on line:"+ QString::number(lineNumber) +" Source:"+ sourceID;
   qDebug()<<logEntry;

   /** can be omitted */
   QFile file("C:\\Data\\debug.txt");
   if (!file.open(QIODevice::WriteOnly | QIODevice::Append | QIODevice::Text))
       return;
   QTextStream out(&file);
   out << logEntry  << endl;
}
```

After you have finished creating your own subclass of `QWebPage`, you still need to set it as active page on your `QWebView`.

```
MyPage page = new MyPage();
this->webView->setPage(page);
```


(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.