



[主页](#) > [博客](#) >

Qt webKit可以做什么(四)——实现本地QObject和JavaScript交互

作者: [Dawei Cheng 程大伟 \(Intel\)](#) (5 篇文章) 日期: 六月 9, 2010 在 12:21 下午

上一篇我们了解了如何在webkit中创建含有web内容的本地应用。这一篇我们将实现JavaScript和本地的QObject的交互。在阅读本篇之前需要对Qt的信号和槽机制和JavaScript有简单的了解。
Qt本地对象和JavaScript交互分为三个步骤

1.
 1. 将本地QObject暴露给webkit和JavaScript
 2. 将本地QObject的信号和JavaScript的槽连接起来
 3. 通过JavaScript调用本地QObject的槽



也就是第1步和第2步结合起来实现 本地QObject的信号和JavaScript的槽连接
第1步和第3步结合起来实现 通过JavaScript调用本地QObject的槽

下面我们就分别看一下第1、2、3步分别如何实现的。

1、将本地QObject暴露给Webkit。主要分为以下几个步骤。

1.
 1. 新建一个QObject，命名为simpleQObject，包含信号和槽。其头文件如下：
 1. #include <QtCore/QObject>
 2. #include <QtCore/QMap>
 3. #include <QtCore/QString>
 4. #include <QtCore/QVariant>
 5. class SampleQObject : public QObject
 6. {
 7. Q_OBJECT
 8. public:
 9. SampleQObject(QObject *parent = 0);
 10. signals: /* 声明QObject signals */
 11. void signal(QMap<QString, QVariant> object);
 12. public slots: /* 声明 QObject slots */
 13. QMap<QString, QVariant> slotThatReturns(const QMap<QString,
 14. QVariant> & object); //QObject 的槽，用来返回字符串
 15. void slotThatEmitsSignal(); //QObject 的槽，用来发射信号，并且记录发射次数
 16. private:
 17. int m_signalEmited;
 18. QMap<QString, QVariant> m_returnObject;
 19. QMap<QString, QVariant> m_emitSignal;
 20. };
 2. 创建实现函数。mainwindow.h 和 mainwindow.cpp。mainwindow.h代码如下
 1.
 1. #include <QMainWindow>
 2. #include "sampleqobject.h"
 3. class MainWindow : public QMainWindow {
 4. Q_OBJECT
 5. public:
 6. MainWindow(QWidget *parent = 0);
 7. ~MainWindow();
 8. private slots:
 9. /*声明 将 QObject 暴露给 Qt WebKit 的函数*/
 10. void addJavaScriptObject();
 11. protected:
 12. void changeEvent(QEvent *e);
 13. private:
 14. Ui::MainWindow *ui;
 15. SampleQObject* m_sampleQObject;
 16. };
 3. mainwindow.cpp关键代码如下：
 1.
 1. //当网页被载入或者刷新时，将暴露给webkit的QObject和webkit JavaScript连接
 2. connect(ui->webView->page()->mainFrame(),
 3. SIGNAL(javascriptWindowObjectCleared()),

```
4. this, SLOT(addJavaScriptObject()));
```

```
1.
```

```
1.
```

```
1.
```

```
1.
```

```
1. void MainWindow::addJavaScriptObject()
2. {
3. //addJavaScriptObject函数的实现：将SimpleQObject和webkit.Javascript连接
4. this->ui->webView->page()->mainFrame()->addToJavaScriptWindowObject
5. ("sampleQObject", this->m_sampleQObject);
6. }
```

2. 将本地QObject的信号和JavaScript的连接线起来



```
1.
```

```
1. 如何发射QObject信号。
```

```
1.
```

```
1.
```

```
1. signals:
2. void signal(QMap<QString, QVariant> object);
3. public slots:
4. void slotThatEmitsSignal();
5. /* this slot is designed to emit signals and count emit times */
6. void SampleQObject::slotThatEmitsSignal()
7. {
8. qDebug() << "SampleQObject::slotThatEmitsSignal";
9. this->m_signalEmitted++; /* count emit times */
10. this->m_emitSignal.clear();
11. this->m_emitSignal["signalsEmitted"] = QVariant(this->m_signalEmitted);
12. this->m_emitSignal["sender"] = QVariant("SampleQObject::slotThatEmitsSignal");
13. qDebug() << "SampleQObject::slotThatEmitsSignal" << this->m_emitSignal;
14. /* 发射信号 */
15. emit signal(this->m_emitSignal);
16. }
```

2. JavaScript槽的实现

```
1.
```

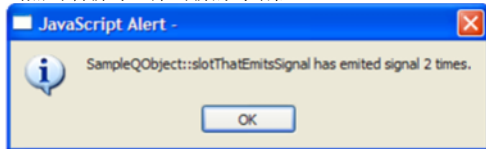
```
1.
```

```
1. $(document).ready(function() {
2. try {
3. /* 将sampleQObjects的信号和JavaScript slot连接起来 */
4. sampleQObject.signal.connect(slot);
5. /* 当sampleQObject发射信号时调用JavaScript的槽 */
6. sampleQObject.slotThatEmitsSignal();
7. }
8. catch(e) {
9. alert(e);
10. }
11. });
12. /* slot函数将会输出 SimpleQObject has emitted signal ? times */
13. function slot(object) {
14. var objectString = object.sender +
15. " has emitted signal " +
16. object.signalsEmitted +
17. " times.";
18. alert(objectString);
19. }
```

3. Run the app.



当点击刷新时，弹出新的对话框：



3. 通过JavaScript调用本地QObject的槽



1.

1. JavaScript信号发射

1.

1.

```
1. try {
2.   var object = {intValue: 1};
3.   /* 声明一个JavaScript object并用sampleQObject的槽的返回值赋值它*/
4.   var returnedObject = sampleQObject.slotThatReturns(object);
5.   /* 输出 "1 added bonus"*/
6.   alert(returnedObject.stringValue);
7. }
```

2. QObject 槽函数

1.

1.

```
1. QMap<QString, QVariant> SampleQObject::slotThatReturns(const QMap<QString, QVariant>& object)
2. {
3.   qDebug() << "SampleQObject::slotThatReturns";
4.   this->m_returnObject.clear();
5.   this->m_returnObject.unite(object);
6.   QString addedBonus = QString::number(object["intValue"].toInt(),
7.   10).append(" added bonus.");
8.   this->m_returnObject["stringValue"] = QVariant(addedBonus);
9.   qDebug() << "SampleQObject::slotThatReturns" << this->m_returnObject;
10.  return this->m_returnObject; // 返回m_returnObject包含有字符串 1 added bonus
11. }
```

3. Run the app



本例子代码下载地址：<http://software.intel.com/file/28111> <http://software.intel.com/file/28112>

注：本例子中部分代码来自wiki.forum.nokia.com

http://wiki.forum.nokia.com/index.php/Exposing_QObjects_to_Qt_Webkit

下一篇我们将探索一下如何将S60下的web runtime widget porting至Qt的webkit上来，如今的WRT widget只能依赖于S60的os，如果porting只QT的webkit，那么将可以实现广泛开发者梦寐以求的跨平台功能。

分类：其他, 移动技术, 英特尔® 软件网络 2.0

评论 (6)

2010年06月24日 21:41



文帅

讲得很详细，很不错的

2010年07月06日 01:31



wokn

你的demo下下来之后不能用 麻烦再传一个 谢谢

2010年07月06日 20:18

[Dawei Cheng 程大伟 \(Intel\)](#)



总分:
485
绿带

你在什么平台下用的？代码我在Qt creator windows SDK环境下是可以用的。

2010年07月19日 07:00



Neil

我把附件下载下来之后解压失败了，是不是压缩文件损坏了？可不可以重新传一个？或者直接发到我的邮箱Neil.king.liu@gmail.com，谢谢了。

2010年08月08日 08:14



lnxprop

使用winzip解压，不要使用winrar

2010年10月21日 22:51



Eddie

请问这例子的网页可以放到web server(如IIS)上执行吗??
我试的似乎不行，是不是缺少了什么元件

引用 (0)

写评论

欲获得技术支持，请访问[软件支持页面](#).
姓名 (必填)*

电子邮件 (必填，不在本页面显示)*

您的 URL (可选)

评论*

提交

*法律信息

©
英
特
尔
公
司