

ID		Creation date	June 8, 2009
Platform	Qt	Tested on devices	5800 XpressMusic
Category	Qt for Symbian	Subcategory	

Keywords (APIs, classes, methods, functions): Qt, XML, SAX, QXmlStreamReader

Overview

This example shows you how you parse XML with QXmlStreamReader. The QXmlStreamReader class provides a fast parser for reading well-formed XML via a simple streaming API.

This snippet can be self-signed. As it does not use any API which require developer/certified signing.

Preconditions

- Download and Install latest version [Qt for Symbian - Installation packages](#) which has links on how to install the latest version

QXmlStreamReader Example

XML file

```
<?xml version="1.0" encoding="UTF-8" ?>
<persons>
<person>
<firstname>John</firstname>
<surname>Doe</surname>
<email>john.doe@example.com</email>
<website>http://en.wikipedia.org/wiki/John_Doe</website>
</person>
<person>
<firstname>Jane</firstname>
<surname>Doe</surname>
<email>jane.doe@example.com</email>
<website>http://en.wikipedia.org/wiki/John_Doe</website>
</person>
<person>
<firstname>Matti</firstname>
<surname>Meikäläinen</surname>
<email>matti.meikalainen@example.com</email>
<website>http://fi.wikipedia.org/wiki/Matti_Meikäläinen</website>
</person>
</persons>
```

Header file

```
#include <QtXml/QXmlStreamReader>

class QXSRExample : public QMainWindow {
    Q_OBJECT

public:
    QXSRExample(QWidget *parent = 0);
    ~QXSRExample() {}

private slots:
    void parseXML();

private:
    QVBoxLayout _layout;

    void setupUI();

    QMap<QMap, QString> parsePerson(QXmlStreamReader& xml);
    void addElementDataToMap(QXmlStreamReader& xml,
        QMap<QMap, QString>& map) const;

    void addPersonsToUI(QList< QMap<QString,QString> >& persons);
};
```

Source file

```
void QXSRExample::parseXML() {
    /* We'll parse the example.xml */
    QFile file(":/example.xml");
    /* If we can't open it, let's show an error message. */
    if (!file->open(QIODevice::ReadOnly | QIODevice::Text)) {
        QMessageBox::critical(this,
            "QXSRExample::parseXML",
            "Couldn't open example.xml",
            QMessageBox::Ok);
    }
    return;
}

/* QXmlStreamReader takes any QIODevice. */
QMap<QMap, QString> QXSRExample::parseXML(QXmlStreamReader& xml) {
    QMap<QMap, QString> persons;
    /* We'll parse the XML until we reach end of it.*/
    while(!xml.atEnd() &&
        !xml.hasError()) {
        /* Read next element.*/
        QXmlStreamReader token = xml.readNext();
        /* If token is just StartDocument, we'll go to next.*/
        if(token == QXmlStreamReader::StartDocument) {
            continue;
        }
        /* If token is StartElement, we'll see if we can read it.*/
        if(token == QXmlStreamReader::StartElement) {
            /* If it's named persons, we'll go to the next.*/
            if(xml.name() == "persons") {
                continue;
            }
        }
        /* If it's named person, we'll dig the information from there.*/
        if(xml.name() == "person") {
```

Using_QXmlStreamReader_to_parse_XML

```

        append(this->parsePerson(xml));
    }
}
}
/* Error handling. */
if(xml.hasError()) {
    QMessageBox::critical(this,
        "QXSRExample::parseXML",
        xml.errorString(),
        QMessageBox::Ok);
}

clearxml();
->addPersonsToUI(persons);
}

 QMap<QString, QString> QXSRExample::parsePerson(QXmlStreamReader& xml) {
    QMap<QString, QString> person;
    /* Let's check that we're really getting a person. */
    if(xml.tokenType() != QXmlStreamReader::StartElement &&
        xml.name() != "person") {
        return person;
    }
    /* Next element... */
    xml.readNext();
    /*
     * We're going to loop over the things because the order might change.
     * We'll continue the loop until we hit an EndElement named person.
     */
    while(!(xml.tokenType() == QXmlStreamReader::EndElement &&
        xml.name() == "person")) {
        if(xml.tokenType() == QXmlStreamReader::StartElement) {
            /* We've found first name. */
            if(xml.name() == "firstname") {
                ->addElementDataToMap(xml, person);
            }
            /* We've found surname. */
            if(xml.name() == "surname") {
                ->addElementDataToMap(xml, person);
            }
            /* We've found email. */
            if(xml.name() == "email") {
                ->addElementDataToMap(xml, person);
            }
            /* We've found website. */
            if(xml.name() == "website") {
                ->addElementDataToMap(xml, person);
            }
        }
        /* ...and next... */
        xml.readNext();
    }
    return person;
}

void QXSRExample::addElementDataToMap(QXmlStreamReader& xml,
    QMap<QString, QString>& map) const {
    /* We need a start element, like <foo> */
    if(xml.tokenType() != QXmlStreamReader::StartElement) {
        return;
    }
    /* Let's read the name... */
    QString elementName(xml.name().toString());

```

Using_QXmlStreamReader_to_parse_XML

```
/* ...go to the next. */
readNext();
/*
 * This elements needs to contain Characters so we know it's
 * actually data, if it's not we'll leave.
 */
if(xml.tokenType() != QXmlStreamReader::Characters) {
return;
}
/* Now we can add it to the map.*/
insert(elementName, xml.text().toString());
}

void QXSRExample::addPersonsToUI(QList< QMap<QString,QString> >& persons) {
while(!persons.isEmpty()) {
 * personGroupBox QGroupBox("Person");
 * layFormLayout QFormLayout;
<QString,QString> person = persons.takeFirst();
->addRow("Firstname", new QLineEdit(person["firstname"]));
->addRow("Surname", new QLineEdit(person["surname"]));
->addRow("Email", new QLineEdit(person["email"]));
->addRow("Website", new QLineEdit(person["website"]));
->setLayout(layFormLayout);
->_layout->addWidget(personGB);
}
}
```

Postconditions

You've been able to parse the XML and display it in the UI.

Supplementary material

- You can test the QXSRExample with a test application. The application is available for download at [Media:QXSRExample.zip](#).

Parsing XML with attributes

Lets take example of parsing following XML file

```
<?xml version="1.0" encoding="UTF-8" ?>
<persons>
<person firstname="abc" surname="def" number="1234" />
<person firstname="ghi" surname="jkl" number="51234" />
<person firstname="mno" surname="pqr" number="61234" />
<person firstname="stv" surname="wxy" number="81234" />
<person firstname="zab" surname="cde" number="61234" />
</persons>
```

Header file

```
#include <QtXml/QXmlStreamReader>

class ContactDatabase
{
public:
    ContactDatabase(QString firstName, QString surName, QString phoneNumber);
    ~ContactDatabase(){}
    QString getFirstName() const;
    QString getSurName() const;
    QString getPhoneNumber() const;
private:
    QString firstName;
    QString surName;
    QString phoneNumber;
};

class QXSRExample : public QMainWindow {
    Q_OBJECT

public:
    QXSRExample(QWidget *parent = 0);
    ~QXSRExample(){}

private slots:
    void parseXML();

private:
    QVBoxLayout * _layout;

    void setupUI();
    void addPersonsToUI();
    void parsePerson(QXmlStreamReader& reader);
    ContactDatabase * database;
};
```

Source file

```
/* The implementaion of parseXML() is same as explained in above example, so omitted from here */

void QXSRExample::parsePerson(QXmlStreamReader& xml)
{
    /* parse each tag and add data to list
    *We'll continue the loop until we hit an EndElement named persons
    */
    while (!xml.atEnd())
    {
        /* check whether it is starting of tag <person> , if yes then parse it
        */
        if (xml.isStartElement() && xml.name() == "person")
        {
            /* get all attributes and its value of a tag in attrs variable.
            */
            QXmlStreamAttributes attrs(xml.attributes());
            /*get value of each attribute from QXmlStreamAttributes */
            QString firstName = attrs.attribute("firstName");
            QString surName = attrs.attribute("surName");
            QString phoneNumber = attrs.attribute("phoneNumber");
        }
    }
}
```

Using_QXmlStreamReader_to_parse_XML

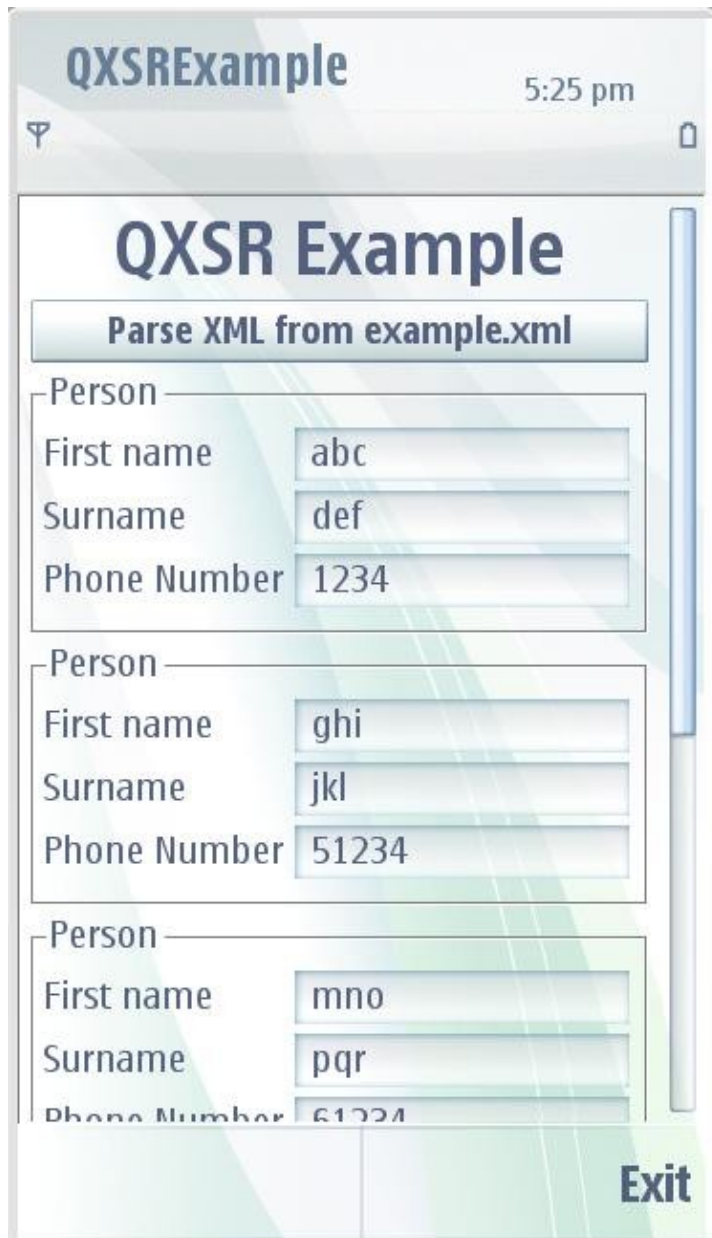
```
if (fName.isEmpty())
{
    readNext();          xml.
    continue;
}
/*create database of one person*/
    (fName.toString()+pName.toString(),pNumber.toString());
/*add database to list*/
    database.append(data);
}
else if(xml.isEndElement() && xml.name() == "persons")
{
    /*if EndElement is persons then stop the loop and create UI */
        ();          addPersonsToUI
return;
}
    readNext(); xml.
}
}

void QXSRExample::addPersonsToUI()
{
    /* add all contact data to UI */
    while(!database.isEmpty())
    {
        * personGB = new QGroupBox("Person");
        * layout = new QFormLayout;
        ContactDatabase * person = ContactDatabase::getFirst();
        ->addRow("Firstname", new QLineEdit(person.getFirstName()));
        ->addRow("Surname", new QLineEdit(person.getSurName()));
        ->addRow("PhoneNumber", new QLineEdit(person.getPhoneNumber()));

        ->setLayout(layout);
        ->_layout->addWidget(personGB);
    }
}
```

Postconditions

You've been able to parse the XML and display it in the UI.



Example Code

- Download [Code Example from here](#)

See also

- [QXmlStreamReader documentation](#)