Using QDomDocument to parse XML

Overview

24 Oct 2010

This code example shows you how to parse well-formed XML with QDomDocument. `QDomDocument` is easy and strait forward way on parsing well known XML documents.

Preconditions

- Download and install the Qt SDK

QDomDocument parsing Example

XML file

```
<?xml version="1.0" encoding="UTF-8" ?>
<persons>
    <person id="1">
        <firstname>John</firstname>
        <surname>Doe</surname>
        <email>john.doe@example.com</email>
        <website>http://en.wikipedia.org/wiki/John_Doe</website>
    </person>
    <person id="2">
        <firstname>Jane</firstname>
        <surname>Doe</surname>
        <email>jane.doe@example.com</email>
        <website>http://en.wikipedia.org/wiki/John_Doe</website>
    </person>
    <person id="3">
        <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

```
class QXSRExample : public QMainWindow
{
    Q_OBJECT

public:
    QXSRExample(QWidget *parent = 0);
    ~QXSRExample();
private slots:
    void parseXML();
private:
    QPointer<QVBoxLayout> _layout;
    void setupUI();
    void addPersonsToUI(QList< QMap<QString,QString> >& persons);
};
```

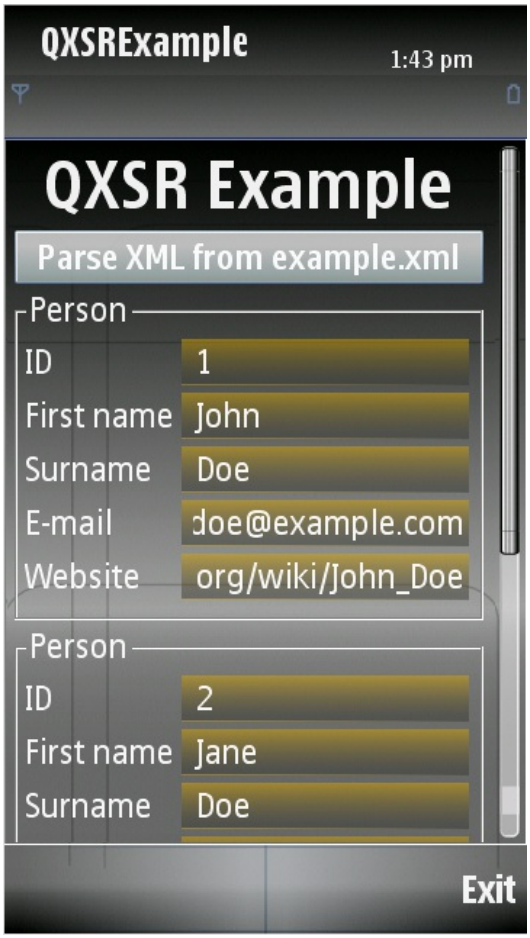
Source file

```
void QXSRExample::parseXML() {
    /* We'll parse the example.xml */
    QFile* file = new QFile("C://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;
    }

    QList< QMap<QString,QString> > persons;

    /* QDomDocument takes any QIODevice. as well as QString buffer*/
    QDomDocument doc("mydocument");
    if (!doc.setContent(file))
    {
     return;
 }

    //Get the root element
    QDomElement docElem = doc.documentElement();

    // you could check the root tag name here if it matters
 QString rootTag = docElem.tagName(); // == persons

 // get the node's interested in, this time only caring about person's
 QDomNodeList nodeList = docElem.elementsByTagName("person");

 //Check each node one by one.
```


(C) Copyright Nokia 2013. All rights reserved.

```
  for(int ii = 0;ii < nodeList.count(); ii++)
     {
   QMap<QString, QString> person;

   // get the current one as QDomElement
      QDomElement el = nodeList.at(ii).toElement();

      person["id"] = el.attribute("id"); // get and set the attribute ID

      //get all data for the element, by looping through all child elements
   QDomNode pEntries = el.firstChild();
         while(!pEntries.isNull()) {
          QDomElement peData = pEntries.toElement();
          QString tagNam = peData.tagName();

          if(tagNam == "firstname") { /* We've found first name. */
     person["firstname"] = peData.text();
          }else if(tagNam == "surname") { /* We've found surname. */
     person["surname"] = peData.text();
          }else if(tagNam == "email") { /* We've found email. */
     person["email"] = peData.text();
          }else if(tagNam == "website") { /* We've found website. */
     person["website"] = peData.text();
          }

          pEntries = pEntries.nextSibling();
         }

         persons.append(person);
      }

     this->addPersonsToUI(persons);
 }
```

Postconditions



You are able to parse the XML and display it in the UI.

Supplementary material

- You can test `QDomDocument` with a test application. The application is available for download at File:QXMLDocExample.zip.
- there is also simple but full RSS example available at File:SimpleRSSExample.zip

See also

QXmlStreamReader to parse XML in Qt

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.