Using QSettings in QML with also json and XML support

This article explains how to use settings in QML

Introduction

This class is used as a context property in QML applications as an easy way to store and get user settings. It supports storing arrays. These can also retrieved in json or xml format.

Usage

Start copying `setting.cpp` and `setting.h` into your project source directory. In the `main.cpp` file include the .h:

```
#include "setting.h"
```

the register the context property

```
Settings settings(appName, fileName);
viewer.rootContext()->setContextProperty("settings", &settings);
```

fileName will be create as follow:

```
QDesktopServices::storageLocation( QDesktopServices::DataLocation ) + "/" + appName +
"/" + fileName;
```

In your QML sources you will now be able to use the property `settings` in ini format as explained below.

single settings

The ini format is mostly like as:

```
[main]
user=Marco
location=Italy
...
```

so to write a new value or overwrite the existing just put in your qml this

```
settings.setValue("main/user", "Marco");
```

to get a value:

```
var user;
user = settings.getValue("main/user");
```

if you want to have a default value if there is not yet a setting named "main/user", do:

```
var user;
user = settings.getValue("main/user", "Marco");
```

if "main/user" exist it will return that value, else will return "Marco".

(C) Copyright Nokia 2014. All rights reserved.

array settings

The ini format for arrays is mostly like as:

```
[main]
user/size=2
user/1/name=Marco
user/1/location=Italy
user/2/name=Marco2
user/2/location=Italy2
...
```

appendToArray

to make this in your qml just use:

```
settings.appendToArray( "main/user", { "name": "Marco" , "location": "Italy" });
settings.appendToArray( "main/user", { "name": "Marco2" , "location": "Italy2" });
```

removeArray

to remove the whole array:

```
settings.removeArray( "main/user");
```

removeArrayEntry

to remove a single entry of the array:

```
settings.removeArrayEntry( "main/user", 1);
```

getIndexOfValueInArray

this will get the first array index of "Marco2" inside the array "main/user" checking it in the "name" field.

```
var i = settings.getIndexOfValueInArray("main/user", "name", "Marco2");
```

so if it is needed to remove "Marco2" from the array, just do:

```
settings.removeArrayEntry( "main/user", settings.getIndexOfValueInArray("main/user",
"name", "Marco2") );
```

checkValueArray

this will return true or false whenever "Marco" is in the array "main/user"

```
var b = settings.checkValueArray("main/user", "name", "Marco");
```

getArrayJson

(C) Copyright Nokia 2014. All rights reserved.

this function will retrieve the array in a json string format. It is possible to use the eval() function for example to fill a ListModel:

```
var users = eval ( settings.getArrayJson("main/users") );
listModel.clear();
for (var i=0; i<users.length; i++)
        {
            listModel.append( {
                                        "name": users[i].name,
                                        "location": users[i].location
                                      } );
        }
```

getArray

this function is like getArrayJson but it returns the values in a QList< QVariantMap > ready to be use without the need to use eval() function.

getArrayXml

this function is useful to fill a XmlListModel. The returned xml format is like:

```
<xml>
   <item>
       <name>Marco</name>
       <location>Italy</location>
   </item>

   <item>
       <name>Marco2</name>
       <location>Italy2</location>
   </item>
...
</xml>
```

so, for example:

```
XmlListModel {
    id: xmlListModel
    xml: getArrayXml("main/users")
    query: "/xml/item"

     XmlRole { name: "name"; query: "name/string()" }
     XmlRole { name: "location"; query: "location/string()" }
}
```

will fill `xmlListModel` with the stored array.

Download

File:Settings-v1.0.zip

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.