



**Join GitHub today**

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Dismiss

Tree a11aff5f16 [quteqoin](#) / [core](#) / [totozmanager](#) / [qqtrequester.cpp](#)

[Find file](#) [Copy path](#)

 **dguihal** Avoid duplicate search results in totoz manager

0cd8732 Mar 30, 2017

1 contributor

98 lines (77 sloc) | 2.18 KB

```

1  #include "qqtrequester.h"
2
3  #include "core/qqsettings.h"
4  #include "ui/totozmanager/qqtxmlparser.h"
5
6  #include <QDebug>
7  #include <QNetworkReply>
8  #include <QXmlSimpleReader>
9  #include <QXmlInputSource>
10
11 QQTRequester::QQTRequester(QObject * parent) :
12     QNetworkAccessor(parent)
13 {
14     m_currKey.clear();
15     m_netReply = NULL;
16
17     m_xmlParser = new QQTXMLParser(this);
18     connect(m_xmlParser, SIGNAL(finished()), this, SLOT(parsingFinished()));
19 }
20
21 QQTRequester::~QQTRequester()
22 {
23     if(m_netReply != NULL)
24         m_netReply->abort();
25     delete m_xmlParser;
26 }
27
28 void QQTRequester::requestFinishedSlot(QNetworkReply * reply)
29 {
30     if(reply->error() == QNetworkReply::NoError)
31     {
32         QByteArray data = reply->readAll();
33
34         QXmlSimpleReader xmlReader;
35         QXmlInputSource xmlSource;
36
37         xmlSource.setData(data);
38         xmlReader.setContentHandler(m_xmlParser);
39         xmlReader.setErrorHandler(m_xmlParser);
40         xmlReader.parse(xmlSource);
41     }
42     else
43     {
44         qDebug() << Q_FUNC_INFO << "reply->error() =" << reply->error() << ", " << reply->errorString();
45         emit requestFinished();
46     }
47
48     m_netReply = NULL;
49     reply->deleteLater();
50 }
51
52 void QQTRequester::searchTotoz(const QString & key)
53 {
54     cancel();
55     m_totozes.clear();
56     m_currKey = key;
57     searchTotoz(key, 0);
58 }
59
60 void QQTRequester::cancel()
61 {
62     if(m_netReply != NULL)
63         m_netReply->abort();
64 }
65
66 void QQTRequester::searchTotoz(const QString & key, int offset)
67 {
68     QQSettings settings;
69
70     QString searchPattern = settings.value(SETTINGS_TOTOZ_SERVER_QUERY_PATTERN,
71     DEFAULT_TOTOZ_SERVER_QUERY_PATTERN).toString()
72     searchPattern.replace("%t", key.toLatin1().toPercentEncoding());
73     searchPattern.replace("%o", QString::number(offset));
74
75
76     QString urlString = settings.value(SETTINGS_TOTOZ_SERVER_URL, DEFAULT_TOTOZ_SERVER_URL).toString();
77     urlString.append("/").append(searchPattern);
78
79     QUrl url(urlString);
80
81     QNetworkRequest request(url);
82     request.setAttribute(QNetworkRequest::CacheLoadControlAttribute,
83     QNetworkRequest::PreferCache);
84
85     m_netReply = httpGet(request);
86 }
87
88 void QQTRequester::parsingFinished()
89 {
90     m_totozes.append(m_xmlParser->totozes());
91
92     if(m_xmlParser->numResults() > 0 &&
93         m_totozes.size() < m_xmlParser->numResults())
94         searchTotoz(m_currKey, m_totozes.size());
95     else
96         emit requestFinished();
97 }

```