

Способ хранения идентификатора записей в браузере

Набор записей в браузере (таблице `QTableWidget` или списке `QListWidget`) может отличаться от записей в контейнере. Например, в браузере отображается только часть записей или в контейнере записи хранятся в неупорядоченном виде, а в браузере они упорядочены и т.д.

В этом случае индекс записи в браузере не совпадает с индексом записи в контейнере и наоборот. Для связи записей в браузере с записями контейнера необходимо использовать уникальный идентификатор, который должен храниться и в браузере и в контейнере.

Для хранения уникального идентификатора в таблице можно использовать скрытую колонку. Однако в визуальном списке такое невозможно. Поэтому применим универсальный подход — поместим идентификатор записи в элемент браузера (`QListWidgetItem` или `QTableWidgetItem`) в качестве дополнительных данных.

Дело в том, что в каждом элементе браузера помимо отображаемого текста может храниться дополнительная информация, например, данные отображаемые в строке состояния при выборе элемента браузера, предполагаемые размеры и т.д.

Для задания этих данных используется метод элемента браузера:

```
void setData ( int role, const QVariant & value )
```

где **role** — роль или назначение данных; некоторые роли предопределены перечнем `Qt::ItemDataRole`:

Константа	Значение	Описание
<code>Qt::DisplayRole</code>	0	Значение, отображаемое в элементе браузера
...	...	...
<code>Qt::StatusTipRole</code>	4	Значение, отображаемое в строке состояния
...	...	...

<code>Qt::SizeHintRole</code>	13	Размеры
...	...	...
<code>Qt::UserRole</code>	32	Первая роль, определяемая пользователем

value — значение, соответствующее роли. Обратите внимание, что тип передаваемого значения — `QVariant`; это класс, который может хранить произвольные значения.

Таким образом, для хранения идентификатора можно воспользоваться ролью `Qt::UserRole`, назначение которой определяет пользователь.

Замечание. В таблице идентификатор можно хранить в любой ячейке строки, но лучше выбить первую.

Пример записи идентификатора в браузер:

```
unsigned int id = 100;    // тестовый идентификатор

// Сохраняем идентификатор в строку таблицы с индексом row
ui.tableWidget->item(row, 0)->setData(Qt::UserRole, id);

// Сохраняем идентификатор в строку списка с индексом row
ui.listWidget->item(row)->setData(Qt::UserRole, id);
```

Для чтения дополнительных данных, связанных с элементом браузера, используется метод :

```
QVariant data ( int role ) const
```

где **role** — роль или назначение тех данных, которые требуется считать.

Метод возвращает значение типа `QVariant`, которое необходимо преобразовать к требуемому типу данных. Для этого используются методы

Класса **QVariant**:

- `bool toBool () const`
- `QChar toChar () const`
- `QDate toDate () const`
- `QDateTime toDateTime () const`
- `double toDouble ( bool * ok = 0 ) const`
- `int toInt ( bool * ok = 0 ) const`
- `QString toString () const`
- `QStringList toStringList () const`
- `QTime toTime () const`
- `uint toUInt ( bool * ok = 0 ) const`

Пример чтения идентификатора из браузера:

```
unsigned int id;    // тестовый идентификатор

// Считываем идентификатор из строки таблицы с индексом row
id = ui.tableWidget->item(row, 0)->data(Qt::UserRole).toUInt();

// Считываем идентификатор из строки списка с индексом row
id = ui.listWidget->item(row)->data(Qt::UserRole).toUInt();
```