

Механизм Values

В QP7.Framework для передачи данных между объектами можно использовать механизм *Values*. *Values* - это коллекция уровня страницы. Срок жизни коллекции *Values* - текущий запрос. То есть она не сохраняется в сессии и не передается между запросами. Данные в коллекцию *Values* попадают из трех основных источников:

- [HttpRequest.QueryString](#) (для ASP - [Request.QueryString](#))
- [HttpRequest.Form](#) (для ASP - [Request.Form](#))
- Вызов функции [AddValue](#) из пользовательского кода

Идея передачи данных между объектами через коллекцию *Values* хорошо ложится в русло технологии ASP, так как в ней применяется прямое исполнения кода (как написано). В случае же ASP.NET из-за событийно-ориентированного подхода есть ряд ограничений использования этого механизма.

Передача данных от родительского объекта к дочернему

Рассмотрим дерево контролов некоторой страницы ASP.NET. При статической загрузке контролов событие *Init* распространяется в обратном порядке (от листьев к корню), поэтому передача данных в этом случае невозможна. Поэтому QP7.Framework использует динамическую загрузку вместе с собственной последовательностью событий *Init*. Стоит отметить что при загрузке объекта через *Presentation*, используется контрол *qp:placeholder*, и он вроде бы должен загружаться статически, но фактически он представляет собой оболочку, которая в свою очередь динамически загружает нужный контрол. Так как сама оболочка загружается статически, то передача данных из родительского объекта в дочерний, который вызван через *qp:placeholder* невозможна. У данной реализации есть один неприятный побочный эффект: из-за использования собственной последовательности событий *Init* перестает правильно работать назначение контролам клиентских ID, а следовательно потом страница не может корректно восстановить *Viewstate*. Это приводит к тому, что при использовании серверных контролов внутри контролов QP7 первые могут сохранить свое состояние. Решение этой проблемы - заключение такого контрола или группы контролов в отдельный объект шаблона (страницы), который должен быть вызван одним из следующих способов:

- ```
<qp:placeholder calls="News" simple="true" runat="server" />
```
- ```
ShowObjectSimple("News", this);
```

то есть с использованием атрибута *simple = true* для *qp:placeholder* или с помощью метода [ShowObjectSimple](#). При этом передача данных в такой объект работать не будет.

Передача данных от родительского объекта к дочернему

Возможна при тех же ограничениях, что и в предыдущем случае. При этом надо внимательно следить за тем, чтобы значения не использовались раньше, чем были получены.

Передача данных между итерациями Publishing Container

Если для вывода информации Publishing Container используется сгенерированный по умолчанию [Repeater](#), то записывать значение для следующей итерации в Value лучше всего в обработчике события [OnItemDataBound](#), а использовать в логике для формирования текущего вывода - в обработчике события [OnItemCreated](#).

Использование Values

При использовании *Values* в качестве составных частей фильтров *Publishing Container* рекомендуется использовать либо функцию [NumValue](#), либо функцию [StrValue](#). Первая подходит для числовых параметров: если на ее входе не число, то она возвращает 0, тем самым гарантируя числовой результат. Вторая функция хорошо подходит для строковых параметров: она удваивает апострофы, в отличие от функции [Value](#), которая их удаляет. Рекомендованные способы применения этих функций в фильтрах:

```
"[content_item_id] = " + NumValue("id")
```

```
"[Name] = '" + StrValue("name") + "'"
```

Аккуратная работа с апострофами необходима, чтобы избежать возможности [SQL-инъекции](#). Если нужно получить значение из коллекции Values в неизменном виде (без удалений и замен апострофов, преобразования к числу) можно воспользоваться функцией [DirtyValue](#). Она может быть полезна при передаче фильтра целиком через коллекцию Values. Но в этом случае необходимо при формировании фильтра удостовериться в невозможности [SQL-инъекции](#). Добавить программно новое значение можно с помощью функции [AddValue](#). Не рекомендуется прямая работа с коллекцией Values, лучше использовать [функции доступа](#), упомянутые ранее в этом разделе.

From:

<http://wiki.qpublishing.ru/> - **QP7.Framework Docs**

Permanent link:

<http://wiki.qpublishing.ru/doku.php?id=development:values>



Last update: **2012/06/19 13:46**