

Настройка на сайте интегрированных правил доступа бэкенда

Часто требуется создать закрытый раздел на сайте с возможностью входа для зарегистрированных пользователей. Данная функциональность позволяет использовать для этой цели пользователей QP7 вместо того, чтобы самостоятельно создавать хранилище пользовательских данных на базе контентов QP7.

Настройка

Нужно в [конфигурационном файле QP7](#) определить переменные через которые будут передаваться идентификаторы пользователя и группы:

```
<app_var app_var_name="security_UID_varname_VB">Session("qp_UID")</app_var>
<app_var app_var_name="security_GID_varname_VB">Session("qp_GID")</app_var>
<app_var
app_var_name="security_UID_varname_CSharp">Session["qp_UID"]</app_var>
<app_var
app_var_name="security_GID_varname_CSharp">Session["qp_GID"]</app_var>
```

Для каждого языка .NET, поддерживаемого QP7.Framework (VB.NET и C#) есть свои переменные, так как синтаксис между языками может различаться. По умолчанию для VB.NET используются переменные `Session("qp_UID")` и `Session("qp_GID")`, а для C# - `Session["qp_UID"]` и `Session["qp_GID"]`.

Для аутентификации можно использовать метод [AuthenticateUser](#). В случае успеха результат выполнения метода будет отличен от 0 и это будет идентификатор пользователя, который нужно сохранить в переменной, заданной в [конфигурационном файле QP7](#).

```
Session["qp_UID"] =
Quantumart.QPublishing.Permissions.AuthenticateUser(username, password);
```

Альтернативный вариант - сохранить в переменной `Session["qp_GID"]` идентификатор группы и выставить `Session["qp_UID"]` в `null`. Тогда QP7 будет использовать правила доступа для группы. Список групп для пользователя можно получить с помощью метода [GetRootGroupsForUser](#).

Если посетители сайта имеют возможность создать нового пользователя, то следует использовать метод [AddUser](#). Пользователь, создаваемый таким образом, по умолчанию не имеет доступа в бэкенд (у него в [профиле](#) выставлена опция **Заблокирован (Disabled)**)

```
Quantumart.QPublishing.Permissions.AddUser(username, password, First_Name,
Last_Name, Email);
```

Обычно новых пользователей помещают в одну или несколько групп с помощью метода [AddUserToGroup](#):

```
Quantumart.QPublishing.Permissions.AddUserToGroup(int userId, int groupId);
```

Чтобы найти нужную группу можно воспользоваться методом [GetGroupInfo](#):

```
DataTable dt = GetGroupInfo(group_name); //group_name is a string  
int group_id = dt.Rows["group_id"];
```

Установите в настройках объекта типа [Publishing Container](#) галочку **Использовать правила доступа бэкенда** и определите желаемый уровень доступа для каждого объекта типа [Publishing Container](#), который должен фильтровать информацию на основе прав доступа бэкенда.

QP7 позволяет программно назначать права на статьи пользователям и группам. Это особенно полезно при наличие на сайте контента, создаваемого пользователем. И в таком случае пользователю можно предоставить возможность самому решить, каким пользователям (группам) дать доступ на созданную им информацию.

QP7 также поддерживает вложенные группы, где дочерняя группа наследует права родительской, таким образом косвенно влияя на права пользователей. Существуют методы позволяющие программно устанавливая отношения между группами.

В QP7 API много методов для работы с режимом интегрированных прав доступа. Наиболее примечательные из них:

- [AddUserToItemPermission](#)
- [AddGroupToItemPermission](#)
- [GetPermissionLevels](#)
- [AddChildGroupToParentGroup](#)

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

Permanent link:
http://wiki.qpublishing.ru/doku.php?id=development:integrated_security



Last update: **2009/08/25 19:37**