Плагин dbquery

—- plugin —- description: Безопасное отображение результатов запросов из базы данных. author : Andreas Gohr email : dokuwiki@cosmocode.de type : helper, syntax lastupdate : 2025-07-24 compatible : kaos depends : conflicts : similar : sql, sqlcomp tags : sql, db, database, mysql, postgres, sqlite, pdo

downloadurl: https://github.com/cosmocode/dokuwiki-plugin-dbquery/zipball/master bugtracker : https://github.com/cosmocode/dokuwiki-plugin-dbquery/issues sourcerepo : https://github.com/cosmocode/dokuwiki-plugin-dbquery donationurl:

screenshot_img :

COSMO

Этот плагин позволяет отображать результаты запроса к базе данных либо в виде таблицы, либо в виде HTML- отформатированного статуса. В отличие от других подобных плагинов, этот плагин строго разделяет определение запросов для выполнения и часть, где они используются. Это позволяет использовать правила ACL для предотвращения выполнения пользователями произвольных запросов к подключенной базе данных.

Подключение к базе данных осуществляется через PDO, сам плагин не зависит от конкретной базы данных.

Установка

Найдите и установите плагин с помощью Менеджера расширений. Инструкции по ручной установке плагинов см. в разделе «Плагины».

Конфигурация

Подключение к базе данных необходимо настроить в диспетчере конфигураций, используя DSN-код подключения, а также имя пользователя и пароль.

DSN определяет способ подключения к базе данных и зависит от драйвера PDO.

Примечание: драйверы PDO должны быть установлены и включены в вашей конфигурации PHP.

Вот несколько типичных DSN:

```
mysql:host=localhost;port=3306;dbname=testdb
mysql:unix_socket=/tmp/mysql.sock;dbname=testdb
pgsql:host=localhost;port=5432;dbname=testdb
sqlite:/opt/databases/mydb.sq3
```

Более подробную информацию о формате DSN, специфичном для вашего драйвера, вы можете найти здесь:

- PDO_SQLITE DSN
- PDO_PGSQL DSN
- PDO_ODBC DSN
- PDO_OCI DSN
- PDO_SQLSRV DSN
- PDO_MYSQL DSN
- PDO_INFORMIX DSN
- PDO_IBM DSN
- PDO_FIREBIRD DSN
- PDO_DBLIB DSN
- PDO_CUBRID DSN

В конфигурации также можно указать пространство имен, в котором определены запросы.

Множество баз данных

Плагин поддерживает настройку нескольких баз данных. Для этого можно указать несколько DSN-соединений — каждое на отдельной строке.

Каждая строка должна начинаться с метки для соответствующего соединения, разделенной пробелом. Эта метка впоследствии используется для выбора соединения, которое будет использоваться для запроса (см. раздел «Макросы»s ниже). Первая строка используется в качестве соединения по умолчанию, если соединение не указано.

При желании вы также можете указать пользователя и пароль (разделенные пробелами). Если они не указаны, используются пользователь и пароль из соответствующих полей конфигурации.

Вот пример определения двух соединений с именами «struct» и «users»:

```
struct sqlite:/var/www/dokuwiki/data/meta/struct.sqlite3
users  mysql:host=localhost;port=3306;dbname=userdb       myuser secret
```

Заполнители пути

При использовании плагина для доступа к существующим базам данных SQLite (например, из структуры) полезно иметь возможность ссылаться на локальный каталог данных. Это становится вдвойне полезным при использовании плагина в фермерской системе, когда необходимо определить псевдонимы в фермерской системе, которые будут наследоваться животными.

Плагин поддерживает использование заполнителей %DATA_DIR%для %META_DIR%ссылок на каталоги данных или data/meta соответственно. Их можно использовать в конфигурации DSN. В фермерской среде эти переменные всегда будут указывать на каталоги животного.

Настройка запроса

Плагин использует предопределенные запросы, к которым впоследствии можно получить доступ по их имени. Все запросы настраиваются в выделенном пространстве имен (`dbquery` по умолчанию). Вам следует **ограничить доступ к этому пространству имен** только для доверенных пользователей с помощью списков контроля доступа (ACL).

Для каждого запроса существует своя страница в пространстве имен dbquery. Имя страницы впоследствии можно использовать для выполнения запроса (см. ниже).

Базовое определение запроса

Чтобы задать SQL-запрос, просто создайте страницу как обычно и добавьте блок кода (можно использовать подсветку синтаксиса DokuWiki) с вашим SQL-запросом.

Примечание: из соображений безопасности разрешены только запросы SELECT.

На эту страницу можно добавить всевозможные другие синтаксисы, например, для описания того, что делает запрос и т. д. Плагин dbquery будет использовать **первый** найденный на этой странице блок кода в качестве SQL-запроса.

```
====== My First Query ======

<code sql>
SELECT * FROM usertable;
</code>
```

При последующем выполнении запроса (см. ниже) пользователю будет показана таблица со всеми результирующими строками.

Замена переменных

Использование предопределенных переменных позволяет сделать запросы более динамичными. Это дает возможность создавать запросы в зависимости от страницы, с которой они выполняются, или от пользователя, для которого они выполняются. Доступны следующие переменные:

`:user`	Имя пользователя, вошедшего в систему в момент выполнения запроса.
`:mail`	Адрес электронной почты текущего пользователя, вошедшего в систему на момент выполнения запроса.
`:groups`	Группы текущего пользователя, вошедшего в систему на момент выполнения запроса. Они должным образом настроены для использования в `IN` запросе.
`:id`	Полный идентификатор страницы, на которой выполняется запрос, с префиксом в виде двоеточия `:`
`:page`	Страница без пространства имен, на которой выполняется запрос.
`:ns`	Пространство имен страницы, на которой выполняется запрос, с префиксом в виде двоеточия `:`. Корневое пространство имен — это`:`

```
====== My First Query ======

<code sql>
SELECT * FROM usertable WHERE primarygroup IN (:groups);
</code>
```

Макросы

На страницу запроса можно добавить дополнительные макросы, которые будут влиять на выполнение запроса или отображение результата. В настоящее время поддерживаются следующие макросы:

Используйте базу данных с меткой «label» вместо подключения по умолчанию. Инструкции по определению нескольких подключений к базам данных см. в разделе «Несколько баз данных » выше.
Когда результат запроса отображается в виде таблицы, таблица транспонируется таким образом, что заголовки столбцов располагаются слева, а строки отображаются в виде столбцов. Это следует делать только для очень коротких наборов результатов и большого количества запрошенных столбцов. Например, когда в результате больше столбцов, чем строк.

Запросы статуса

Часто возникает необходимость отобразить конкретную информацию в зависимости от результата запроса. Запросы статуса позволяют это сделать.

Запрос статуса обладает следующими свойствами:

В результате получается ровно одна строка. В результате содержится поле с именем или псевдонимом.`status` Результат может дополнительно содержать другое поле, названное или имеющее псевдоним.`result` Для запросов, соответствующих этому определению, плагин не будет отображать таблицу результатов, а вместо этого будет искать пользовательский HTML-код для отображения в соответствии с определением на странице запроса.

Для определения этого HTML- кода необходимо добавить дополнительные разделы на страницу запроса. Каждый раздел должен быть назван в соответствии с вашим статусом и содержать блок кода с указанием `html` типа.

```
====== Balance Checking ======

<code sql>
SELECT
       IIF(balance >= 0, 'green', 'red') AS status,
       balance AS result
  FROM account
 WHERE user = :user;
</code>


===== These are the possible status codes =====
```

```
==== red ====

<code html>
<div style="background-color: #F8CECC">In debt: :result</div>
</code>


==== green ====

<code html>
<div style="background-color: #D5E8D4">All okay: :result</div>
</code>
```

Приведенный выше пример должен прояснить ситуацию. SQL-запрос проверяет баланс текущего пользователя и возвращает либо то, `red` либо другое `green` в качестве статуса. Поиск в разделах ниже отобразит либо красный, либо зеленый прямоугольник, а в `result` поле будет отображаться фактический баланс.

Выполнение запроса

Для выполнения предопределенного запроса используется следующий синтаксис запроса: `{{QUERY:queryname}}`, где `queryname` — это имя страницы запроса в пространстве имен dbquery.

В результате получается либо таблица со всеми строками, либо HTML-код статуса .

Примечание: DokuWiki обычно кэширует содержимое страниц. Для запросов с высокой динамичностью необходимо отключить кэширование страниц , вставив ~~NOCACHE~~ макрос на страницу. По умолчанию это не делается, поскольку могут быть запросы, которые можно обновлять только раз в день (см. cachetime).

Отображение контента

По умолчанию плагин автоматически определяет синтаксис ссылок DokuWiki в содержимом результатов и соответствующим образом создает ссылки при отображении табличных результатов. Однако есть несколько нюансов:

- Синтаксис ссылки должен содержать только содержимое ячейки, например, она начинается и заканчивается соответственно символами `[[` и `]]`.
- Распознаются только внутренние ссылки `http://`, `https://`.
- Эта функция не использует полный парсер вики для анализа содержимого, поэтому никакой другой синтаксис не распознается и не обрабатывается особым образом.

Плагин может опционально обрабатывать полное содержимое вики-синтаксиса. Для включения этой функции необходимо обращаться к столбцу, содержащему вики-синтаксис, используя псевдоним, заканчивающийся на `_wiki:`

```
SELECT name, content AS content_wiki FROM mytable;
```

В приведенном выше примере содержимое `content` столбца будет интерпретировано как синтаксис Wiki.

Примечание: в целях совместимости заголовки всегда будут интерпретироваться как выделенные жирным шрифтом абзацы, чтобы не мешать работе оглавления, редактированию разделов и другим функциям.

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=wiki:plugin:dbquery

Last update: **2026/01/18 18:58**