Работа с Postgresql в C#

Язык программирования C#

Не так давно занялся разработкой приложения на C# работающего на базе PostreSQL, появилась информация которой спешу поделиться. В статье я опишу что необходимо для подключения к PostgreSQL из приложения написанного на C#, и как с ним работать на примерах.

Для того чтобы иметь возможность работать с базами данных PostgreSQL в нашем случае будем использовать Npgsql — .Net Data Provider for Postgresql скачиваем, добавляем Npgsql.dll в свой проект и копируем в папку с скомпилированным проектом необходимые библиотеки (чтобы работало): Npgsql.dll и Mono.Security.dll.

Начнём с подключения к базе:

string conn\_param = "Строка подключения к базе"; //Например: "Server=127.0.0.1;Port=5432;User Id=postgres;Password=mypass;Database=mybase;"

string sql = "текст запроса к базе данных";

NpgsqlConnection conn = new NpgsqlConnection(MainParams.ConnectionString);

NpgsqlCommand comm = new NpgsqlCommand(sql, conn);

conn.Open(); //Открываем соединение.

result = comm.ExecuteScalar().ToString(); //Выполняем нашу команду.

conn.Close(); //Закрываем соединение.

Как видно, ничего сложного, всё как всегда.

comm.ExecuteScalar() — возвращает значение.

comm.ExecuteNonQuery() — выполняет запрос на изменение данных.

Пример выборки массива данных:

NpgsqlConnection con = new NpgsqlConnection("строка подключения");

NpgsqlCommand com = new NpgsqlCommand("select \* from my\_table order by \"NAME\"", con);

con.Open();

NpgsqlDataReader reader;

reader = com.ExecuteReader();

while (reader.Read())

{

try

{

string result = reader.GetString(1);//Получаем значение из второго столбца! Первый это (0)!

}

catch { }

}

con.Close();

## Настройка подключения PHP и PostgreSQL

Первое что нужно сделать, это убедиться, что в каталоге с PHP присутствует библиотека libpq.dll. Затем в файле php.ini найди строку, отвечающую за подключение модуля PostgreSQL и раскомментировать ее, а именно

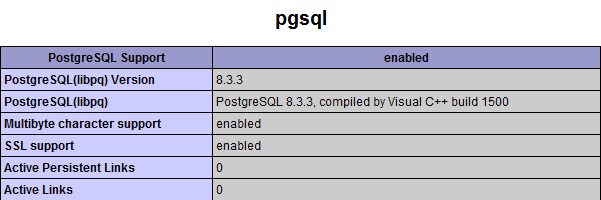
Не подключено

;extension=php\_pgsql.dll

Подключено

extension=php\_pgsql.dll

Перезапускаем Apache и запускаем phpinfo(), и если Вы найдете блок pgsql, то все, библиотеку PostgreSQL Вы подключили.



Если у Вас нет данного раздела, то попробуйте скопировать libpq.dll из каталога PostgreSQL (по умолчанию C:\Program Files\PostgreSQL и так далее) в каталог с PHP с заменой старого, и перезапустите Apache еще раз.

## Основные функции PHP для работы с PostgreSQL

**pg\_connect** – открывает соединение с базой данных, возвращает указатель соединения.

**pg\_query** – выполняет запрос к базе данных, возвращает результат запроса.

**pg\_fetch\_assoc** – преобразовывает результат запроса в ассоциативный массив.

**pg\_close** – закрывает соединение с базой данных.

Пример работы перечисленных функций

<?php

$connect\_string = "host=localhost port=5432 dbname=testbase user=admin password=12345";

$dbconnect = pg\_connect($connect\_string);

$query = "select 'Привет!' as field\_1, 123 as field\_2";

$result = pg\_query($dbconnect, $query);

$result = pg\_fetch\_assoc($result);

echo $result['field\_1'] . '</br>' . $result['field\_2'];

pg\_close($dbconnect);

?>

Результат



В данном случае мы подключаемся к серверу, который расположен локально, по порту 5432, к базе testbase под пользователем admin с паролем 12345 и выполняем запрос (просто статические данные) к базе данных, затем мы преобразовываем результат в ассоциативный массив, выводим на экран и закрываем соединение.

**pg\_num\_rows** – возвращает количество строк в наборе данных, который вернул нам запрос.

**pg\_num\_fields** - возвращает количество полей в наборе данных, который вернул нам запрос.

**pg\_fetch\_result** – возвращает одно значение из результата запроса.

**pg\_fetch\_row** - преобразовывает результат запроса в массив.

Пример

<?php

$connect\_string = "host=localhost port=5432 dbname=testbase user=admin password=12345";

$dbconnect = pg\_connect($connect\_string);

$query = "select 'Привет!' as field\_1, 123 as field\_2";

$result = pg\_query($dbconnect, $query);

$NumRows = pg\_num\_rows($result);

$NumFields = pg\_num\_fields($result);

$field\_1 = pg\_fetch\_result($result, 0, 0);

$result\_array = pg\_fetch\_row($result);

echo 'Количество строк - ' . $NumRows . '</br>';

echo 'Количество полей - ' . $NumFields . '</br>';

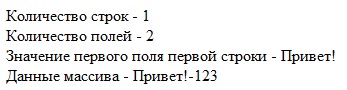
echo 'Значение первого поля первой строки - ' .$field\_1 . '</br>';

echo 'Данные массива - ' . $result\_array[0] . '-' . $result\_array[1];

pg\_close($dbconnect);

?>

Результат



**pg\_fetch\_object** – преобразовывает результат запроса в объект.

**pg\_field\_name** – возвращает название поля по его номеру.

**pg\_field\_num** – возвращает номер поля по его названию.

**pg\_version** – возвращает сведения о протоколе взаимодействия, версии клиента и версии сервера в виде массива.

**pg\_escape\_string** – экранирует спецсимволы в строке.

**pg\_dbname** – возвращает название базы данных.

**pg\_affected\_rows** – возвращает количество затронутых строк при update, insert и delete.

Пример

<?php

$connect\_string = "host=localhost port=5432 dbname=testbase user=admin password=12345";

$dbconnect = pg\_connect($connect\_string);

$query = "select 'Привет!' as field\_1, 123 as field\_2";

$result = pg\_query($dbconnect, $query);

$result\_object = pg\_fetch\_object($result);

echo 'Значение свойства - ' . $result\_object->field\_1. '</br>';

echo 'Название первого поля - ' . pg\_field\_name($result, 0). '</br>';

echo 'Номер поля field\_1 - ' . pg\_field\_num($result, 'field\_1'). '</br>';

echo 'Версия PostgreSQL - ' . pg\_version($dbconnect)['server'] . '</br>';

pg\_close($dbconnect);

?>