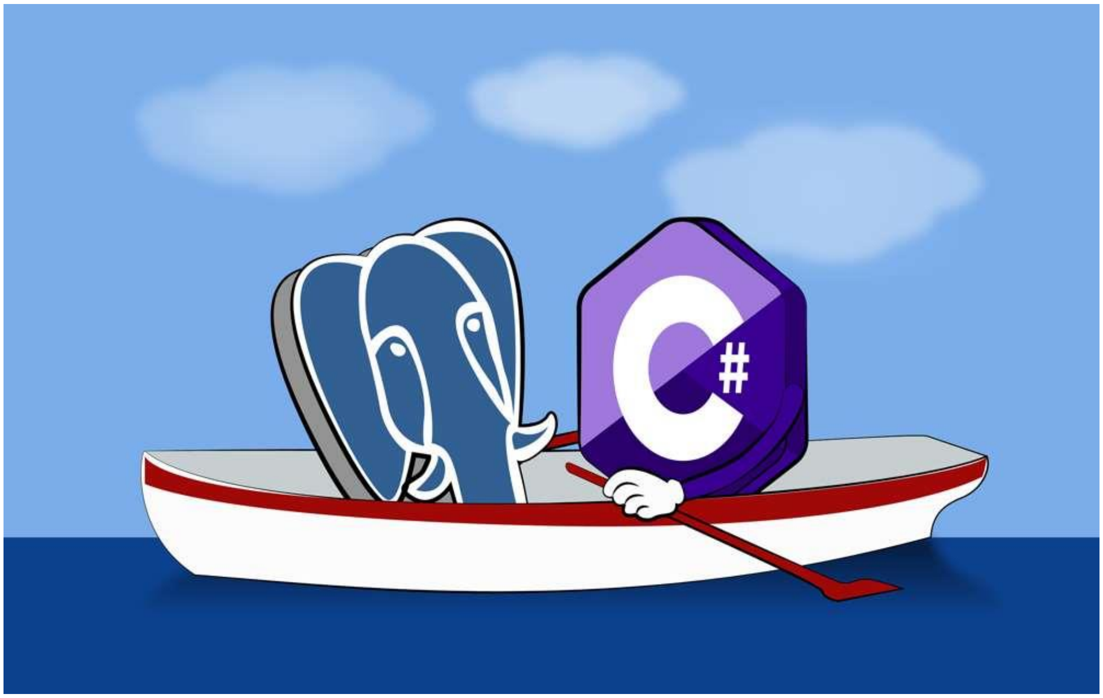

.NET WITH POSTGRSSQL

Documentation

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you’re ready to add your content, just click here and start typing.]

Dhruvil Dobariya

dhruvildobariya21@gmail.com

INDEX

CRUD OPERATION

1.1 INTRODUCTION

- ASP.NET Core API implementation is same as we do with **SqlClicnt** or **MySqlClient**.
- We just change provider in DAL which is **Npgsql**.

Example:

```
using Npgsql;
using ProductAPI.Models;
using System.Data;

namespace ProductAPI.DAL.Services
{
    public class DBProductContext : IDbProductContext
    {
        private readonly string _connectionString = "Server = localhost; Database = 
ProductDB; Port = 5432; Username = postgres; Password=Admin;";

        #region GetProducts

        public List<Product> GetProducts()
        {
            try
            {
                List<Product> products = new List<Product>();
                using(var connection = new NpgsqlConnection(_connectionString))
                {
                    var dataAdupter = new NpgsqlDataAdapter("Select \"Id\", 
\"Name\", \"Price\", \"Quantity\" from \"Product\"", connection);
                    DataTable dt = new DataTable();
                    dataAdupter.Fill(dt);
                    foreach(DataRow item in dt.Rows)
                    {
                        Product product = new Product();
                        if (item["Id"] != DBNull.Value)
                        {
                            product.Id = Convert.ToInt32(item["Id"]);
                        }
                        if (item["Name"] != DBNull.Value)
                        {
                            product.Name = item["Name"].ToString();
                        }
                        if (item["Price"] != DBNull.Value)
                        {
                            product.Price = Convert.ToDecimal(item["Price"]);
                        }
                        if (item["Quantity"] != DBNull.Value)
                        {
                            product.Quantity = Convert.ToInt32(item["Quantity"]);
                        }
                        products.Add(product);
                    }
                }
                return products;
            }
            catch { throw; }
        }

        #endregion GetProducts

        #region GetProductById

        public Product GetProductById(int id)
        {
            try
            {
                Product product = new Product();
                using (var connection = new NpgsqlConnection(_connectionString))
                {
                    var dataAdupter = new NpgsqlDataAdapter($"Select \"Id\",
\"Name\", \"Price\", \"Quantity\" from \"Product\" Where \"Id\" = {id}",
connection);
                    DataTable dt = new DataTable();
                    dataAdupter.Fill(dt);
                    if(dt.Rows.Count > 0)
                    {
                        DataRow item = dt.Rows[0];
                        if (item["Id"] != DBNull.Value)
                        {
                            product.Id = Convert.ToInt32(item["Id"]);
                        }
                        if (item["Name"] != DBNull.Value)
                        {
                            product.Name = item["Name"].ToString();
                        }
                        if (item["Price"] != DBNull.Value)
                        {
                            product.Price = Convert.ToDecimal(item["Price"]);
                        }
                        if (item["Quantity"] != DBNull.Value)
                        {
                            product.Quantity = Convert.ToInt32(item["Quantity"]);
                        }
                    }
                }
                return product;
            }
            catch { throw; }
        }

        #endregion GetProductById

        #region AddProduct

        public int AddProduct(Product product)
        {
            try
            {
                using(var connection = new NpgsqlConnection(_connectionString))
                {
                    NpgsqlCommand command = new NpgsqlCommand("Insert into 
\"Product\" (\"Name\", \"Price\", \"Quantity\") values (@Name, @Price, @Quantity)", 
connection);
                    command.Parameters.AddWithValue("@Name", product.Name);
                    command.Parameters.AddWithValue("@Price", product.Price);
                    command.Parameters.AddWithValue("@Quantity", product.Quantity);
                    connection.Open();
                    return command.ExecuteNonQuery();
                }
            }
            catch { throw; }
        }

        #endregion AddProduct

        #region UpdateProduct

        public int UpdateProduct(Product product)
        {
            try
            {
                using (var connection = new NpgsqlConnection(_connectionString))
                {
                    NpgsqlCommand command = new NpgsqlCommand("Update \"Product\" 
set \"Name\" = @Name, \"Price\" = @Price, \"Quantity\" = @Quantity where \"Id\" = 
@Id", connection);
                    command.Parameters.AddWithValue("@Id", product.Id);
                    command.Parameters.AddWithValue("@Name", product.Name);
                    command.Parameters.AddWithValue("@Price", product.Price);
                    command.Parameters.AddWithValue("@Quantity", product.Quantity);
                    connection.Open();
                    return command.ExecuteNonQuery();
                }
            }
            catch { throw; }
        }

        #endregion UpdateProduct

        #region DeleteProduct

        public int DeleteProduct(int id)
        {
            try
            {
                using (var connection = new NpgsqlConnection(_connectionString))
                {
                    NpgsqlCommand command = new NpgsqlCommand("Delete from 
\"Product\" where \"Id\" = @Id", connection);
                    command.Parameters.AddWithValue("@Id", id);
                    connection.Open();
                    return command.ExecuteNonQuery();
                }
            }
            catch { throw; }
        }

        #endregion DeleteProduct
    }
}
```