

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4**

Настройка Hibernate

тема

Преподаватель

Студент КИ18-166 031831229

номер группы, зачетной книжки

подпись, дата

подпись, дата

А.К. Погребников

инициалы, фамилия

В.А. Прекель

инициалы, фамилия

Красноярск 2020

## 1 Цель работы

Настроить программное подключение к БД.

## 2 Общая постановка задачи

Подключиться к БД, сделать запросы.

## 3 Ход работы

Используется ADO.NET.

### Листинг 1 – MyStore/MyStore.AdoNet/Program.cs

```
using System;
using System.Collections.Generic;
using System.Text;

using Npgsql;

using NpgsqlTypes;

using MyStore.Data.Entity;
using MyStore.Data;

Console.OutputEncoding = Encoding.UTF8;
var conn = new
NpgsqlConnection("Host=localhost;Database=postgres;Username=postgres;Password=qw
erty123");
conn.Open();

using var cmd = new NpgsqlCommand("SELECT version();", conn);
string ver = cmd.ExecuteScalar() as string ?? "";
Console.WriteLine(ver);
Console.WriteLine();

string GetCustomerFirstName(int id)
{
    using var command = new NpgsqlCommand("SELECT \"FirstName\" FROM
\"Customers\" WHERE \"CustomerId\"=@id;", conn);
    command.Parameters.AddWithValue("id", NpgsqlDbType.Integer, id);
    return command.ExecuteScalar() as string ?? "";
}

Console.WriteLine("Имя покупателя с id=1");
Console.WriteLine(GetCustomerFirstName(1));
Console.WriteLine();

IEnumerable<string> GetCustomerFirstNames(int startId, int endId)
{
    using var command =
        new NpgsqlCommand("SELECT \"FirstName\" FROM \"Customers\" WHERE
\"CustomerId\" BETWEEN @startId AND @endId;",
        conn);
    command.Parameters.AddWithValue("startId", NpgsqlDbType.Integer, startId);
    command.Parameters.AddWithValue("endId", NpgsqlDbType.Integer, endId);
```

```

        using var reader = command.ExecuteReader();
        while (reader.Read())
        {
            yield return reader.GetString(0);
        }
    }

    Console.WriteLine("Имена покупателей с id от 1 по 5");
    foreach (var i in GetCustomerFirstNames(1, 5))
    {
        Console.WriteLine(i);
    }

    Console.WriteLine();

    int AddCustomer(Customer customer)
    {
        using var command =
            new NpgsqlCommand(
                @"INSERT INTO ""Customers"" VALUES
                (DEFAULT, @firstName, @lastName, @honorific, @email, @passwordHash,
                @passwordSalt, NULL)
                RETURNING ""CustomerId""", conn);
        command.Parameters.AddWithValue("firstName", NpgsqlDbType.Varchar,
            customer.FirstName);
        command.Parameters.AddWithValue("lastName", NpgsqlDbType.Varchar,
            customer.LastName);
        command.Parameters.AddWithValue("honorific", NpgsqlDbType.Varchar,
            customer.Honorific);
        command.Parameters.AddWithValue("email", NpgsqlDbType.Varchar,
            customer.Email);
        command.Parameters.AddWithValue("passwordHash", NpgsqlDbType.Bytea,
            customer.PasswordHash);
        command.Parameters.AddWithValue("passwordSalt", NpgsqlDbType.Integer,
            customer.PasswordSalt);
        return (int) (command.ExecuteScalar() ?? 0);
    }

    var salt = Crypto.GenerateSaltForPassword();
    var newId = AddCustomer(new Customer
    {
        FirstName = "Максим",
        LastName = "Тимофеев",
        Honorific = "Даю.",
        Email = "dawfdfwa@fsaf.ru",
        PasswordHash = Crypto.ComputePasswordHash("123456", salt),
        PasswordSalt = salt
    });
    Console.WriteLine($"Создан новый покупатель с id={newId} и именем
    {GetCustomerFirstName(newId)}");

    conn.Close();

```