

SQL заявки в C#



Учителски екип

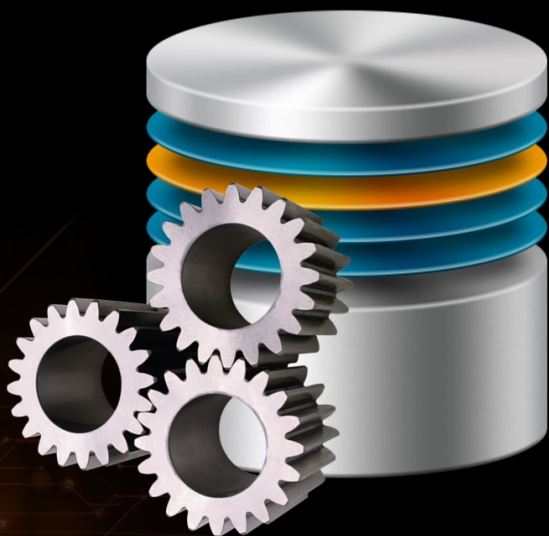
Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. SQL Client доставчик на данни





SQL Client доставчик на данни

Изпълнение на заявки

SqlClient доставчик на данни

- **SqlConnection**

- Създава връзката към БД на SQL Server

- **SqlCommand**

- Изпълнява SQL команди върху SQL Server чрез създадената връзка
- Може да приема параметри (**SqlParameter**)

- **SqlDataReader**

- Извлича данни (множество от записи) от SQL Server като резултат от изпълнението на SQL заявка

SqlConnection класът

- **SqlConnection** създава връзка към SQL Server БД
 - Нуждае се от валиден низ на връзката
- Пример за низ на връзката string/:

Име на БД

```
Data Source=(local)\SQLEXPRESS;Initial  
Catalog=Northwind;Integrated Security=true;
```

- Свързване към SQL Server:

```
SqlConnection con = new SqlConnection(  
    "Server=.;  
    Database=Northwind;  
    Integrated Security=true");  
con.Open();
```

SqlConnection – Пример

- Създаване и отваряне на връзка към SQL Server (база данни **ItKaria**)

```
SqlConnection dbCon = new SqlConnection(  
    "Server=.\SQLEXPRESS; " +  
    "Database=ItKaria; " +  
    "Integrated Security=true");  
dbCon.Open();  
using (dbCon)  
{  
    // Use the connection to execute SQL commands here ...  
}
```


Низ на връзката

- Низът на връзката към БД
 - Дефинира параметрите, които са нужни, за да се създаде връзка с базата данни
- Настройки за SQL Server низа:
 - **Data Source / Server** – име на сървъра / IP адрес + име на инстанцията на БД
 - **Database / Initial Catalog** – име на БД
 - **User ID / Password** – потребител и парола
 - **Integrated Security** – **false** ако се подават потребител и парола

Работа със SqlConnection

- Изрично отваряне и затваряне на връзката
 - **Open()** и **Close()** методи
 - Работи чрез пула на връзките /connection pool/
- Връзките към БД са обекти имплементиращи **IDisposable**
 - Винаги използвайте **using** конструкцията в C#!

SqlCommand

- По-важни методи

- **ExecuteScalar()**

- Връща единствена стойност (стойността на първата колона на първия ред от множеството от резултати) (като **System.Object**)

- **ExecuteReader()**

- Връща **SqlDataReader**

- Това е курсор върху върнатите записи (резултатно множество - result set)

- **CommandBehavior** – задава някои опции

- **ExecuteNonQuery()**

- Използва се за SQL команди, които не са за извличане, например **INSERT**
 - Връща броя на засегнатите редове (**int**)

SqlCommand – Пример

```
SqlConnection dbCon = new SqlConnection(  
    "Server=.; " +  
    "Database=ItKariera; " +  
    "Integrated Security=true");  
dbCon.Open();  
using(dbCon)  
{  
    SqlCommand command = new SqlCommand(  
        "SELECT COUNT(*) FROM Employees", dbCon);  
    int employeesCount = (int) command.ExecuteScalar();  
    Console.WriteLine("Employees count: {0} ", employeesCount);  
}
```

SqlDataReader

- **SqlDataReader** извлича поредица от записи (курсор) върнати като резултат от SQL команда
 - Данните са достъпни само за четене (не може да се променят)
 - Преход само напред по редове – не можем да се върнем назад
- Важни свойства и методи:
 - **Read()** – премества курсора напред, като връща `false`, в случай че няма следващ запис
 - **Indexer[]** – връща стойността на текущия запис по дадено име на колона или индекс
 - **Close()** – затваря курсора и освобождава ресурсите

SqlDataReader – Пример

```
SqlConnection dbCon = new SqlConnection(...);
dbCon.Open();
using(dbCon)
{
    SqlCommand command = new SqlCommand("SELECT * FROM Employees", dbCon);
    SqlDataReader reader = command.ExecuteReader();
    using (reader)
    {
        while (reader.Read())
        {
            string firstName = (string)reader["FirstName"];
            string lastName = (string)reader["LastName"];
            decimal salary = (decimal)reader["Salary"];
            Console.WriteLine("{0} {1} - {2}", firstName, lastName, salary);
        }
    }
}
```

Извлича още и още редове,
докато приключи

SQL заявки в C#



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

