

CRUD операции чрез ORM



Учителски екип

Обучение за ИТ кариера

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





CREATE



READ



UPDATE



DELETE

C

R

U

D

CRUD операции

С MiniORM

Какво са CRUD операциите?

- **CRUD** – Създаване **/Create/** - пример: добавяне на нов запис в таблицата
- **CRUD** – Прочитане **/Read/** - пример: прочитане на запис от таблицата
- **CRUD** – Променяне **/Update/** - пример: промяна на един или няколко елемента от запис в таблицата
- **CRUD** – Изтриване **/Delete/** - пример: изтриване на запис в таблицата



Прочитане на информация

От базата данни чрез MiniORM

Използване на DbContext класа

- Създайте инстанция на **DbContext**:

```
var context = new ItCareerDbContext(connectionString);
```

- В конструктора вие може да подадете низ за връзка към БД
- **DbContext** свойства:
 - Всички класове на данни (таблици) са налични като свойства
 - Например: **DbSet<Employee> Employees { get; }**

Прочитане на информацията чрез LINQ заявка

- Изпълнение на **LINQ-към-Данни** заявка чрез клас от данни:

```
var context = new ItCareerDbContext(connectionString)

var employees = context.Employees
    .Where(e => e.JobTitle == "Design Engineer")
    .ToArray();
```

- **Employees** СВОЙСТВО В **DbContext**:

```
public class ItCareerDbContext : DbContext
{
    public DbSet<Employee> Employees { get; }
    public DbSet<Project> Projects { get; }
    public DbSet<Department> Departments { get; }
}
```

Прочитане на информация с LINQ заявка

- Можем и да ползваме **разширяващите методи** за заявка

```
var context = new SoftUniDbContext(connectionString)
var employees = context.Employees
    .Where(c => c.JobTitle == "Design Engineering")
    .Select(c => c.FirstName)
    .ToList();
```

- Намиране на елемент по **ID**

```
var context = new SoftUniEntities()
var project = context.Projects.Single(e => e.Id == 2);
Console.WriteLine(project.Name);
```


Създаване на нови данни

- За да създадете нов запис в таблицата в БД, ползвайте метод **Add(...)** на съответния DbSet:

```
var project = new Project()  
{  
    Name = "Judge System"  
};
```

Създаваме нов **Project** обект

```
context.Projects.Add(project);  
context.SaveChanges();
```

Добавяме обекта към **DbSet**

Изпълняваме SQL заявки

Обновяване на съществуващи данни

- **DbContext** позволява обновяване на свойствата на данните и запазването им в БД
 - Просто заредете данните, променете ги и извикайте **SaveChanges()**
- **DbContext** автоматично следи промените върху данните

```
Employees employee =  
    context.Employees.First();  
employee.FirstName = "Alex";  
context.SaveChanges();
```

SELECT на
първата поръчка

Изпълнява SQL
UPDATE

Изтриване на съществуващи данни

- Изтриването се случва чрез **Remove()** изпълнен върху конкретна колекция от данни
- **SaveChanges()** метода изпълнява самото изтриване в БД

```
Employees employee =  
    softUniEntities.Employees.First();  
softUniEntities.Employees.Remove(employee);  
softUniEntities.SaveChanges();
```

Маркира дадения ред
за изтриване при
следващото запазване

Изпълнява SQL DELETE
командата

CRUD операции чрез ORM



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

