

תכנות באינטרנט

ASP.Net Core

Razor pages

גלעד מרקמן

קריית החינוך פארק המדע,
נס ציונה



ASP.NET Core

GM
Internet Programming
Courses

עדכון בסיס הנתונים Insert

עבודה בשיטה הלא מקושרת

• התחברות למסד הנתונים

- `SqlConnection con = new SqlConnection(connectionString);`

• בניית פקודת SQL

- `SqlCommand cmd = new SqlCommand("SQL String", con);`

• בניית DataAdapter

- `SqlDataAdapter adapter = new SqlDataAdapter(cmd);`

• בניית DataSet לאחסון נתונים מקומי

- `DataSet ds = new DataSet();`

• טעינת הנתונים ממסד הנתונים ל DataSet

- `adapter.Fill(ds, "users");`

• עדכון הנתונים ב DataSet

-

• עדכון בסיס הנתונים על פי ה DataSet המעודכן

- `SqlCommandBuilder builder = new SqlCommandBuilder(adapter);`
- `adapter.Update(ds, "users");`

- נבנה טופס שכולל את כל השדות ברשומה של המשתמש.
- בתחתית הטופס נוסיף מקום להודעות שגיאה מהשרת (כגון: שם המשתמש כבר קיים).
- הפעולה OnGet – תחזיר למשתמש טופס ריק.
- הפעולה OnPost – תבצע את העדכון בבסיס הנתונים.
- בצד השרת אנחנו נשתמש במאפיין מסוג User הכולל את כל השדות הנדרשים. הזנה בטופס תעדכן את המאפיין הרלוונטי באובייקט.
- את האובייקט נשלח לפעולה Insert ב Helper. הפעולה תחזיר 1 אם ההוספה הצליחה ו 1- אם המשתמש כבר קיים.

```
<h1>Register</h1>

<h4>User</h4>
<hr />
<div class="row">
  <div class="col-md-4">
    <form method="post">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="NewUser.FirstName" class="control-label"></label>
        <input asp-for="NewUser.FirstName" class="form-control" />
      </div>
      <div class="form-group">
        <label asp-for="NewUser.LastName" class="control-label"></label>
        <input asp-for="NewUser.LastName" class="form-control" />
      </div>
      <div class="form-group">
        <label asp-for="NewUser.Username" class="control-label"></label>
        <input asp-for="NewUser.Username" class="form-control" />
      </div>
      <div class="form-group">
        <label asp-for="NewUser.Password" class="control-label"></label>
        <input asp-for="NewUser.Password" class="form-control" />
      </div>
      <div class="form-group">
        <label asp-for="NewUser.Email" class="control-label"></label>
        <input asp-for="NewUser.Email" class="form-control" />
      </div>
    </form>
  </div>
</div>
```

```
<div class="form-group">
  <label asp-for="NewUser.Phone" class="control-label"></label>
  <input asp-for="NewUser.Phone" class="form-control" />
</div>
<div class="form-group">
  <label asp-for="NewUser.Birthday" class="control-label"></label>
  <input asp-for="NewUser.Birthday" class="form-control" />
</div>
@if(Model.msg != "")
{
  <div class="form-group">
    <div class="alert alert-danger" role="alert">
      @Model.msg
    </div>
  </div>
}
<div class="form-group">
  <input type="submit" value="Create" class="btn btn-primary" />
</div>
</form>
</div>
</div>
```

```
public class RegisterModel : PageModel
{
    [BindProperty]
    public User NewUser { get; set; }
    public string msg { get; set; } = string.Empty;

    public void OnGet()
    {
    }

    public IActionResult OnPost()
    {
        Helper helper = new Helper();
        int n = helper.Insert(NewUser, "Users");
        if (n == -1)
        {
            msg = "Username already taken.";
            return Page();
        }
        return RedirectToPage("Index");
    }
}
```

- המאפיין `NewUser` – קשור לטופס.
- המאפיין `msg` – נועד לכתובת הודעות שגיאה.
- הפעולה `OnPost` קוראת לפעולה `Insert` ב `Helper` אשר מבצעת את עדכון בסי הנתונים.
- אם מוחזר 1- סימן ששם המשתמש כבר קיים.
- לא לשכוח בדיקת טופס ב JS.

```
public int Insert(User user, string table)
// The Method recieve a user objects and insert it to the Database as new row.
// if the user is already taken the method will return -1.
{
    // התחברות למסד הנתונים
    SqlConnection con = new SqlConnection(conString);

    // SQL בניית פקודת
    string SQLStr = $"SELECT * FROM {table} WHERE Username Like '{user.Username}'";
    SqlCommand cmd = new SqlCommand(SQLStr, con);

    // בניית DataSet
    DataSet ds = new DataSet();

    // טעינת סכימת הנתונים
    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
    adapter.Fill(ds, table);

    if (ds.Tables[table].Rows.Count > 0)
    {
        return -1;
    }
}
```

- אנחנו מבצעים שאילתא עם שם המשתמש החדש שהוזן.
- אם קיים כבר שם משתמש כזה אנחנו מחזירים 1- ולא מעדכנים.
- אם לא קיים שם משתמש כזה אנחנו ממשיכים.

```
// בניית השורה להוספה
DataRow dr = ds.Tables[table].NewRow();
dr["Firstname"] = user.FirstName;
dr["Lastname"] = user.LastName;
dr["Username"] = user.Username;
dr["Password"] = user.Password;
dr["Email"] = user.Email;
dr["Phone"] = user.Phone;
dr["Birthday"] = user.Birthday.ToString();

ds.Tables[table].Rows.Add(dr);

// עדכון הדאטה סט בבסיס הנתונים
SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
int n = adapter.Update(ds, table);
return n;
}
```

- יוצרים שורה חדשה DataRow באמצעות הטבלה, כך שהיא תואמת למבנה הטבלה.
- מזינים לשדות בשורה החדשה את הנתונים מהאובייקט User.
- מוסיפים את השורה לטבלה.
- מעדכנים את הטבלה בבסיס הנתונים.
- מקבלים את מספר השורות ששוננו (תמיד יהיה 1).

- פעולות עדכון ומחיקה של משתמש יודגמו באמצעות השיטה המקושרת.

- למימוש פעולות אילו בשיטה הלא מקושרת הנכם מופנים אל הפעולות ב Helper:

- .Update_disconnected

- .Delete_disconnected

[קישור למחלקה Helper ב-GitHub](#)

- הדגמנו פעולות שונות באמצעות השיטה הלא מקושרת:

- סינון ומיון

- כניסה Login

- הדגמנו פעולות עדכון של בסיס הנתונים באמצעות השיטה הלא מקושרת. פעולת ההרשמה Register המוסיפה רשומה לטבלה.

- למדנו על asp-page-handler. כיצד ניתן ליצור פעולות OnPost שונות בצד השרת, כגון: OnPostFilter, OnPostSort