

Forum:[Hack Forums](#)

Thread: [\[Suggest_Me\] Which way is better : ASP .NET MVC registration form](#)

SIM-1947090 - March 27, 2015, 5:25 am

Edited

Never mind, i found a better solution. I will post it later.

Sorry.

phrootoo - March 27, 2015, 7:27 am

Smells like SQL injection. Please used named parameters in your queries...

SIM-1947090 - March 27, 2015, 8:46 am

(Mar 27, 2015 07:27 AM)phrootoo Wrote: [\[CITE\]](#) Smells like SQL injection. Please used named parameters in your queries...

Yes you are correct, i wasnt paying attention to security flaws, but the way how to send queries to the database. Using either the first method i posted or the second one.

So, does it mean i should use the first example? (the one with FormCollection and manually typed query?) Or the other one, where the queries are sort of "automated" (if im not mistaken...lol)?

Naveen - March 28, 2015, 8:53 am

Use the stored procedures instead of typing query directly.

SIM-1947090 - March 28, 2015, 10:01 am

Cant edit my post god damit..

Anyways, i found this solution, i combined few tutorials from google search and i came on this one:

Controller:

PHP Code

```
<?php
using RegistrationLoginForm.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace RegistrationLoginForm.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

```

// Register
[HttpGet]
public ActionResult Register()
{
    return View();
}
[HttpPost]
public ActionResult Register(People user)
{
    if (ModelState.IsValid)
    {
        using (PeopleDBContext cont = new PeopleDBContext())
        {
            cont.PeopleCon.Add(user);
            cont.SaveChanges();
            ModelState.Clear();
            user = null;
        }
    }

    return View(user);
}

// Login
[HttpGet]
public ActionResult Login()
{
    return View();
}
[HttpPost]
public ActionResult Login(People user)
{
    if (IsValid(user.Email, user.Password))
    {
        return RedirectToAction("Index", "Home");
    }
    else
    {
        ModelState.AddModelError("", "Wrong data");
    }

    // return user model/object to the view
    return View(user);
}

// Check if user is valid (exists in database etc..)
private bool IsValid(string email, string password)
{
    bool isValid = false;

    using (PeopleDBContext db = new PeopleDBContext())
    {
        // take user from database
    }
}

```

```

var user = db.PeopleCon.FirstOrDefault(u => u.Email == email);

// if user exists
if (user != null)
{
    // if the password from database matches the typed password
    if (user.Password == password)
    {
        isValid = true;
    }
}

return isValid;
}
}

```

Model:

PHP Code

```
<?php
```

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```
namespace RegistrationLoginForm.Models
```

```

{
    public class People
    {
        // Required for database creation - Code First!
        public int Id { get; set; }

        [Required]
        [StringLength(20)]
        public string Name { get; set; }

        [Required]
        [StringLength(20)]
        public string Surname { get; set; }

        [Required]
        [EmailAddress]
        [StringLength(150)]
        [Display(Name = "Email address: ")] // this will be passed to the View
        public string Email { get; set; }

        [Required]
        [DataType(DataType.Password)]
        [StringLength(20, MinimumLength = 6)]
        [Display(Name = "Password: ")] // pass this value to the View
    }
}

```

```
    public string Password { get; set; }  
    }  
}
```

The rest of the code isn't really important I guess.

So, how about this one? Is it good enough for start?

But of course, I have to crypt some values (like passwords), do some additional checking (if email already exists in database, password confirmation and so on)

My focus was on storing and taking data to/from database at most part.