

Aula08 – ViewModel/Contexto

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: allbert.almeida@fatec.sp.gov.br

Site: <http://www.allbert.com.br>



/allbert.almeida

Objetivo

- O objetivo dessa aula é entender a necessidade de utilização das ViewsModels e construir uma aplicação que utilize essa referência.

Conceito

- View Model representa um conjunto de uma ou mais Models e outros dados que serão representados em uma View que necessita exibir determinado conjunto de informações;
- No ASP.Net MVC os View Models nos permitem modelar várias entidades a partir de um ou mais modelos em um único objeto.

Modelo

```
public class Pessoa
{
    public int Id { get; set; }
    [Required]
    [MaxLength(200)]
    public string Nome { get; set; }
    public string Endereco { get; set; }
    public string Cidade { get; set; }
    public string Estado { get; set; }
    public int UsuarioId { get; set; }
    public virtual Usuario Usuario { get; set; }
}
```

```
public class Usuario
{
    public int Id { get; set; }
    [Required]
    [MaxLength(200)]
    [EmailAddress]
    public string Email { get; set; }
    [RegularExpression("^.*(?!={4,8})(?!={0-9})(?!={a-z})(?!={A-Z})")]
    public string Senha { get; set; }
    [Compare("Senha")]
    public string ConfirmaSenha { get; set; }
    public bool MantenhaConectado { get; set; }
}
```

Alteração Classe Usuário

```
public class Usuario
{
    public int Id { get; set; }
    [Required]
    [MaxLength(200)]
    [EmailAddress]
    public string Email { get; set; }
    public string Senha { get; set; }
}
```

Alteração Classe Contexto

```
protected override void OnModelCreating(DbModelBuilder mb)
{
    var usu = mb.Entity<Usuario>();
    usu.ToTable("usu_usuario");
    usu.Property(x => x.Id).HasColumnName("usu_codigo");
    usu.Property(x => x.Email).HasColumnName("usu_email");
    usu.Property(x => x.Senha).HasColumnName("usu_senha");

    var pes = mb.Entity<Pessoa>();
    pes.ToTable("pes_pessoa");
    pes.Property(x => x.Id).HasColumnName("pes_codigo");
    pes.Property(x => x.Nome).HasColumnName("pes_nome");
    pes.Property(x => x.Endereco).HasColumnName("pes_endereco");
    pes.Property(x => x.Cidade).HasColumnName("pes_cidade");
    pes.Property(x => x.Estado).HasColumnName("pes_estado");
    pes.Property(x => x.UsuarioId).HasColumnName("usu_codigo");
}
```

Compilar a solução

- Compilar a solução para que os modelos sejam validados e seja possível criar o “CRUD” da classe Usuario; Pessoa;

Usar o comando

- Enable-Migrations
- Adicionar na classe Configuration

```
AutomaticMigrationsEnabled = true;
```

```
AutomaticMigrationDataLossAllowed = true;
```

```
SetSqlGenerator("MySql.Data.MySqlClient", new  
MySql.Data.Entity.MySqlMigrationSqlGenerator());
```


Usar os comandos

- Add-Migration IntialDb
- Update-Database -Verbose

Configuration

- Adicionar na classe Seed (Configuration)

```
context.Usuario.AddOrUpdate(
    p => p.Email,
    new Models.Usuario { Id = 1, Email = "allbert1@allbert.com.br", Senha = "123" },
    new Models.Usuario { Id = 2, Email = "allbert2@allbert.com.br", Senha = "123" },
    new Models.Usuario { Id = 3, Email = "allbert3@allbert.com.br", Senha = "123" });
context.Pessoa.AddOrUpdate(
    p => p.Nome,
    new Models.Pessoa { Id = 1, Nome = "Allbert", UsuarioId = 1 });
```

- Update-Database -Verbose

Adicionar UsuarioVM

```
public class Cadastro
{
    [Required]
    [MaxLength(255)]
    public string Nome { get; set; }
    [Required]
    [EmailAddress]
    public string Email { get; set; }
    [DataType(DataType.Password)]
    [RegularExpression("(?=.*\\d)(?=.*[a-z])(?=.*[A-Z]).{6,12})", ErrorMessage = "A senha deve conter aos
menos uma letra maiúscula, minúscula e um número. Deve ser no mínimo 6 caracteres")]
    public string Senha { get; set; }
    [DataType(DataType.Password)]
    [Compare("Senha")]
    [Display(Name = "Confirma Senha")]
    public string ConfirmaSenha { get; set; }
}
public class Acesso
{
    [Required]
    [EmailAddress]
    public string Email { get; set; }
    [DataType(DataType.Password)]
    [RegularExpression("(?=.*\\d)(?=.*[a-z])(?=.*[A-Z]).{6,12})", ErrorMessage = "A senha deve conter aos
menos uma letra maiúscula, minúscula e um número. Deve ser no mínimo 6 caracteres")]
    public string Senha { get; set; }
}
```

Adicionar UsuarioController

- Adicionar exibição (view) usando o modelo Create, da classe Cadastro;

Adicionar CadastroController

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Cadastro(Cadastro cad) {
    if (ModelState.IsValid)
    {
        Usuario usu = new Usuario();
        usu.Email = cad.Email;
        usu.Senha = Funcoes.HashTexto(cad.Senha, "SHA512");
        db.Usuario.Add(usu);
        Pessoa pes = new Pessoa();
        pes.Usuario = usu;
        pes.Nome = cad.Nome;
        db.Pessoa.Add(pes);
        db.SaveChanges();
        return RedirectToAction("Index", "Home");
    }
    return View(cad);
}
```