

LAB21: ASP.NET Identity

Neste laboratório é apresentado o modelo básico de utilização do *ASP.NET Identity*.

Exercício 1 –Modelo básico de autenticação

1. Dentro do Visual Studio crie um novo projeto: **File | NewProject | ASP.NET Web Application**. Utilize o nome "Lab19" para o nome do projeto. Clique **OK**.
2. Clique **Change Authentication** garanta que a opção **Individual User Accounts** esteja selecionada. Clique **OK**.
3. Abra a pasta **Account** e note que as operações básicas de autenticação já estão disponíveis.
4. Neste *template* modelo de autenticação é implementado utilizando o *Entity Framework* com *Code First*, ou seja, as tabelas de autenticação serão geradas automaticamente pelo .NET ao executar a aplicação (ao utilizar uma operação que necessite dos dados de autenticação).
5. Antes de gerar o banco de dados de autenticação, acrescente uma propriedade para armazenar o nome do usuário ao profile (note que o usuário é identificado pelo seu *email*, que é armazenado na propriedade *UserName*. Não vamos alterar este padrão). Abra o arquivo **IdentityModels.cs** na pasta **Models** e acrescente uma propriedade **Nome** à classe **ApplicationUser**:

```
publicString Nome { get; set; }
```

6. Abra o form **Register.aspx** na pasta **Account** acrescente componentes para realizar a entrada do nome do usuáriologo após o *Email*:

```
<divclass="form-group">
<asp:Labelrunat="server"AssociatedControlID="TxtNome"CssClass="col-md-2 control-label">User name</asp:Label>
<divclass="col-md-10">
<asp:TextBoxrnat="server"ID="TxtNome"CssClass="form-control"/>
<asp:RequiredFieldValidatorrunat="server"ControlToValidate="TxtNome"
CssClass="text-danger"ErrorMessage="The name field is required."/>
</div>
</div>
```

7. Para finalizar, acrescente esta propriedade na inicialização do objeto que será persistido. No code-behind, encontre o método **CreateUser_Click** e altere a instanciação do objeto **ApplicationUser** para:

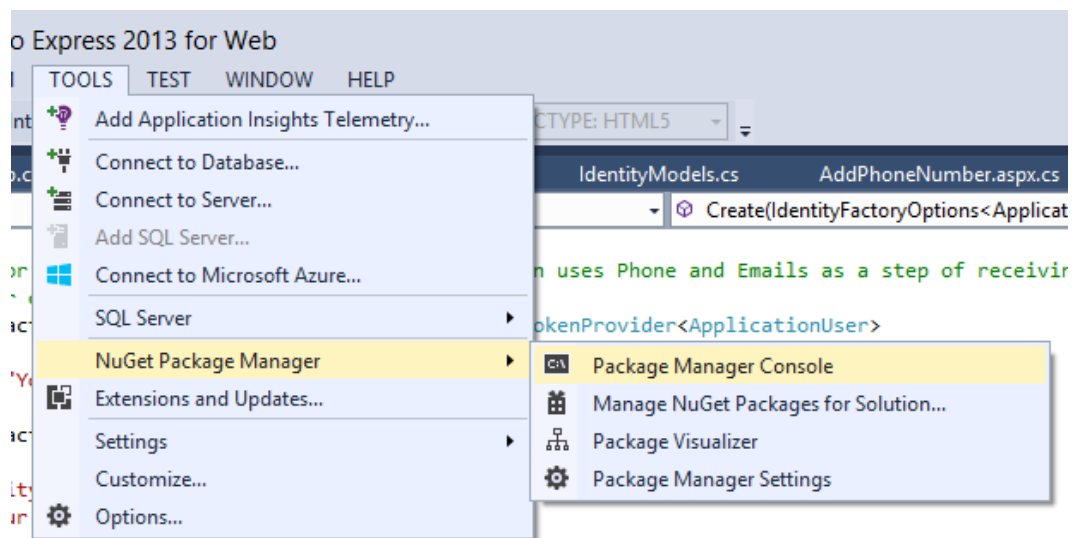
```
var user = newApplicationUser() { UserName = Email.Text,
Email = Email.Text,
Nome=TxtNome.Text };
```

8. Abra o arquivo **IdentityConfig.cs**, na pasta **App_Start** (caso o arquivo não esteja visível, selecione a opção **Show All Files**), e analise as regras de formação de senhas (no método **Create**). Altere-as, se desejar.
9. Execute a aplicação e crie um usuário (utilize a opção *Register* no menu). Após criar um usuário note a mudança no modelo de apresentação das opções no menu principal. Feche a aplicação e veja como isto é implementado no arquivo de layout da aplicação (**Site.master**).

- Abra a pasta **App_Data** e selecione a opção **Show All Files**, abra o arquivo gerado e analise as tabelas geradas para autenticação. Visualize os dados do seu usuário na tabela **AspNetUsers**.

Exercício 2 – Alterações no profile – Migrations

- Acrescente um nova propriedade, de nome **Empresa**, para armazenar também a o local de trabalho do usuário. Para isto siga os passos acima descritos para adicionar informações ao profile do usuário.
- Execute a aplicação e note que não funciona, pois o banco de dados de autenticação já foi gerado.
- Após a geração das tabelas de autenticação não é mais possível adicionar ou alterar as informações de profile do usuário. Para isto será necessário configurar e utilizar o **Entity Framework Migrations**. Inicialmente abra a console do gerenciador de pacotes:



- Na console, execute os seguintes comandos:
 - Enable-Migrations (habilita o *Migrations*)
 - Add-Migration "Empresa" (cria um ponto de sincronização)
 - Update-Database (atualiza as tabelas)
- Execute a aplicação novamente e crie um novo usuário (desta vez deverá funcionar).

Exercício 3 – Acesso às informações de profile

- Abra a página default da aplicação (**Default.aspx**) e acrescente um label de nome **LbIMensagem**, logo no início da página.
- No code-behind da página acrescente o código abaixo:

```
protectedvoid Page_Load(object sender, EventArgs e)
{
    if (!User.Identity.IsAuthenticated)
        LblMensagem.Text = "Usuario não autenticado";
    else
    {
        var currentUserId = User.Identity.GetUserId();

        var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        var currentUser = manager.FindById(currentUserId);

        LblMensagem.Text = currentUser.Nome + " - " + currentUser.Empresa;
    }
}
```

3. Será necessário acrescentar os seguintes namespaces:

```
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
```

4. Execute a aplicação e teste.
5. Retome o laboratório anterior e altere os métodos de remoção, edição e criação de álbuns para que apenas usuários logados possam realizar as operações.