**How to Create New Feature (Sub System)**

**Version 1.1**

در این مثال، می‌خواهیم یک Feature به نام Hrm ایجاد نماییم

* در پروژه Domain، یک پوشه به نام Hrm ایجاد می‌کنیم.
* در داخل پوشه Hrm، یک کلاس به نام UserSite ایجاد می‌کنیم.
* از آن‌جایی که رابطه بین کلاس User موجود در پوشه Identity، و این کلاس (UserSite)، یک رابطه یک به چند می‌باشد، Navigation Property های لازم را در این دو کلاس ایجاد می‌کنیم.

**نکته:** برای تمام Navigation Property ها، چه آن‌هایی که یک هستند و چه آن‌هایی که چند (IList) هستند، Set آن‌ها را به صورت Private تعریف می‌کنیم.

**نکته:** تمام Navigation Property های چند (IList) را، درConstructor، حتماً New می‌کنیم!

* در پروژه Persistence، و در کلاس DatabaseContext، یک Property از جنس DbSet، برای این کلاس (UserSite) ایجاد می‌کنیم.
* در پروژه Persistence، و در پوشه Features، و در پوشه Identity، و در کلاس  UserConfiguration، ارتباط یک به چند بین User و UserSite را تعریف می‌کنیم.
* در پروژه Persistence، و در پوشه Features، یک پوشه به نام Hrm ایجاد می‌کنیم.
* در پوشه Hrm، یک کلاس به نام UserSiteConfiguration ایجاد می‌کنیم.
* در صورتی که بخواهیم برای جدول UserSite، اطلاعات اولیه و یا تستی (در زمان ایجاد بانک اطلاعاتی) ایجاد نماییم، در پروژه Persistence، و در پوشه Extensions، کلاس ModelBuilderExtensions را ویرایش می‌کنیم. که البته در این مثال خاص، نیازی به اطلاعات اولیه و تستی نداریم!
* بانک اطلاعاتی را حذف می‌کنیم و پروژه را یک‌بار اجرا می‌کنیم تا ببینیم که آیا در ایجاد بانک اطلاعاتی و ایجاد جداول مشکلی وجود دارد یا خیر؟
* در پروژه Server، و در پوشه Pages، و در پوشه Features، پوشه‌ای به نام Hrm ایجاد می‌کنیم.
* در داخل این پوشه، اگر عملیات مربوط به CRUD، مربوط به Admin (مسئولین پایگاه) باشد، پوشه‌ای به نام Admin ایجاد می‌کنیم و اگر عملیات مربوط به CRUD، مربوط به کاربران سایت (کاربر معمولی/کاربر ویژه) باشد، پوشه‌ای به نام User ایجاد می‌کنیم. در این مثال، از آن‌جایی که مدل UserSite، صرفاً مربوط به کاربران سایت بوده و آن‌ها باید سایت‌های خود را در سامانه مدیریت نمایند، پوشه‌ای به نام User ایجاد می‌کنیم.
* در داخل پوشه User، پوشه‌ای (با نام جمع) به نام UserSites ایجاد می‌کنیم.
* در داخل پوشه UserSites، اقدام به ایجاد صفحات مربوط به مدیریت سایت‌های کاربر می‌کنیم:

Create.cshtml

Delete.cshtml

Details.cshtml

Index.cshtml

Update.cshtml

* قبل از این که اقدام به تکمیل صفحات فوق نماییم، باید ViewModel های مربوط به صفحات فوق را در پوشه ViewModels ایجاد نماییم.
* **نکته:** باید همیشه به یاد داشته باشیم که ساختار پوشه‌بندی در پروژه ViewModels، باید بر اساس پوشه‌بندی در پروژه اصلی باشد!
* برای این منظور وارد پروژه ViewModels شده، وارد پوشه Features شده، و مجدداً پوشه‌ای به نام Hrm ایجاد می‌کنیم.
* با توجه به پوشه‌بندی پروژه Server، در پروژه ViewModels، و در داخل پوشه Hrm، پوشه‌ای به نام User ایجاد می‌کنیم.
* در پوشه Hrm، پوشه‌ای به نام User ایجاد می‌کنیم.
* در پوشه User، پوشه‌ای به نام UserSites ایجاد می‌کنیم.
* در پوشه UserSites، فایل‌ها/کلاس‌های ذیل را ایجاد می‌کنیم:
* CreateViewModel.cs
* DetailsOrDeleteViewModel.cs
* IndexItemViewModel.cs
* UpdateViewModel.cs
* مجدداً وارد پروژه Server می‌شویم و ابتدا صفحه Index را با توجه به ViewModel ایجاد شده، ایجاد می‌کنیم.
* قبل از آن که بقیه صفحات را ایجاد نماییم، منوی مربوط به صفحه مدیریت سایت‌های کاربر را ایجاد می‌کنیم، پروژه را یک‌بار اجرا می‌کنیم تا از وجود گزینه منو، و نیز اوکی بودن صفحه Index (فهرست سایت‌های کاربر)، اطمینان حاصل کنیم.
* برای ایجاد منو، در پروژه Server، وارد پوشه Pages شده، وارد پوشه Features شده، وارد پوشه Identity شده، وارد پوشه PartialViews شده، وارد فایل Navbar.cshtml\_ می‌شویم و یک منو برای کاربر ایجاد می‌کنیم:

<li><hr class="dropdown-divider" /></li>  
<li>  
 <a class="dropdown-item" asp-page="/Features/Hrm/User/UserSites/Index">  
 @(Resources.DataDictionary.ManageOf)  
 @(Resources.DataDictionary.Sites)  
 </a>  
</li>

* حال اقدام به ایجاد صفحه Create می‌کنیم. پس از ایجاد این صفحه نیز، مجدداً پروژه را اجرا می‌کنیم تا از اوکی بودن این صفحه، اطمینان حاصل کنیم.

**نکته:** با توجه به این که ممکن است که در مدل مربوطه، بحث مربوط به Unique بودن فیلد(هایی) مطرح باشد، باید همیشه حواسمان به تکراری بودن آن فیلدها در زمان ایجاد و ویرایش باشد! و پیغام مناسبی را به کاربر اعلام نماییم. در خصوص این مثال، نشانی سایت (SiteUrl) برای هر شخص، منحصر به فرد می‌باشد، و باید در زمان ایجاد و یا ویرایش، حواسمان به این موضوع باشد!

**نکته مهم:** در همه صفحاتی که ایجاد می‌کنیم، باید حواسمان به Permission صفحه مورد نظر باشد! در این مثال، از آن‌جایی که هر کاربری، بدون در نظر گرفتن هیچ پیش‌شرطی، می‌تواند سایت‌های خودش را مدیریت نماید، لذا همه صفحات (نمایش/ایجاد/ویرایش/حذف/فهرست) را با دسترسی Simple User تنظیم می‌کنیم:

[Infrastructure.Security.CustomAuthorize(minRoleCode:  
 Domain.Features.Identity.Enums.RoleEnum.SimpleUser)]