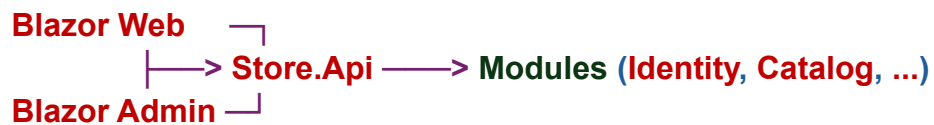


اسکلت بندی پروژه

Shop.sln



یه تصویر ذهنی درست از اتصال فرانت ها به بکند :



توضیح ساده هر بخش

1. Store.Api

Backend اصلی:

- API
 - Auth entry point
 - اتصال ماژولها
 - Redis / DB / DI
-

2. Store.Web

Blazor فروشگاه برای کاربر:

- دیدن محصولات
 - سبد خرید
 - سفارش
-

3. Store.Admin

Blazor پنل مدیریت:

- مدیریت محصول
 - سفارشها
 - کاربران
-

4. Store.Modules (مهمترین بخش)

این همون قلب سیستمه.

هر ماژول:

- Identity
- Catalog

Basket •

Ordering •

Payment •

نکته خیلی مهم (همون چیزی که پرسیدی)

آیا این ماژول‌ها باید جدا Project باشن؟

✓بله (در معماری تمیز)

✗ولی Microservice نه

یعنی:

• جدا Project برای نظم ✓

• ولی همه داخل یک Solution و یک Deploy ✓

چرا این ساختار خوبه؟

چون:

✓هم Clean Architecture داری

✓هم Modular هستی

✓هم Overengineering نکردی

✓هم 100 user k رو جواب میدی (با Redis + DB درست)

جواب مستقیم به سوال اصلیت

? آیا Identity لازمه؟

✓بله، همیشه لازمه

حتی اگر:

• ASP.NET Core Identity

• JWT built-in

استفاده کنی

چون هنوز اینو داری:

• User

- Role
- Permission
- Session

? آیا Basket / Ordering / Payment باید جدا باشن؟

✓بله، ولی فقط در سطح ماژول

نه Service جدا

مهمترین نکته‌ای که باید بفهمی

این دو تا فرق دارن:

✗ اشتباه:

هر ماژول = پروژه + سرویس جدا + دیتابیس جدا

✓درست برای تو:

هر ماژول = پروژه جدا داخل یک سیستم واحد