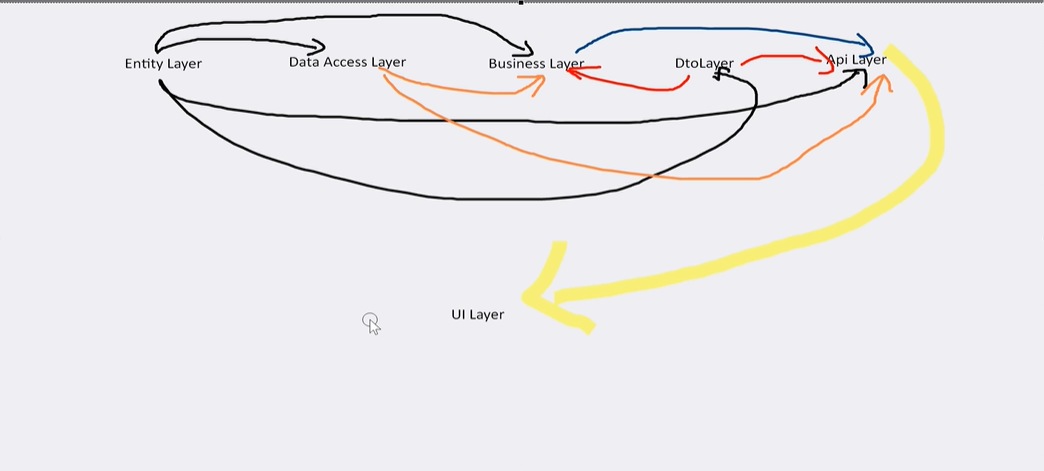
New project -> empty solution.

3 tane dosya

1. Apiconsume -> new project - > asp .net core api -> HotelProject.WebApi (location: C:\Users\Suhrob\source\repos\Core\_Hotel\_API\ApiConsume) icine kaydet diyecez
2. Apiconsume -> new project - > classLibrary -> HotelProject.EntityLayer -> ApiConsume diye location degistirecez ->
3. DataAccessLayer (aynisi)
4. BusinessLayer (aynisi)
5. Frontend -> new project -> asp .net core Web app (Model-View-Controller) -> HotelProject.WebUI ->location = C:\Users\Suhrob\source\repos\Core\_Hotel\_API\Frontend
6. Solution explorer -> manage nuger pakage -> proejelin hepsini secib 4 tane kutuphane yukleyicez (Core,Design,SqlServer,Tools)
7. Referanslari ekleme ->
8. 
9. DataAccessLayer icine Context class olustur icine ->

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer("server=DESKTOP-1GEE0JQ;database=CoreApiDb;integrated security=true;TrustServerCertificate=True;");

}

public DbSet<Room> Rooms { get; set; }

public DbSet<Staff> Staffs { get; set; }

public DbSet<Service> Services { get; set; }

public DbSet<Subscribe> Subscribes { get; set; }

public DbSet<Testimonial> Testimonials { get; set; }

Baglisyoruz ve veri tabana yansitiyoruz.

Swagger calisa bilmesi icin swagger kutuphanesini indirmemiz lazim.

Program.cs tede bazi degisikler olacak.

builder.Services.AddEndpointsApiExplorer(); //api falan yazicak onu silecez

builder.Services.AddSwaggerGen();

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

1. HotelProject.WebApi icine controllerei tanimlayicaz:

namespace HotelProject.WebApi.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class RoomController : ControllerBase

{

private readonly IRoomService \_roomService;

public RoomController(IRoomService roomService)

{

\_roomService = roomService;

}

[HttpGet]

public IActionResult RoomList()

{

var values = \_roomService.TGetList();

return Ok(values);

}

[HttpPost]

public IActionResult AddRoom(Room p)

{

\_roomService.TInsert(p);

return Ok();

}

[HttpDelete]

public IActionResult DeleteRoom(int id)

{

var values = \_roomService.TGetById(id);

\_roomService.TDelete(values);

return Ok(values);

}

[HttpPut]

public IActionResult UpdateRoom(Room p)

{

\_roomService.TGetById(p.RoomId);

return Ok();

}

[HttpGet("{id}")]

public IActionResult GetRoom(int id)

{

var values = \_roomService.TGetById(id);

return Ok(values);

}

1. }
2. Tanimlarken api controller sememiz lazim.
3. Calisa bilmesi isin asagdaki kodlari hepsi icin ayri ayri yazmak lazim.

builder.Services.AddDbContext<Context>();

builder.Services.AddScoped<IStaffDal,EfStaffDal>();

builder.Services.AddScoped<IStaffService,StaffManager>();

builder.Services.AddScoped<IServiceDal, EfServiceDal>();

builder.Services.AddScoped<IServiceService, ServiceManager>();

builder.Services.AddScoped<IRoomDal, EfRoomDal>();

builder.Services.AddScoped<IRoomService,RoomManager>();

builder.Services.AddScoped<ISubscribeDal, EfSubscribeDal>();

builder.Services.AddScoped<ISubscribeService, SubscribeManager>();

builder.Services.AddScoped<ITestimonialDal, EfTestimonialDal>();

1. builder.Services.AddScoped<ITestimonialService, TestimonialManager>();
2. Temani yukeyip PartialView boldukten sonra Program.cs se Cors eklemek lazim.

builder.Services.AddCors(opt =>

{

opt.AddPolicy("OtelApiCors", policy =>

{

policy.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod();

});

});

app.UseCors("OtelApiCors");

app.UseAuthorization(); // bu zaten vardi ustune ekleyicez

1. IdentityUser kutuphanesini indirdikten sonra
2. AppUser.cs class icine ->

public class AppUser:IdentityUser<int>// deyince hazir proportiler gelecek

{

public string Name { get; set; } = string.Empty;

public string Surname { get; set; } = string.Empty;

public string ImageUrl { get; set; } = string.Empty;

public bool Status { get; set; }

}

Biz sadece bize lazim olan proportilri ekleye bilriz.

1. AppRoles.cs olsturacaz

public class AppRole:IdentityRole<int>

{

}

Burada role tanimlaya bilriz

1. Context.cs

//DbContext asagda DbContext vardi onun yerine IdentityDbContext yazdim

public class Context:IdentityDbContext<AppUser, AppRole, int>

{

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer("server=DESKTOP-1GEE0JQ;database=CoreApiDb;integrated security=true;TrustServerCertificate=True;");

}

public DbSet<Room> Rooms { get; set; }

public DbSet<Staff> Staffs { get; set; }

public DbSet<Service> Services { get; set; }

public DbSet<Subscribe> Subscribes { get; set; }

public DbSet<Testimonial> Testimonials { get; set; }

}

Boyle degisti AppUser tablosu AspNetUsers tablosuna yansidi

1. Temaniin yiklenmesi
2. Default controller index not use
3. ViewComponents -> Default -> \_HeadPartial.cs ->

// \_HeadPartial adında bir ViewComponent sınıfı tanımlanıyor

// ViewComponent'ler genellikle reusable UI component'leri oluşturmak için kullanılır

public class \_HeadPartial : ViewComponent // ViewComponent sınıfından türetiliyor

{

// ViewComponent'in ana metodudur - Invoke metodu otomatik olarak çağrılır

public IViewComponentResult Invoke()

{

return View(); // ViewComponent'in varsayılan view'ını döndürür

}

}}

1. Views -> Shared -> Components (olusturacaz) -> \_HeadPartial (olusturracaz) -> Default.cshtml (Partial View olusturacaz)

Icinede @await Component.InvokeAsync("\_HeadPartial")

Buradaki aldigimiz html kodlarin yerine yazicaz.