Document Rapid Api - Api Consume

✅ **Konu Tanımı**  
API (**Application Programming Interface**) yazılımların birbiriyle iletişim kurmasını sağlayan arayüzdür. Bir uygulamanın diğerinin verilerini veya işlevlerini kullanmasına izin verir.

🧠 **Mantığı**  
API, farklı yazılımların “nasıl konuşacağını” tanımlar. Böylece geliştiriciler, sistemin iç detaylarını bilmeden belirli işlevlere erişebilir. Bu, hem güvenlik hem de hız sağlar.

📍 **Nerede ve Nasıl Kullanılır?**

* Farklı sistemlerin veri alışverişi yapması gerektiğinde (örneğin, web sitesi ile banka sistemi arasında ödeme işlemi)
* Mobil uygulamanın sunucudan veri çekmesi
* Üçüncü parti servislerin entegrasyonu (örneğin, Google Maps API ile harita ekleme)

📌 **Günlük Hayattan Basit Bir Örnek**  
Bir restoranda yemek siparişi verirken garson, sizin mutfakla doğrudan konuşmadan siparişinizi alıp iletir.

* **Siz** → Kullanıcı
* **Garson** → API
* **Mutfak** → Sunucu / Sistem

💻 **C# Kod Örneği**  
Aşağıdaki örnek, bir web API’den JSON veri çekip ekrana yazdırır:

using System;

using System.Net.Http;

using System.Threading.Tasks;

class Program

{

static async Task Main()

{

HttpClient client = new HttpClient();

string url = "https://api.exchangerate-api.com/v4/latest/USD";

string json = await client.GetStringAsync(url);

Console.WriteLine(json);

}

}

Yukarıdaki kod, bir döviz kuru API’sinden USD bazlı tüm kurları JSON formatında getirir.

🔁 **Alternatifler ve Kıyaslama**

* **API**: Sadece arayüz tanımlar, veriyi nereden ve nasıl alacağınızı bilir.
* **Database Connection**: Veriye doğrudan erişim sağlar, ancak API kadar güvenli ve soyut değildir.
* **Web Scraping**: Veriyi HTML’den çeker, API kadar güvenilir veya hızlı değildir.

📚 **Terim Sözlüğü**

| **İngilizce Terim** | **Türkçe Açıklama** |
| --- | --- |
| API (Application Programming Interface) | Uygulama Programlama Arayüzü |
| HTTP | Web üzerinden veri iletişim protokolü |
| JSON | Veri alışveriş formatı (JavaScript Object Notation) |
| HttpClient | C#’ta HTTP isteği göndermeye yarayan sınıf |
| Endpoint | API’nin erişim noktası |
| Request | İstek |
| Response | Yanıt |

✅ **Konu Tanımı**  
DTO (**Data Transfer Object**), veriyi bir katmandan diğerine taşımak için kullanılan basit veri taşıma nesnesidir. İçinde genellikle **property**’ler bulunur, iş mantığı içermez.

🧠 **Mantığı**

* Amaç: **Veriyi taşırken gereksiz bilgileri ayıklamak** ve sadece ihtiyaç duyulan alanları göndermek.
* Katmanlı mimaride, **Entity** (veritabanı modeli) doğrudan kullanıcıya gönderilmez; önce DTO ile sadeleştirilir.
* Güvenlik, hız ve bağımsızlık sağlar.

📍 **Nerede ve Nasıl Kullanılır?**

* **API** cevaplarında, veritabanı modelini olduğu gibi göndermemek için.
* Birden fazla tablonun birleşiminden elde edilen özel verileri taşımak için.
* Sadece gerekli alanların dışa açılması gerektiğinde.

📌 **Günlük Hayattan Basit Bir Örnek**  
Bir kargo şirketinden adres bilgisi istenir.

* **Veritabanında**: Ad, soyad, TC kimlik no, adres, telefon, ödeme bilgileri vb.
* **DTO ile gönderilen**: Ad, soyad, adres (TC kimlik ve ödeme bilgisi gizli kalır).

💻 **C# Kod Örneği**

// Entity (Veritabanı Modeli)

public class User

{

public int Id { get; set; }

public string FullName { get; set; }

public string Email { get; set; }

public string PasswordHash { get; set; } // Güvenlik gereği DTO'da olmayacak

}

// DTO (Data Transfer Object)

public class UserDto

{

public string FullName { get; set; }

public string Email { get; set; }

}

// Kullanım Örneği

public UserDto MapToDto(User user)

{

return new UserDto

{

FullName = user.FullName,

Email = user.Email

};

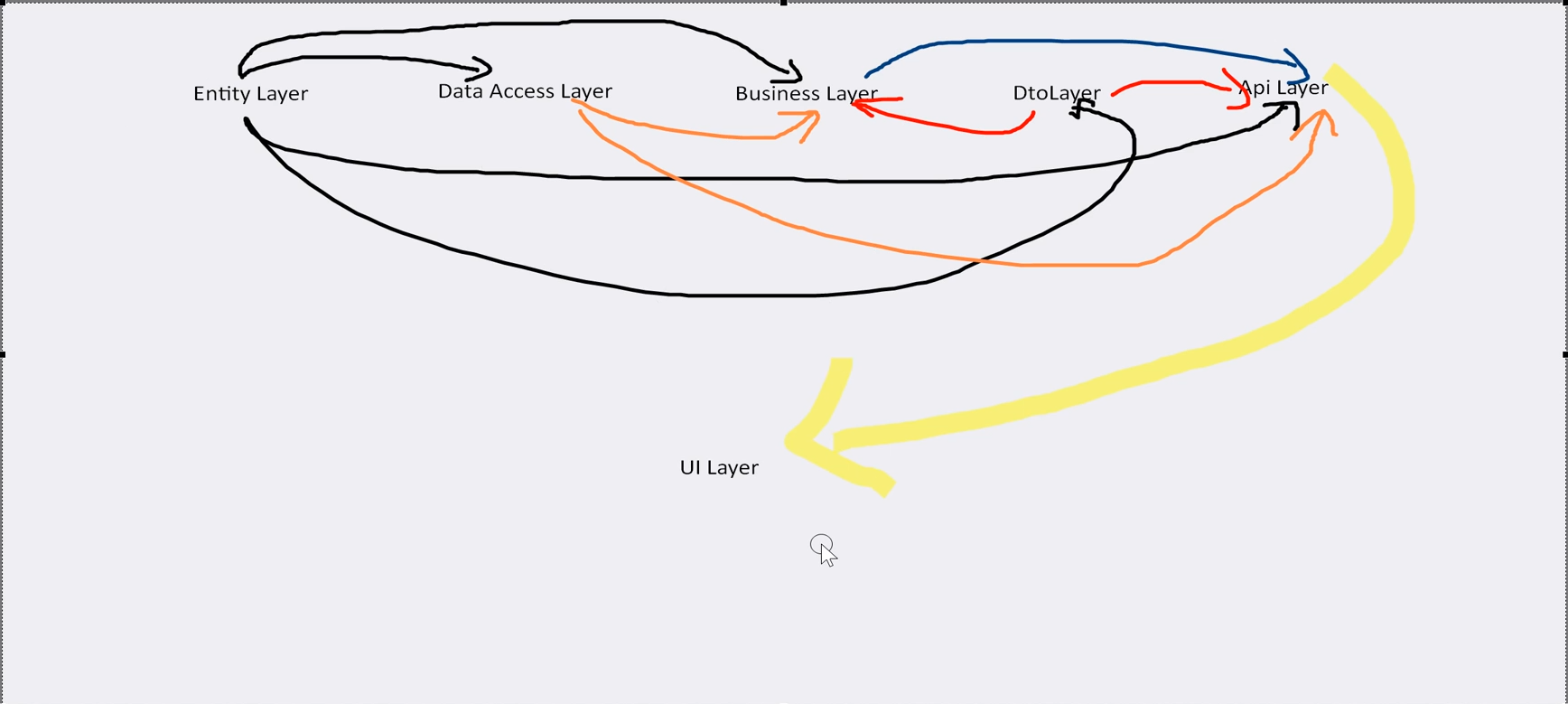
}

🔁 **Alternatifler ve Kıyaslama**

* **Entity**: Veritabanı modeli, tüm alanlar vardır.
* **DTO**: Sadece taşınacak veya dışa açılacak alanlar vardır.
* **ViewModel**: Genelde UI (görsel arayüz) için özelleştirilmiş veri modelidir.

📚 **Terim Sözlüğü**

| **İngilizce Terim** | **Türkçe Açıklama** |
| --- | --- |
| DTO (Data Transfer Object) | Veri Taşıma Nesnesi |
| Entity | Veritabanı tablosunu temsil eden nesne |
| Property | Sınıfın özellikleri |
| ViewModel | Arayüz için özel veri modeli |
| Mapping | İki model arasında veri dönüştürme işlemi |



namespace HotelProject.WebApi.Controllers

{

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

[ApiController]

public class RoomController : ControllerBase

{

[HttpGet] // listelemek için

public IActionResult RoomList()

{

return Ok(); // Burada genellikle bir liste döndürülür, örneğin: return Ok(roomList); return view gibi

}

[HttpPost] // eklemek için

public IActionResult RoomAdd()

{

return Ok();

}

[HttpDelete] // silmek için

public IActionResult RoomDelete()

{

return Ok();

}

[HttpPut] // güncellemek için

public IActionResult RoomUpdate()

{

return Ok();

}

[HttpGet("{id}")] // İdye göre getirmek için

public IActionResult GetRoom()

{

return Ok();

}

}

}

Swashbuckle.AspNetCore ve api çalıştırma

Kur bu önemli ardından

{

"$schema": "https://json.schemastore.org/launchsettings.json",

"profiles": {

"http": {

"commandName": "Project",

"dotnetRunMessages": true,

"launchBrowser": true,

"applicationUrl": "http://localhost:5023",

"launchUrl": "swagger/",

"environmentVariables": {

"ASPNETCORE\_ENVIRONMENT": "Development"

}

}

}

}

Ve

namespace HotelProject.WebApi

{

public class Program

{

public static void Main(string[] args)

{

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

// Learn more about configuring OpenAPI at https://aka.ms/aspnet/openapi

builder.Services.AddOpenApi();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.MapOpenApi();

app.UseSwagger();

app.UseSwaggerUI();

}

app.UseAuthorization();

app.MapControllers();

app.Run();

}

}

}

Örnek içi dolu hali

namespace HotelProject.WebApi.Controllers

{

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

[ApiController]

public class RoomController : ControllerBase

{

private readonly IRoomService \_roomService;

public RoomController(IRoomService roomService)

{

\_roomService = roomService;

}

[HttpGet] // listelemek için

public IActionResult RoomList()

{

var values = \_roomService.TGetList();

return Ok(values); // Burada genellikle bir liste döndürülür, örneğin: return Ok(StaffList); return view gibi

}

[HttpPost] // eklemek için

public IActionResult RoomAdd(Room room)

{

// Burada genellikle bir model alırsınız ve onu eklemek için kullanırsınız.

// Örneğin:

\_roomService.TInsert(room);

return Ok();

}

[HttpDelete("{id}")] // silmek için

public IActionResult RoomDelete(int id)

{

// Silme işlemi için genellikle bir id alırsınız.

var result = \_roomService.TGetById(id);

if (result is null)

{

return NotFound();

}

\_roomService.TDelete(result);

return Ok();

}

[HttpPut] // güncellemek için

public IActionResult RoomUpdate(Room room)

{

// Güncelleme işlemi için genellikle bir model alırsınız.

\_roomService.TUpdate(room);

return Ok();

}

[HttpGet("{id}")] // İdye göre getirmek için

public IActionResult GetRoom(int id)

{

// İdye göre bir staff bilgisi almak için genellikle bir id alırsınız.

var result = \_roomService.TGetById(id);

return Ok(result);

}

}

}

✅ **Konu Tanımı**  
CORS (**Cross-Origin Resource Sharing**), bir web uygulamasının farklı bir domain (alan adı) üzerindeki API’ye güvenli şekilde istek göndermesine izin veren tarayıcı güvenlik mekanizmasıdır.

🧠 **Mantığı**  
Tarayıcılar, güvenlik nedeniyle **farklı domain’ler** arası API isteklerini varsayılan olarak engeller.

* CORS, API tarafında ayarlanan izinler ile “hangi domain’lerin” veri alabileceğini belirtir.
* Böylece sadece güvenilir kaynaklar API’ye erişebilir.

📍 **Nerede ve Nasıl Kullanılır?**

* Frontend (React, Angular, Vue) ile Backend (C#, Java, Node.js) farklı sunucularda çalışıyorsa.
* Örn: http://localhost:3000 (Frontend) → https://api.site.com (Backend) istek atarken.
* API tarafında **AllowOrigins, AllowMethods, AllowHeaders** ayarları ile yönetilir.

📌 **Günlük Hayattan Basit Bir Örnek**  
Bir apartmana giriş yapmak için güvenlik görevlisi sadece **izinli kişilerin** girmesine izin verir.

* Apartman = API
* Güvenlik = CORS
* Ziyaretçi listesi = AllowOrigins listesi

💻 **C# Kod Örneği** (ASP.NET Core)

builder.Services.AddCors(options =>

{

options.AddPolicy("OtelApiCors",

builder => builder.AllowAnyOrigin()

.AllowAnyMethod()

.AllowAnyHeader());

});

app.UseCors("OtelApiCors");

//app useauthorization hemen üstüne

Yukarıdaki örnekte sadece OtelApiCors adresinden gelen API isteklerine izin veriliyor.

🔁 **Alternatifler ve Kıyaslama**

* **CORS**: Tarayıcı bazlı güvenlik kontrolü, API erişim izni sağlar.
* **API Key / JWT**: Kimlik doğrulama yapar ama CORS yerine geçmez.
* **Reverse Proxy**: Farklı domain’leri tek bir noktadan sunarak CORS ihtiyacını azaltabilir.

✅ **Konu Tanımı**  
IHttpClientFactory, **HttpClient** nesnelerini merkezi olarak yönetmek ve verimli şekilde oluşturmak için kullanılan ASP.NET Core servisidir.  
Amaç: **HttpClient** kullanımındaki performans ve kaynak yönetimi problemlerini çözmek.

🧠 **Mantığı**  
Normalde her istek için new HttpClient() oluşturmak:

* Fazla bellek tüketir.
* TCP bağlantılarının hızlı kapanmaması nedeniyle **socket exhaustion** (bağlantı tükenmesi) hatası verir.

IHttpClientFactory:

* HttpClient örneklerini **pool** (havuz) mantığıyla yönetir.
* Gereksiz bağlantı açılmasını engeller.
* Konfigürasyonları merkezi olarak tanımlar.

📍 **Nerede ve Nasıl Kullanılır?**

* API çağrılarında HttpClient yönetimini optimize etmek istediğinde.
* Birden fazla API ile çalışıyorsan ve farklı ayar/config istiyorsan.
* **Dependency Injection (DI)** ile kullanılır.

📌 **Günlük Hayattan Basit Bir Örnek**  
Bir şirketin kargo gönderimi için kendi araç filosu var.

* **Her sipariş için yeni araç satın almak** = new HttpClient() → Masraflı.
* **Mevcut araç filosunu düzenli kullanmak** = IHttpClientFactory → Daha verimli.

💻 **C# Kod Örneği**

private readonly IHttpClientFactory \_httpClientFactory;

public StaffController(IHttpClientFactory httpClientFactory)

{

\_httpClientFactory = httpClientFactory;

}

public async Task<IActionResult> Index()

{

var client = \_httpClientFactory.CreateClient();

var responseMessage = await client.GetAsync("http://localhost:5023/api/Staff");

if (responseMessage.IsSuccessStatusCode)

{

var jsonData = await responseMessage.Content.ReadAsStringAsync();

var values = JsonConvert.DeserializeObject<List<StaffViewModel>>(jsonData);

return View(values);

}

return View();

}

Burada IHttpClientFactory, HttpClient nesnesini **havuzdan** alır ve yönetir.

Program.cs de önemli

builder.Services.AddHttpClient();

ÖRNEK CONTROLLER

private readonly IHttpClientFactory \_httpClientFactory;

public StaffController(IHttpClientFactory httpClientFactory)

{

\_httpClientFactory = httpClientFactory;

}

public async Task<IActionResult> Index()

{

var client = \_httpClientFactory.CreateClient();

var responseMessage = await client.GetAsync("http://localhost:5023/api/Staff");

if (responseMessage.IsSuccessStatusCode)

{

var jsonData = await responseMessage.Content.ReadAsStringAsync();

var values = JsonConvert.DeserializeObject<List<StaffViewModel>>(jsonData); //jsondan dönüştür

return View(values);

}

return View();

}

[HttpGet]

public IActionResult StaffInsert()

{

return View();

}

[HttpPost]

public async Task<IActionResult> StaffInsert(AddStaffViewModel model)

{

var client = \_httpClientFactory.CreateClient();

var jsonData = JsonConvert.SerializeObject(model); //jsona dönüştür

StringContent stringContent = new StringContent(jsonData, Encoding.UTF8, "application/json"); //dönüşüm işlemi uyguladık

var responseMessage = await client.PostAsync("http://localhost:5023/api/Staff", stringContent);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return View();

}

public async Task<IActionResult> StaffDelete(int id)

{

var client = \_httpClientFactory.CreateClient();

var responseMessage = await client.DeleteAsync($"http://localhost:5023/api/Staff/{id}");

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return View();

}

[HttpGet]

public async Task<IActionResult> StaffUpdate(int id)

{

var client = \_httpClientFactory.CreateClient();

var responseMessage = await client.GetAsync($"http://localhost:5023/api/Staff/{id}");

if (responseMessage.IsSuccessStatusCode)

{

var jsonData = await responseMessage.Content.ReadAsStringAsync();

var values = JsonConvert.DeserializeObject<UpdateStaffViewModel>(jsonData);

return View(values);

}

return View();

}

[HttpPost]

public async Task<IActionResult> StaffUpdate(UpdateStaffViewModel model)

{

var client = \_httpClientFactory.CreateClient();

var jsonData = JsonConvert.SerializeObject(model);

StringContent stringContent = new StringContent(jsonData, Encoding.UTF8, "application/json");

var responseMessage = await client.PutAsync("http://localhost:5023/api/Staff/", stringContent);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return View();

}

AUTO MAPPER



Bu iki paket kurulacak aynı sürüm olmazsa hata verir dependecy injection paketi sürüm dışı diyor ama bilmiyorum kurmadığımız zaman hata veriyor sonra dto sınıfları oluşturuyoruz bir klasörün altında

Update ve add için aşağıda add ve update örneği vardır

public class RoomAddDto

{

[Required(ErrorMessage ="Oda Numarasını Giriniz")]

public string? RoomNumber { get; set; }

public string? RoomCoverImage { get; set; }

[Required(ErrorMessage = "Lütfen Fiyat bilgisi Giriniz")]

public int RoomPrice { get; set; }

[Required(ErrorMessage = "Oda Başlığı Giriniz")]

public string? RoomTitle { get; set; }

[Required(ErrorMessage = "Oda Yatak Sayısı Giriniz")]

public string? BedCount { get; set; }

[Required(ErrorMessage = "Oda Banyo Sayısı Giriniz")]

public string? BathCount { get; set; }

public string? Wifi { get; set; }

public string? RoomDescription { get; set; }

}

public int RoomID { get; set; }

[Required(ErrorMessage = "Oda Numarasını Giriniz")]

public string? RoomNumber { get; set; }

public string? RoomCoverImage { get; set; }

[Required(ErrorMessage = "Lütfen Fiyat bilgisi Giriniz")]

public int RoomPrice { get; set; }

[Required(ErrorMessage = "Oda Başlığı Giriniz")]

public string? RoomTitle { get; set; }

[Required(ErrorMessage = "Oda Yatak Sayısı Giriniz")]

public string? BedCount { get; set; }

[Required(ErrorMessage = "Oda Banyo Sayısı Giriniz")]

public string? BathCount { get; set; }

public string? Wifi { get; set; }

public string? RoomDescription { get; set; }

Sonra ui içerisinde bir mappin klasörü açıyoruz ve config ayarları yapıyoruz örnek olması adına profileden inheritance olması önemli 2 şekilde yapılır örnek mevcut ardından program cs var

public class AutoMapperConfig:Profile

{

public AutoMapperConfig()

{

CreateMap<RoomAddDto, Room>();

CreateMap<Room, RoomAddDto>();

CreateMap<RoomUpdateDto, Room>().ReverseMap(); //yukarıdaki 2 adımda yaptığımız kodu tek adımda yapıyoruz

}

}

Ardından program.cs de bunu yazmamız gerekir

builder.Services.AddAutoMapper(typeof(Program).Assembly);

ve son olarak api controller ekle güncelle listele olarak

public class RoomDtoController : ControllerBase

{

private readonly IRoomService \_roomService;

private readonly IMapper \_mapper;

public RoomDtoController(IRoomService roomService, IMapper mapper)

{

\_roomService = roomService;

\_mapper = mapper;

}

[HttpGet]

public IActionResult Index()

{

var values = \_roomService.TGetList();

return Ok(values);

}

[HttpPost]

public IActionResult AddRoom(RoomAddDto roomAddDto)

{

if (!ModelState.IsValid)

{

return BadRequest();

}

var values = \_mapper.Map<Room>(roomAddDto);

\_roomService.TInsert(values);

return Ok();

}

[HttpPut]

public IActionResult UpdateRoom(RoomUpdateDto roomUpdateDto)

{

if (!ModelState.IsValid)

{

return BadRequest();

}

var values = \_mapper.Map<Room>(roomUpdateDto);

\_roomService.TUpdate(values);

return Ok("Başarıyla Güncellendi");

}

}

Buraya kadar ki kısım swagger kısmıydı peki UI

Hemen hemen aynı bir dto klasörü ve her bir entity için ayri bir klasör oluştur ve her klasör içinde yapılacak işlemleri belirle örnek olması için

namespace HotelProject.WebUI.Dtos.ServiceDto

{

public class ResultServiceDto

{

public int ServiceID { get; set; }

public string? ServiceIcon { get; set; }

public string? ServiceTitle { get; set; }

public string? ServiceDescription { get; set; }

}

}

Ve

using System.ComponentModel.DataAnnotations;

namespace HotelProject.WebUI.Dtos.ServiceDto

{

public class CreateDtoService

{

[Required(ErrorMessage ="Hizmet İkon Linki Giriniz")]

public string? ServiceIcon { get; set; }

[Required(ErrorMessage = "Hizmet Başlığı Giriniz")]

public string? ServiceTitle { get; set; }

[Required(ErrorMessage = "Hizmet Açıklaması Giriniz")]

public string? ServiceDescription { get; set; }

}

}

Ardından bir adette configurasyon klasörü oluşturup configureleri gir

public class AutoMapperConfig:Profile

{

public AutoMapperConfig()

{

CreateMap<ResultServiceDto, Service>().ReverseMap();

CreateMap<CreateDtoService, Service>().ReverseMap();

CreateMap<UpdateDtoService, Service>().ReverseMap();

}

}

Tabi burası içinde automapper kütüphaneleri unutmamak gerekir yine program cs de aynısı olacak

builder.Services.AddAutoMapper(typeof(Program));

viewda rahatca tanımlamak için viewimportsda tanımla

@using HotelProject.WebUI

@using HotelProject.WebUI.Models

@using HotelProject.WebUI.Models.Staff

@using HotelProject.WebUI.Models.Testimonial

@using HotelProject.WebUI.Dtos.ServiceDto

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

Böylelikle view da

@model List< ResultServiceDto>

@{

ViewData["Title"] = "Index";

Layout = "~/Views/AdminLayout/\_AdminLayout.cshtml";

int count = 0;

}

<div class="col-lg-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<h4 class="card-title">Hizmet Tablosu</h4>

<div class="table-responsive">

<table class="table text-center align-middle">

<thead>

<tr>

<th class="col-1">#</th>

<th class="col-3">Hizmet Başlığı</th>

<th class="col-2">Sil</th>

<th class="col-2">Güncelle</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

<td>@(++count)</td>

<td>@item.ServiceTitle</td>

<td><a href="/Service/ServiceDelete/@item.ServiceID" class="btn btn-outline-danger btn-lg btn-rounded btn-fw">Sil</a></td>

<td><a href="/Service/ServiceUpdate/@item.ServiceID" class="btn btn-outline-warning btn-lg btn-rounded btn-fw">Güncelle</a></td>

</tr>

}

</tbody>

</table>

<br />

<a href="/Service/ServiceInsert/" class="btn btn-outline-primary btn-lg btn-rounded btn-fw">Yeni Hizmet Ekle</a>

</div>

</div>

</div>

</div>

@model List< ResultServiceDto> rahatça böyle tanımlayabilirsin

controller kısmı ise

public class ServiceController : Controller

{

private readonly IHttpClientFactory \_httpClientFactory;

public ServiceController(IHttpClientFactory httpClientFactory)

{

\_httpClientFactory = httpClientFactory;

}

public async Task<IActionResult> Index()

{

var client = \_httpClientFactory.CreateClient();

var responseMessage = await client.GetAsync("http://localhost:5023/api/Service");

if (responseMessage.IsSuccessStatusCode)

{

var jsonData = await responseMessage.Content.ReadAsStringAsync();

var values = JsonConvert.DeserializeObject<List<ResultServiceDto>>(jsonData);

return View(values);

}

return View();

}

[HttpGet]

public IActionResult ServiceInsert()

{

return View();

}

[HttpPost]

public async Task<IActionResult> ServiceInsert(UpdateDtoService updateDto)

{

if (ModelState.IsValid)

{

var client = \_httpClientFactory.CreateClient();

var jsonData = JsonConvert.SerializeObject(updateDto);

StringContent stringContent = new StringContent(jsonData, Encoding.UTF8, "application/json");

var responseMessage = await client.PostAsync("http://localhost:5023/api/Service", stringContent);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

}

return View();

}

İşte dto çıkarmak bu kadar basit

ÖRNEK BİR GİRİŞ YAPTIRMA KODLARI

private readonly UserManager<AppUser> \_userManager;

public RegisterController(UserManager<AppUser> userManager)

{

\_userManager = userManager;

}

[HttpGet]

public IActionResult Index()

{

return View();

}

[HttpPost]

public async Task<IActionResult> Index(CreateNewUserDto userDto)

{

if (!ModelState.IsValid)

{

return View();

}

var appUser = new AppUser

{

Name = userDto.Name,

SurName = userDto.SurName,

Email = userDto.Mail,

UserName = userDto.UserName,

};

var result = await \_userManager.CreateAsync(appUser, userDto.Password);

if (result.Succeeded)

{

return RedirectToAction("Index", "Login");

}

return View();

}

Örnek view

@model CreateNewUserDto

@{

Layout = null;

}

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Hotel Kayıt ol</title>

<!-- plugins:css -->

<**link** rel="stylesheet" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/vendors/mdi/css/materialdesignicons.min.css">

<**link** rel="stylesheet" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/vendors/css/vendor.bundle.base.css">

<!-- endinject -->

<!-- Plugin css for this page -->

<!-- End plugin css for this page -->

<!-- inject:css -->

<!-- endinject -->

<!-- Layout styles -->

<**link** rel="stylesheet" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/css/style.css">

<!-- End layout styles -->

<**link** rel="shortcut icon" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/images/favicon.png" />

</head>

<body>

<div class="container-scroller">

<div class="container-fluid page-body-wrapper full-page-wrapper">

<div class="row w-100 m-0">

<div class="content-wrapper full-page-wrapper d-flex align-items-center auth login-bg">

<div class="card col-lg-4 mx-auto">

<div class="card-body px-5 py-5">

<h3 class="card-title text-left mb-3">Otel Kayıt Sistemi</h3>

<**form** method="post">

<div class="form-group">

<label>Ad</label>

<**input** **type**="text" class="form-control p\_input" **asp-for**="Name">

</div>

<div class="form-group">

<label>Soyad</label>

<**input** **type**="text" class="form-control p\_input" **asp-for**="SurName">

</div>

<div class="form-group">

<label>Kullanıcı Adı</label>

<**input** **type**="text" class="form-control p\_input" **asp-for**="UserName">

</div>

<div class="form-group">

<label>Email</label>

<**input** **type**="email" class="form-control p\_input" **asp-for**="Mail">

</div>

<div class="form-group">

<label>Şifre</label>

<**input** **type**="password" class="form-control p\_input" **asp-for**="Password">

</div>

<div class="form-group">

<label>Şifre</label>

<**input** **type**="password" class="form-control p\_input" **asp-for**="ConfirmPassword">

</div>

<div class="text-center">

<button type="submit" class="btn btn-primary btn-block enter-btn">Kayıt ol</button>

</div>

<p class="sign-up text-center">Zaten Bir hesabın varmı?<a href="/Login/Index/"> Giriş Yap</a></p>

</**form**>

</div>

</div>

</div>

<!-- content-wrapper ends -->

</div>

<!-- row ends -->

</div>

<!-- page-body-wrapper ends -->

</div>

<!-- container-scroller -->

<!-- plugins:js -->

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/vendors/js/vendor.bundle.base.js"></**script**>

<!-- endinject -->

<!-- Plugin js for this page -->

<!-- End plugin js for this page -->

<!-- inject:js -->

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/off-canvas.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/hoverable-collapse.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/misc.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/settings.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/todolist.js"></**script**>

<!-- endinject -->

</body>

</html>

Örnek giriş ekranı

private readonly SignInManager<AppUser> \_signInManager;

public LoginController(SignInManager<AppUser> signInManager)

{

\_signInManager = signInManager;

}

[HttpGet]

public IActionResult Index()

{

return View();

}

[HttpPost]

public async Task<IActionResult> Index(LoginUserDto userDto)

{

if (ModelState.IsValid)

{

var result = await \_signInManager.PasswordSignInAsync(userDto.Username, userDto.Password, false, false);

if (result.Succeeded)

{

return RedirectToAction("Index", "Staff");

}

else

{

return View();

}

}

return View();

}

Ve

View

@model LoginUserDto

@{

Layout = null;

}

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Hotel Giriş Yap</title>

<!-- plugins:css -->

<**link** rel="stylesheet" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/vendors/mdi/css/materialdesignicons.min.css">

<**link** rel="stylesheet" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/vendors/css/vendor.bundle.base.css">

<!-- endinject -->

<!-- Plugin css for this page -->

<!-- End plugin css for this page -->

<!-- inject:css -->

<!-- endinject -->

<!-- Layout styles -->

<**link** rel="stylesheet" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/css/style.css">

<!-- End layout styles -->

<**link** rel="shortcut icon" href="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/images/favicon.png" />

</head>

<body>

<div class="container-scroller">

<div class="container-fluid page-body-wrapper full-page-wrapper">

<div class="row w-100 m-0">

<div class="content-wrapper full-page-wrapper d-flex align-items-center auth login-bg">

<div class="card col-lg-4 mx-auto">

<div class="card-body px-5 py-5">

<h3 class="card-title text-left mb-3">Otel Giriş Sistemi</h3>

<**form** method="post">

<div class="form-group">

<label>Ad</label>

<**input** **type**="text" class="form-control p\_input" **asp-for**="Username">

</div>

<div class="form-group">

<label>Şifre</label>

<**input** **type**="password" class="form-control p\_input" **asp-for**="Password">

</div>

<div class="text-center">

<button type="submit" class="btn btn-primary btn-block enter-btn">Giriş Yap</button>

</div>

<p class="sign-up text-center"> Bir hesabın yokmu?<a href="/Register/Index/"> Kayıt ol</a></p>

</**form**>

</div>

</div>

</div>

<!-- content-wrapper ends -->

</div>

<!-- row ends -->

</div>

<!-- page-body-wrapper ends -->

</div>

<!-- container-scroller -->

<!-- plugins:js -->

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/vendors/js/vendor.bundle.base.js"></**script**>

<!-- endinject -->

<!-- Plugin js for this page -->

<!-- End plugin js for this page -->

<!-- inject:js -->

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/off-canvas.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/hoverable-collapse.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/misc.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/settings.js"></**script**>

<**script** src="~/corona-free-dark-bootstrap-admin-template-1.0.0/template/assets/js/todolist.js"></**script**>

<!-- endinject -->

</body>

</html>

VE DTO SINIFLARINI VE VİEWIMPORTA EKLEMEYİ UNUTMA ONLAR YUKARIDA ANLATILDI

Rapid api bize bir çok api sunuyor onu kopyaladık viewmodele yapıştırmak için paste pastespecial ve json formatında yapıştır dersek bizim için verir

{

"status": true,

"message": "Success",

"timestamp": 1755242606230,

"data": {

"exchange\_rates": [

{

"exchange\_rate\_buy": "0.06775776",

"currency": "WST"

},

{

"exchange\_rate\_buy": "98.20554024",

"currency": "KHR"

},

{

"currency": "BIF",

"exchange\_rate\_buy": "72.32303564"

},

{

"currency": "BGN",

"exchange\_rate\_buy": "0.04111738"

},

{

"exchange\_rate\_buy": "2.14660480",

"currency": "BTN"

},

{

"exchange\_rate\_buy": "0.16939697",

"currency": "BOB"

},

{

"exchange\_rate\_buy": "2.96566950",

"currency": "SSP"

},

{

"exchange\_rate\_buy": "0.43164359",

"currency": "ZAR"

},

{

"exchange\_rate\_buy": "1.67531384",

"currency": "AFN"

},

{

"currency": "BYR",

"exchange\_rate\_buy": "626.61281669"

},

{

"exchange\_rate\_buy": "0.02451476",

"currency": "PAB"

},

{

"exchange\_rate\_buy": "0.02451476",

"currency": "BSD"

},

{

"currency": "RON",

"exchange\_rate\_buy": "0.10639895"

},

{

"exchange\_rate\_buy": "0.63966231",

"currency": "HNL"

},

{

"exchange\_rate\_buy": "0.98071263",

"currency": "UYU"

},

{

"exchange\_rate\_buy": "0.23485873",

"currency": "SEK"

},

{

"currency": "VES",

"exchange\_rate\_buy": "3.29265221"

},

{

"currency": "SOS",

"exchange\_rate\_buy": "14.01019633"

},

{

"currency": "TND",

"exchange\_rate\_buy": "0.07047870"

},

{

"currency": "LKR",

"exchange\_rate\_buy": "7.38079289"

},

{

"exchange\_rate\_buy": "5.88966367",

"currency": "YER"

},

{

"currency": "LBP",

"exchange\_rate\_buy": "2194.45302714"

},

{

"currency": "VND",

"exchange\_rate\_buy": "643.68830374"

},

{

"exchange\_rate\_buy": "308.43397887",

"currency": "UZS"

},

{

"exchange\_rate\_buy": "10.34258726",

"currency": "KMF"

},

{

"currency": "BRL",

"exchange\_rate\_buy": "0.13237847"

},

{

"exchange\_rate\_buy": "0.37801755",

"currency": "MVR"

},

{

"exchange\_rate\_buy": "0.36142842",

"currency": "SCR"

},

{

"exchange\_rate\_buy": "32.02848959",

"currency": "ARS"

},

{

"currency": "GBP",

"exchange\_rate\_buy": "0.01808340"

},

{

"currency": "USD",

"exchange\_rate\_buy": "0.02451476"

},

{

"exchange\_rate\_buy": "0.04902951",

"currency": "BZD"

},

{

"exchange\_rate\_buy": "0.10194707",

"currency": "PGK"

},

{

"currency": "GHS",

"exchange\_rate\_buy": "0.26414650"

},

{

"currency": "CLP",

"exchange\_rate\_buy": "23.50340988"

},

{

"currency": "PLN",

"exchange\_rate\_buy": "0.08958305"

},

{

"currency": "CAD",

"exchange\_rate\_buy": "0.03380830"

},

{

"exchange\_rate\_buy": "1.39623784",

"currency": "PHP"

},

{

"exchange\_rate\_buy": "0.84806084",

"currency": "BEF"

},

{

"currency": "UAH",

"exchange\_rate\_buy": "1.01613659"

},

{

"currency": "MOP",

"exchange\_rate\_buy": "0.19777525"

},

{

"currency": "ANG",

"exchange\_rate\_buy": "0.04388141"

},

{

"exchange\_rate\_buy": "3.49791019",

"currency": "ESP"

},

{

"currency": "HRK",

"exchange\_rate\_buy": "0.15839597"

},

{

"exchange\_rate\_buy": "2.50869451",

"currency": "XPF"

},

{

"exchange\_rate\_buy": "0.79648458",

"currency": "THB"

},

{

"exchange\_rate\_buy": "1.56612476",

"currency": "MZN"

},

{

"currency": "STN",

"exchange\_rate\_buy": "0.51826398"

},

{

"exchange\_rate\_buy": "0.03770051",

"currency": "AUD"

},

{

"exchange\_rate\_buy": "2102286.00000000",

"currency": "VEB"

},

{

"currency": "MZM",

"exchange\_rate\_buy": "1451.85497238"

},

{

"currency": "FRF",

"exchange\_rate\_buy": "0.13790135"

},

{

"exchange\_rate\_buy": "0.51494471",

"currency": "CZK"

},

{

"exchange\_rate\_buy": "3.49791019",

"currency": "ADP"

},

{

"exchange\_rate\_buy": "13.79009374",

"currency": "XAF"

},

{

"exchange\_rate\_buy": "1.18406295",

"currency": "EGP"

},

{

"exchange\_rate\_buy": "0.18786394",

"currency": "GTQ"

},

{

"currency": "SLL",

"exchange\_rate\_buy": "566.95954693"

},

{

"exchange\_rate\_buy": "3.16730496",

"currency": "KES"

},

{

"exchange\_rate\_buy": "0.40878357",

"currency": "MDL"

},

{

"exchange\_rate\_buy": "418.94898366",

"currency": "ZMK"

},

{

"exchange\_rate\_buy": "0.32893566",

"currency": "EEK"

},

{

"exchange\_rate\_buy": "37.58511817",

"currency": "NGN"

},

{

"currency": "SKK",

"exchange\_rate\_buy": "0.63333447"

},

{

"currency": "STD",

"exchange\_rate\_buy": "503.66219454"

},

{

"currency": "DEM",

"exchange\_rate\_buy": "0.04111781"

},

{

"currency": "BWP",

"exchange\_rate\_buy": "0.32686345"

},

{

"currency": "CUP",

"exchange\_rate\_buy": "0.58835419"

},

{

"exchange\_rate\_buy": "0.17585048",

"currency": "CNY"

},

{

"exchange\_rate\_buy": "0.03147205",

"currency": "SGD"

},

{

"exchange\_rate\_buy": "4.91392682",

"currency": "LRD"

},

{

"currency": "PKR",

"exchange\_rate\_buy": "6.92664749"

},

{

"exchange\_rate\_buy": "0.01477559",

"currency": "LVL"

},

{

"exchange\_rate\_buy": "87.22454568",

"currency": "UGX"

},

{

"currency": "JMD",

"exchange\_rate\_buy": "3.92787666"

},

{

"currency": "SYP",

"exchange\_rate\_buy": "4.87321626"

},

{

"exchange\_rate\_buy": "0.08925823",

"currency": "QAR"

},

{

"exchange\_rate\_buy": "0.08707519",

"currency": "PEN"

},

{

"exchange\_rate\_buy": "0.07258823",

"currency": "LTL"

},

{

"currency": "NZD",

"exchange\_rate\_buy": "0.04135069"

},

{

"exchange\_rate\_buy": "7.16354653",

"currency": "GRD"

},

{

"currency": "MYR",

"exchange\_rate\_buy": "0.10326841"

},

{

"currency": "ZMW",

"exchange\_rate\_buy": "0.56874231"

},

{

"exchange\_rate\_buy": "107.95347643",

"currency": "MGA"

},

{

"exchange\_rate\_buy": "3.20437211",

"currency": "HTG"

},

{

"currency": "LSL",

"exchange\_rate\_buy": "0.43164359"

},

{

"exchange\_rate\_buy": "8.31192769",

"currency": "HUF"

},

{

"exchange\_rate\_buy": "0.08028583",

"currency": "BYN"

},

{

"currency": "ETB",

"exchange\_rate\_buy": "3.44873372"

},

{

"exchange\_rate\_buy": "0.02451476",

"currency": "BMD"

},

{

"exchange\_rate\_buy": "0.04388141",

"currency": "XCG"

},

{

"exchange\_rate\_buy": "2.46324211",

"currency": "RSD"

},

{

"exchange\_rate\_buy": "0.22962974",

"currency": "TJS"

},

{

"currency": "XCD",

"exchange\_rate\_buy": "0.06618984"

},

{

"currency": "JOD",

"exchange\_rate\_buy": "0.01735767"

},

{

"exchange\_rate\_buy": "0.09004025",

"currency": "AED"

},

{

"currency": "SDD",

"exchange\_rate\_buy": "15.54450877"

},

{

"exchange\_rate\_buy": "0.03147205",

"currency": "BND"

},

{

"exchange\_rate\_buy": "0.73441305",

"currency": "TWD"

},

{

"currency": "CVE",

"exchange\_rate\_buy": "2.32635853"

},

{

"exchange\_rate\_buy": "0.19201428",

"currency": "HKD"

},

{

"exchange\_rate\_buy": "0.09198672",

"currency": "SAR"

},

{

"exchange\_rate\_buy": "14.75940942",

"currency": "SDG"

},

{

"currency": "MUR",

"exchange\_rate\_buy": "1.11272510"

},

{

"currency": "LUF",

"exchange\_rate\_buy": "0.84806084"

},

{

"exchange\_rate\_buy": "8.13908903",

"currency": "MRO"

},

{

"exchange\_rate\_buy": "0.43164359",

"currency": "SZL"

},

{

"exchange\_rate\_buy": "0.00902511",

"currency": "MTL"

},

{

"exchange\_rate\_buy": "0.46036263",

"currency": "MXN"

},

{

"exchange\_rate\_buy": "0.12499605",

"currency": "FIM"

},

{

"exchange\_rate\_buy": "3.43456909",

"currency": "NPR"

},

{

"currency": "TTD",

"exchange\_rate\_buy": "0.16533365"

},

{

"exchange\_rate\_buy": "0.00001191",

"currency": "XAU"

},

{

"exchange\_rate\_buy": "1.95505110",

"currency": "RUB"

},

{

"exchange\_rate\_buy": "0.21450412",

"currency": "SVC"

},

{

"exchange\_rate\_buy": "63.49398973",

"currency": "TZS"

},

{

"currency": "SBD",

"exchange\_rate\_buy": "0.20210023"

},

{

"exchange\_rate\_buy": "42.51594637",

"currency": "MWK"

},

{

"currency": "SRD",

"exchange\_rate\_buy": "0.91949962"

},

{

"exchange\_rate\_buy": "13.79009374",

"currency": "XOF"

},

{

"currency": "ILS",

"exchange\_rate\_buy": "0.08290278"

},

{

"currency": "SRG",

"exchange\_rate\_buy": "72.02569549"

},

{

"currency": "MKD",

"exchange\_rate\_buy": "1.29388884"

},

{

"exchange\_rate\_buy": "5989.41880342",

"currency": "VEF"

},

{

"exchange\_rate\_buy": "3.01053400",

"currency": "ISK"

},

{

"currency": "ITL",

"exchange\_rate\_buy": "40.70568873"

},

{

"exchange\_rate\_buy": "0.01978709",

"currency": "CHF"

},

{

"exchange\_rate\_buy": "4.35646449",

"currency": "DJF"

},

{

"exchange\_rate\_buy": "177.27346319",

"currency": "PYG"

},

{

"exchange\_rate\_buy": "2.14660480",

"currency": "INR"

},

{

"exchange\_rate\_buy": "212.22350091",

"currency": "GNF"

},

{

"exchange\_rate\_buy": "0.02010210",

"currency": "KYD"

},

{

"exchange\_rate\_buy": "0.01230427",

"currency": "CYP"

},

{

"currency": "BAM",

"exchange\_rate\_buy": "0.04111738"

},

{

"exchange\_rate\_buy": "0.04632797",

"currency": "NLG"

},

{

"exchange\_rate\_buy": "0.00924329",

"currency": "BHD"

},

{

"currency": "IEP",

"exchange\_rate\_buy": "0.01655640"

},

{

"currency": "TRL",

"exchange\_rate\_buy": "420457.20000000"

},

{

"currency": "AOA",

"exchange\_rate\_buy": "22.69160028"

},

{

"exchange\_rate\_buy": "0.05866178",

"currency": "TOP"

},

{

"currency": "COP",

"exchange\_rate\_buy": "99.36597816"

},

{

"exchange\_rate\_buy": "9.40679058",

"currency": "AMD"

},

{

"currency": "CRC",

"exchange\_rate\_buy": "12.36682471"

},

{

"exchange\_rate\_buy": "529.80997984",

"currency": "LAK"

},

{

"currency": "MAD",

"exchange\_rate\_buy": "0.22063649"

},

{

"currency": "XPD",

"exchange\_rate\_buy": "0.00001286"

},

{

"currency": "KWD",

"exchange\_rate\_buy": "0.00749294"

},

{

"exchange\_rate\_buy": "0.00002136",

"currency": "XPT"

},

{

"currency": "BDT",

"exchange\_rate\_buy": "2.97854380"

},

{

"exchange\_rate\_buy": "1.78393859",

"currency": "GMD"

},

{

"currency": "FJD",

"exchange\_rate\_buy": "0.05569385"

},

{

"exchange\_rate\_buy": "3.18316453",

"currency": "DZD"

},

{

"exchange\_rate\_buy": "5.03791341",

"currency": "SIT"

},

{

"exchange\_rate\_buy": "0.25041580",

"currency": "NOK"

},

{

"exchange\_rate\_buy": "0.04902951",

"currency": "BBD"

},

{

"currency": "ALL",

"exchange\_rate\_buy": "2.04538940"

},

{

"exchange\_rate\_buy": "1.50888304",

"currency": "DOP"

},

{

"currency": "KRW",

"exchange\_rate\_buy": "33.87942371"

},

{

"currency": "XAG",

"exchange\_rate\_buy": "0.00094404"

},

{

"exchange\_rate\_buy": "70.62709131",

"currency": "CDF"

},

{

"currency": "AWG",

"exchange\_rate\_buy": "0.04388141"

},

{

"currency": "GYD",

"exchange\_rate\_buy": "5.12848998"

},

{

"currency": "RWF",

"exchange\_rate\_buy": "35.41169337"

},

{

"exchange\_rate\_buy": "0.15689567",

"currency": "DKK"

},

{

"currency": "OMR",

"exchange\_rate\_buy": "0.00943818"

},

{

"exchange\_rate\_buy": "87.98384532",

"currency": "MNT"

},

{

"currency": "KZT",

"exchange\_rate\_buy": "13.20514818"

},

{

"currency": "NIO",

"exchange\_rate\_buy": "0.89783575"

},

{

"exchange\_rate\_buy": "0.04112350",

"currency": "BGL"

},

{

"currency": "ROL",

"exchange\_rate\_buy": "1039.70623145"

},

{

"exchange\_rate\_buy": "0.02102286",

"currency": "EUR"

},

{

"currency": "ATS",

"exchange\_rate\_buy": "0.28928175"

},

{

"exchange\_rate\_buy": "27.04183067",

"currency": "IQD"

},

{

"exchange\_rate\_buy": "394.79549296",

"currency": "IDR"

},

{

"currency": "MRU",

"exchange\_rate\_buy": "0.97862904"

},

{

"exchange\_rate\_buy": "2.92252877",

"currency": "VUV"

},

{

"currency": "PTE",

"exchange\_rate\_buy": "4.21470415"

},

{

"currency": "AZM",

"exchange\_rate\_buy": "124.52087899"

},

{

"currency": "AZN",

"exchange\_rate\_buy": "0.04167509"

},

{

"currency": "JPY",

"exchange\_rate\_buy": "3.61567911"

},

{

"exchange\_rate\_buy": "0.13295088",

"currency": "LYD"

},

{

"exchange\_rate\_buy": "841.92470965",

"currency": "IRR"

},

{

"exchange\_rate\_buy": "2.14207068",

"currency": "KGS"

},

{

"exchange\_rate\_buy": "0.06618984",

"currency": "GEL"

},

{

"currency": "NAD",

"exchange\_rate\_buy": "0.43164359"

},

{

"exchange\_rate\_buy": "2719.64553687",

"currency": "GHC"

},

{

"exchange\_rate\_buy": "373.93916756",

"currency": "TMM"

},

{

"currency": "MMK",

"exchange\_rate\_buy": "51.54051337"

},

{

"currency": "TRY",

"exchange\_rate\_buy": "1.00000000"

}

],

"base\_currency": "TRY",

"base\_currency\_date": "2025-08-14"

}

}

Bu yapı

public class Rootobject

{

public bool status { get; set; }

public string message { get; set; }

public long timestamp { get; set; }

public Data data { get; set; }

}

public class Data

{

public Exchange\_Rates[] exchange\_rates { get; set; }

public string base\_currency { get; set; }

public string base\_currency\_date { get; set; }

}

public class Exchange\_Rates

{

public string exchange\_rate\_buy { get; set; }

public string currency { get; set; }

}

Bu yapıya dönüştü.

using System.Net.Http.Headers;

var client = new HttpClient();

var request = new HttpRequestMessage

{

Method = HttpMethod.Get,

RequestUri = new Uri("https://booking-com15.p.rapidapi.com/api/v1/meta/getExchangeRates?base\_currency=TRY"),

Headers =

{

{ "x-rapidapi-key", "94e4eaa144msha687f3ef56acd9dp1adad8jsn3d0b1a394640" },

{ "x-rapidapi-host", "booking-com15.p.rapidapi.com" },

},

};

using (var response = await client.SendAsync(request))

{

response.EnsureSuccessStatusCode();

var body = await response.Content.ReadAsStringAsync();

Console.WriteLine(body);

}

Bize zaten bu veriliyor

using Microsoft.AspNetCore.Mvc;

using Newtonsoft.Json;

using RapidApi.Consume.Models;

using System.Net.Http.Headers;

using System.Threading.Tasks;

namespace RapidApi.Consume.Controllers

{

public class ExchangeController : Controller

{

public async Task<IActionResult> Index()

{

var client = new HttpClient();

var request = new HttpRequestMessage

{

Method = HttpMethod.Get,

RequestUri = new Uri("https://booking-com15.p.rapidapi.com/api/v1/meta/getExchangeRates?base\_currency=TRY"),

Headers =

{

{ "x-rapidapi-key", "94e4eaa144msha687f3ef56acd9dp1adad8jsn3d0b1a394640" },

{ "x-rapidapi-host", "booking-com15.p.rapidapi.com" },

},

};

using (var response = await client.SendAsync(request))

{

response.EnsureSuccessStatusCode();

var body = await response.Content.ReadAsStringAsync();

var root = JsonConvert.DeserializeObject<ExchangeViewModel.Rootobject>(body);

var exchangeRates = root?.data?.exchange\_rates?.ToList() ?? new List<ExchangeViewModel.Exchange\_Rates>();

return View(exchangeRates);

}

}

}

}

Sonra view

@model List<ExchangeViewModel.Exchange\_Rates>

@{

Layout = null;

int count = 0;

}

<**link** href="~/lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />

<div class="container">

<br />

<table class="table table-bordered ">

<tr>

<th>#</th>

<th>Döviz Kodu</th>

<th>Birim Fiyatı</th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>@{

count++;

}

@count</td>

<td>@item.currency</td>

<td>@item.exchange\_rate\_buy</td>

</tr>

}

</table>

</div>

JWT İŞLEMLERİ



Paketi kur ardından program cs de bu kodları yaz

builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme).AddJwtBearer(opt =>

{

opt.RequireHttpsMetadata = false; // https üzerinden gelmeyen token'lar? kabul et

opt.TokenValidationParameters = new TokenValidationParameters() // token do?rulama parametreleri

{

ValidIssuer = "https://localhost", // bu tokeni kim üretiyor

ValidAudience = "https://localhost", // bu tokeni kim kullanabilir

IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("aspnetcoreapiapi")), // bunla ?ifrele

ValidateIssuerSigningKey = true, // imzay? do?rula,

ValidateLifetime = true, // token süresini do?rula

};

});

Ardından modelsde bir sınıf aç ve bu kodu yaz

namespace WebApiJwt.Models

{

public class CreateToken

{

public void TokenCreate()

{

var bytes = Encoding.UTF8.GetBytes("aspnetcoreapiapi"); // şifreleme anahtarını belirleyin

SymmetricSecurityKey key = new SymmetricSecurityKey(bytes); // Anahtar oluşturma

SigningCredentials credentials = new SigningCredentials(key, SecurityAlgorithms.HmacSha256); // İmzalama kimlik bilgilerini oluşturma ve algoritmasını belirleme

JwtSecurityToken token = new JwtSecurityToken(issuer: "https://localhost", audience: "https://localhost", notBefore: DateTime.Now, expires: DateTime.Now.AddMinutes(3), signingCredentials: credentials);

// Token oluşturma alıcı ve verici bilgilerini, geçerlilik süresini ve imzalama kimlik bilgilerini belirleme ne zaman geçerli olacağını belirleme ve ne zaman sona ereceğini belirleme

JwtSecurityTokenHandler handler = new JwtSecurityTokenHandler(); // Token işleyici oluşturma

handler.WriteToken(token); // Token'ı yazma

}

}

}

MAİL İŞLEMLERİ



Kurulumunu yap

Gmaile git ve 2 adımlı doğrulamayı aç ve A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Yukarıdaki işlemleri yap ve şifreyi al sonra kod kısmında ise

Ve model sınıfı

namespace HotelProject.WebUI.Models.Mail

{

public class AdminMailViewModel

{

public string? Name { get; set; }

public string? SenderMail { get; set; }

public string? ReceiverMail { get; set; }

public string? Subject { get; set; }

public string? Body { get; set; }

}

}

using HotelProject.WebUI.Models.Mail;

using MailKit.Net.Smtp;

using Microsoft.AspNetCore.Mvc;

using MimeKit;

namespace HotelProject.WebUI.Controllers

{

public class AdminMailController : Controller

{

[HttpGet]

public IActionResult Index()

{

return View();

}

[HttpPost]

public IActionResult Index( AdminMailViewModel adminMail)

{

MimeMessage mimeMessage = new MimeMessage();

MailboxAddress mailboxAddressFrom = new MailboxAddress("Yazılım Hotel", "ebtuyou1@gmail.com");

mimeMessage.From.Add(mailboxAddressFrom);

MailboxAddress mailboxAddressTo = new MailboxAddress("User", adminMail.ReceiverMail);

mimeMessage.To.Add(mailboxAddressTo);

var bodyBuilder = new BodyBuilder();

bodyBuilder.TextBody = adminMail.Body;

mimeMessage.Body = bodyBuilder.ToMessageBody();

mimeMessage.Subject = adminMail.Subject;

SmtpClient client = new SmtpClient();

client.Connect("smtp.gmail.com", 587, false);

client.Authenticate("ebtuyou1@gmail.com", "kurzpwyuuaybvfzf");

client.Send(mimeMessage);

client.Disconnect(true);

return View();

}

}

}

Ve view kısmı

@model AdminMailViewModel

@{

ViewData["Title"] = "Index";

Layout = "~/Views/AdminLayout/\_AdminLayout.cshtml";

}

<**form** class="forms-sample" method="post">

<div class="form-group mb-3">

<label >Alıcı Mail</label>

<**input** **type**="text" class="form-control" **asp-for**="ReceiverMail" placeholder="Alıcı mail giriniz">

</div>

<div class="form-group mb-3">

<label>Konu</label>

<**input** **type**="text" class="form-control" **asp-for**="Subject" placeholder="Konu">

</div>

<div class="form-group mb-3">

<label>Mesajınız</label>

@\* <input type="text" class="form-control" asp-for="Body" placeholder="Mesajınız"> \*@

<**textarea** **asp-for**="Body" class="form-control" rows="10" cols="3"></**textarea**>

</div>

<button type="submit" class="btn btn-outline-primary btn-lg btn-rounded btn-fw me-2">Mail Gönder</button>

</**form**>