Промышленное программирование

Лекция 8

❶ OpenAPI и библиотека Swashbuckle
❷ DTO и библиотека AutoMapper
❸ Хранение собственных настроек

OpenAPI

Web API Server

1. Как клиенту узнать, какие HTTP-методы по каким адресам доступны?
   → Нужна документация

2. Можно ли автоматически сгенерировать клиентские классы?
   → Нужен строгий формат описания API

OpenAPI

```
{
  "openapi": "3.0.1",
  "info": {
    …
  },
  "paths": {
    "/api/user": {
      "get": {
        "summary": "Returns list of all users.",
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "text/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": …
                  }
    …
  },
  "components": …
}
```

Swashbuckle

***.csproj** Устанавливаем nuget-пакет

```
<Project Sdk="Microsoft.NET.Sdk.Web">
    …
    <ItemGroup>
        <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
    </ItemGroup>
</Project>
```

Program.cs “Публикуем” swagger.json и UI

```
…
builder.Services.AddRouting(options => options.LowercaseUrls = true);
builder.Services.AddSwaggerGen();
…
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();    // http://localhost/swagger/v1/swagger.json
    app.UseSwaggerUI(); // http://localhost/swagger
}
…
```

Swashbuckle

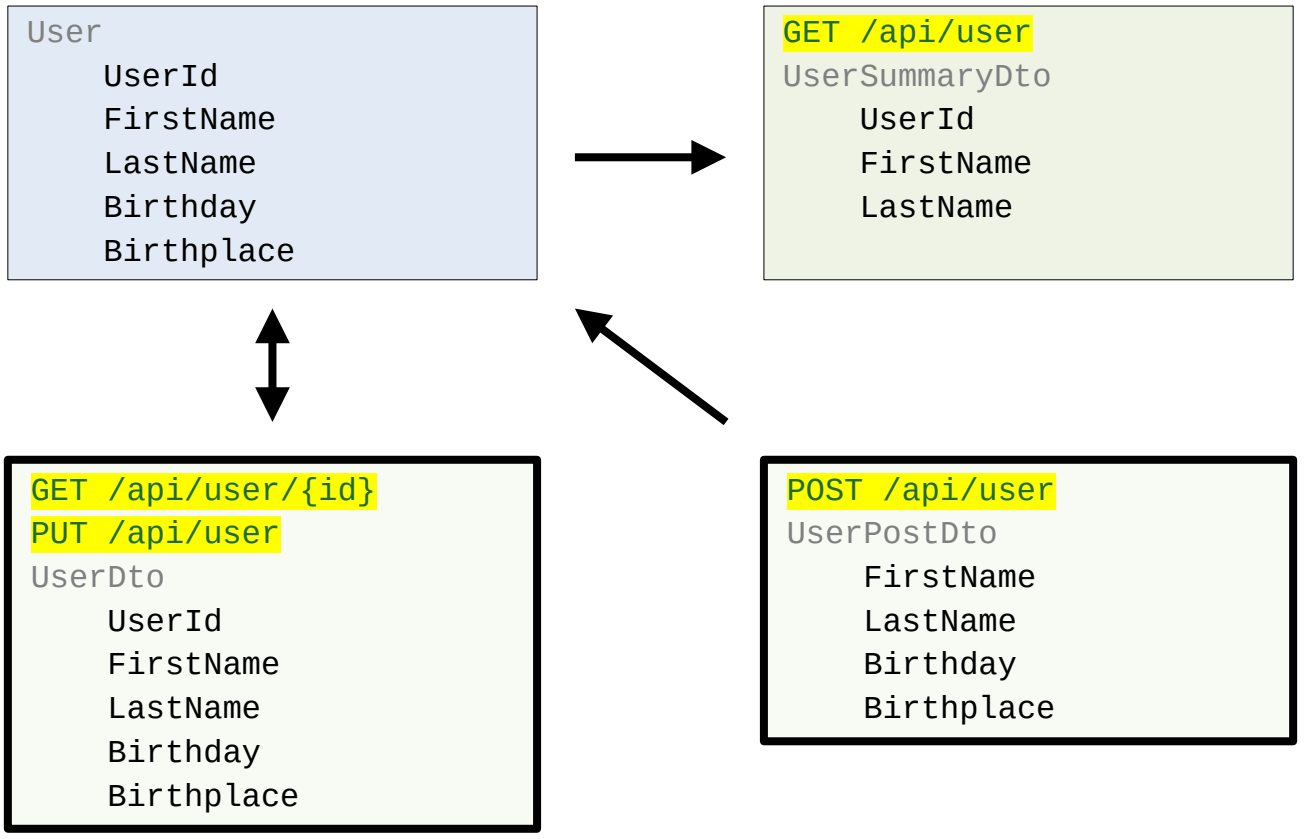
***.csproj** Включаем в проекте генерацию документационного файла

```
<Project Sdk="Microsoft.NET.Sdk.Web">
    <PropertyGroup>
        <GenerateDocumentationFile>True</GenerateDocumentationFile>
        <NoWarn>$(NoWarn);1591</NoWarn>
        <Nullable>enable</Nullable>
        <TargetFramework>net6.0</TargetFramework>
    </PropertyGroup>
…
</Project>
```

Program.cs Указываем Swagger где искать документационный файл

```
builder.Services.AddSwaggerGen(options =>
{
    var docName = $"{Assembly.GetExecutingAssembly().GetName().Name}.xml";
    var docPath = Path.Combine(AppContext.BaseDirectory, docName);
    options.IncludeXmlComments(docPath);
});
```

DTO


User
UserId
FirstName
LastName
Birthday
Birthplace
GET /api/user
UserSummaryDto
UserId
FirstName
LastName
GET /api/user/{id}
PUT /api/user
UserDto
UserId
FirstName
LastName
Birthday
Birthplace
POST /api/user
UserPostDto
FirstName
LastName
Birthday
Birthplace

AutoMapper

***.csproj** Устанавливаем nuget-пакет

```
<Project Sdk="Microsoft.NET.Sdk.Web">
    …
    <ItemGroup>
        <PackageReference Include="AutoMapper.Extensions.Microsoft.DependencyInjection" Version="12.0.0" />
    </ItemGroup>
</Project>
```

AutoMapper

MappingProfile.cs Объявляем профиль

```
using AutoMapper;

public class MappingProfile : Profile
{
    public MappingProfile()
    {
        CreateMap<User, UserDto>().ReverseMap();
        CreateMap<UserPostDto, User>();
    }
}
```

Program.cs Регистрируем профиль

```
…
builder.Services.AddAutoMapper(Assembly.GetExecutingAssembly());
…
```

AutoMapper

```
[ApiController]
[Route("api/[controller]")]
public class UserController : ControllerBase
{
    private readonly Context _context;
    private readonly IMapper _mapper;

    public UserController(Context context, IMapper mapper)
    {
        _context = context;
        _mapper = mapper;
    }

    [HttpGet]
    public IEnumerable<UserDto> Get()
    {
        return _mapper.Map<IEnumerable<UserDto>>(_context.Users);
    }

    [HttpPost]
    public void Post(UserPostDto userDto)
    {
        var user = _mapper.Map<User>(userDto);
        user.UserId = …;
        _context.Users.Add(user);
    }
}
```

Файл appsettings.json

```
{
  "AllowedHosts": "*",
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "UserWall.Server": {
    "PageSizeLimit": 10
  }
}
```

Как получить доступ к пользовательским данным в файле конфигурации?
Configuration in ASP.NET Core

“Топорный” вариант

```
public class UserController : ControllerBase
{
    private readonly IConfiguration _config;

    public UserController(IConfiguration config)
    {
        _config = config;
    }

    [HttpGet]
    public IEnumerable<UserDto> Get()
    {
        var pageSizeLimit = config.GetSection("UserWall.Server")["PageSizeLimit"];
        …
    }
}
```

Более предпочтительный вариант

Options.cs

```
public sealed class Options
{
    public int PageSizeLimit { get; set; }
}
```

Program.cs

```
var assemblyName = Assembly.GetExecutingAssembly().GetName().Name;
builder.Services.Configure<Options>(builder.Configuration.GetSection(assemblyName));
```

UserController.cs

```
public class UserController : ControllerBase
{
    private readonly Options _options;

    public UserController(IOptions<Options> options)
    {
        _options = options.Value;
    }
}
```