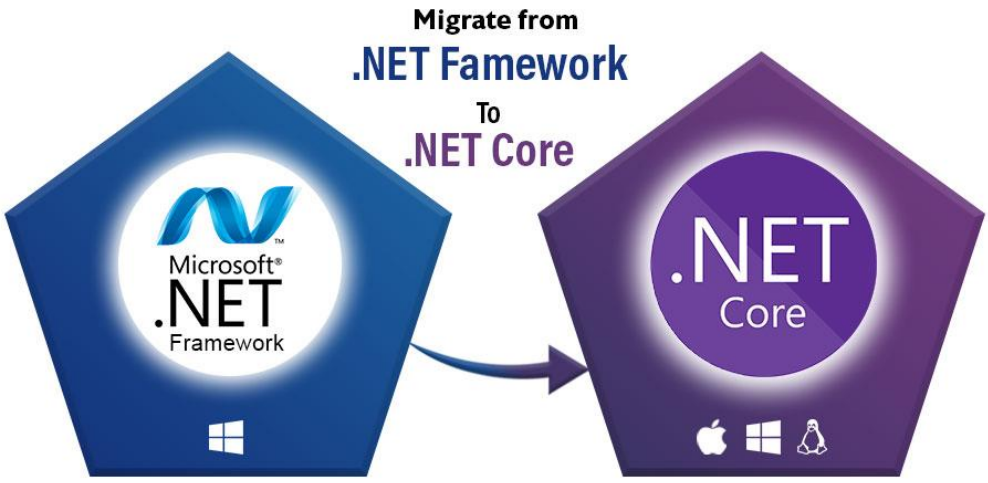

MIGRATION

Documentation

Abstract

In this module I learn how to migrate any .NET framework application in .NET Core

Dhruvil Dobariya
dhruvildobariya21@gmail.com

INDEX

ASP.NET Web API 2 To ASP.NET Core Web API

1.1 Introduction

- We have different steps to migrate ASP.NET Web API 3 to ASP.NET Core Web API project.
  - Create new project
  - Remove unnecessary files
  - Configure ASP.NET Core application based on ASP.NET Web API 2 application
  - Migrate Models
  - Migrate Controllers

1.2 Create new project

- We have different steps to create new project of ASP.NET Core Web API.
- From the File menu, select **New > Project**
- Enter **Web API** in the search box.
- Select the **ASP.NET Core Web API** template and click on Next.
- In the Configure your new project dialog, give name the project like "ProductAPI" and click on Next.
- In the Additional information dialog,
  - Choose the version of .NET (.NET 6 or above)
  - Confirm the checkbox for Use controllers(uncheck to use minimal APIs) is checked
  - Check Enable OpenAPI support (if we want to use of swagger)
  - Click on Create
- Now we have ASP.NET Core Web API project.

1.3 Remove unnecessary files

- Microsoft give build in created file in project to run as example.
- So, first we must delete those file.
- For this, we have different steps,
- Remove the **WeatherForecast.cs** and **Controllers/WeatherForecastController.cs** example files from the new ProductsCore project.
- Change **launchUrl** properties from "weatherforcast" to "product"(if we don't enable OpenAPI).

1.4 Configure ASP.NET Core application based on ASP.NET Web API 2 application

- Now ASP.NET Core Web API project doesn't contains **Global.asax**, **WebConfig.cs** and **Web.config**.
- First we want to migrate all custom configuration from **Web.config** to **application.json** based on our requirement.
- Now, in ASP.NET Core Web API (.NET 6 or above) project all global level event and methods and dependencies are managed and register in **"Program.cs"**.
- So we must configure the **Program.cs** based on **Global.asax** and **WebConfig.cs** and other child config file of **WebConfig.cs**.

1.5 Migrate Models

- If we use any architecture, like 3 tier architecture, N tier architecture, Onion architecture, Clean architecture and other which separate Models and place in other layer then we don't want to migrate models, because we have separate class library for Models.
- So just we detach form framework application and attach it in core application.
- But If our .NET framework project contains folder of Models the we have different steps to migrate it.
- In Solution Explorer, right-click the project. Select **Add > New Folder**. Name the folder Models.
- Now migrate all the models form ASP.NET Web API 2 application to ASP.NET Core Web API application.
- Now change namespace of all the models based on .NET Core application and our requirement.

1.6 Migrate Controller

- First we should move all the controller file from ASP.NET Web API 2 application to ASP.NET Core Web API application.
- Now we should change all the namespace based on our application.
- In ASP.NET Core Web API 2 all the controller inherit by **ApiController** class when in ASP.NET Core Web API all the controller inherit by **ControllerBase** class.
- Second thing if we want to use controller as API controller in ASP.NET Core Web API then we must give attribute **[ApiController]** on controller.
- Third thing is in ASP.NET Web API 2 we configure route using conventional based routing and attribute based routing where in ASP.NET Core Web API we must use attribute based routing for API controllers.

Migration

- Fourth ting is in ASP.NET Web API 2 we use return type IHttpActionResult and HttpResponceMessage where in ASP.NET Core Web API we use IActionResult or ActionResult.

Example:

```
using Microsoft.AspNetCore.Mvc;
using ProductsAPI.Models;

namespace ProductsAPI.Controllers;

[Route("api/[controller]")]
[ApiController]
public class ProductsController : ControllerBase
{
    Product[] products = new Product[]
    {
            new Product
            {
                Id = 1, Name = "Tomato Soup", Category = "Groceries", Price = 1
            },
            new Product
            {
                Id = 2, Name = "Yo-yo", Category = "Toys", Price = 3.75M
            },
            new Product
            {
                Id = 3, Name = "Hammer", Category = "Hardware", Price = 16.99M
            }
    };

    [HttpGet]
    public ActionResult<IEnumerable<Product>> GetAllProducts()
    {
        return Ok(products);
    }

    [HttpGet("{id}")]
    public ActionResult<Product> GetProduct(int id)
    {
        var product = products.FirstOrDefault((p) => p.Id == id);
        if (product == null)
        {
            return NotFound();
        }
        return Ok(product);
    }
}
```

Migration

- Now we have new .NET Core application which we are successfully done migration form ASP.NET Web API 2 to ASP.NET Core Web API.