File-based apps

Бердников Вадим

Однофайловые приложения

```
$ echo 'Console.WriteLine("Hello, World!")' > example.cs
$ dotnet example.cs
Hello, World!
```

Однофайловые приложения

```
1 #!/usr/bin/env dotnet
2 Console.WriteLine("Hello, World!");
```

```
$ chmod +x example.cs
$ ./example.cs
Hello, World!
```

Однофайловые приложения (без файлов)

```
$ echo 'Console.WriteLine("Hello, World!");' | dotnet run -
Hello, World!
```

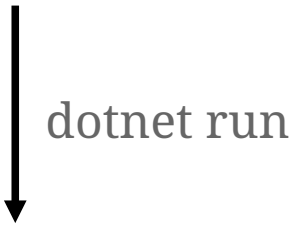
Однофайловые приложения (без файлов)

```
$ echo 'Console.WriteLine("Hello, World!");' | dotnet run -
Hello, World!
```

```
$ curl -s https://csharp-scripts.ru/some-script.cs | dotnet run -
```

Как это работает?

Файл с кодом



Генерация файла проекта

Сборка и запуск приложения так же, как и обычно

Как это работает?

Кэш в Linux:

```
/home/%user%/.local/share/dotnet/runfile/...
```

```
$ dotnet build example.cs
```

Кэш в Windows:

```
C:\Users\%user%\AppData\Local\Temp\dotnet\runfile\...
```

```
$ dotnet clean file-based-apps
```

Что работает иначе?

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net10.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <PublishAot>true</PublishAot>
    <PackAsTool>true</PackAsTool>
    <UserSecretsId>example-
808f5b947101fb86d3db40be6b476817439845b870fa9743f9fa405c5ce91132</UserSecretsId>
  </PropertyGroup>

</Project>
```

Конвертация в обычный проект

```
~/example/..
  Directory.Build.props
  example.cs
  otherExample.cs
  otherExample.run.json
  readme.md
  some-binary
```

```
$ dotnet project convert example.cs
Укажите выходной каталог (example): convertedExample
```

```
~/example/..
  convertedExample
    Directory.Build.props
    example.cs
    example.csproj
    otherExample.run.json
    readme.md
    some-binary
  Directory.Build.props
  example.cs
  otherExample.cs
  otherExample.run.json
  readme.md
  some-binary
```

Добавление пакетов Nuget

```
#:package APIVerve.API.RandomJoke@1.*
#:package Uwuify.Humanizer@*
```

```
1  //example.cs
2  #:package APIVerve.API.RandomJoke@1.*
3  #:package Uwuify.Humanizer@*
4
5  using Uwuify.Humanizer;
6  using APIVerve.API.RandomJoke;
7
8  if (args.Length < 1)
9  {
10     Console.WriteLine("Insert your api-key as an argument");
11     return;
12 }
13
14 var apiClient = new RandomJokeAPIClient(args[0]);
15 while (true)
16 {
17     try
18     {
19         var response = await apiClient.ExecuteAsync();
20         if (response.Error is not null)
21         {
22             Console.WriteLine("Error occured.".Uwuify());
23         }
24         else
25         {
26             Console.WriteLine(response.Data.Setup.Uwuify());
27             Console.WriteLine(response.Data.Punchline.Uwuify());
28             Console.WriteLine();
29         }
30     }
31     catch (Exception e)
32     {
33         Console.WriteLine($"Error: {e.Message.Uwuify()}");
34     }
35     await Task.Delay(5000);
36 }
```

Другие директивы

```
1 //directives.cs
2 #:sdk Microsoft.NET.Sdk.Web
3 #:project someProject/someProject.csproj
4 #:property PublishAot=false
5
6 using someProject;
7
8 var builder = WebApplication.CreateBuilder();
9 builder.Services.AddSingleton<DateTimeProvider>();
10
11 var app = builder.Build();
12 app.MapGet("time", (DateTimeProvider provider) => provider.Now);
13 app.Run();
```

```
1 //someProject/DateTimeProvider.cs
2 namespace someProject;
3
4 public class DateTimeProvider
5 {
6     public DateTime Now => DateTime.Now;
7 }
```

Дополнительная конфигурация

dotnet cli будет искать в папке с приложением:

- [ApplicationName].run.json
- Directory.Build.props
- Directory.Packages.props
- nuget.config
- global.json
- appsettings.json (в случае с веб-приложениями)

Поддержка IDE

Пока не работает в:

- Rider
- Visual Studio

Неплохо работает в:

- VS Code
- Любой редактор, поддерживающий LSP + Roslyn language server

Сценарии использования

```
#:sdk Microsoft.NET.Sdk.Web

var builder = WebApplication.CreateBuilder();
var app = builder.Build();
app.MapGet("/", () => EndpointHandler(""));
app.MapGet("/{**path}", EndpointHandler);

Console.WriteLine($"Working in {AppContext.BaseDirectory}");
app.Run();

IResult EndpointHandler(string path)
{
    var fullPath = Path.Combine(AppContext.BaseDirectory, path);
    if (fullPath.Split(['\\', '/']).Any(x => x == "..")) return Results.NotFound();
    if (Directory.Exists(fullPath))
    {
        var dirs = Directory.EnumerateDirectories(fullPath)
            .Select(x => EntryToRef(path, x, "orange"));
        var files = Directory.EnumerateFiles(fullPath)
            .Select(x => EntryToRef(path, x, "green"));

        Console.WriteLine($"Directory '{fullPath}' contents has been sent.");
        return Results.Text(string.Join("<br>", dirs.Concat(files)), contentType: "text/html");
    }
    if (!File.Exists(fullPath)) return Results.NotFound();

    Console.WriteLine($"File '{fullPath}' has been sent.");
    return Results.File(fullPath, contentType: "text/plain");
}

static string EntryToRef(string parentDir, string entry, string entryColor)
    => $"<a href ='{Path.Combine(parentDir, entry.Split(['\\', '/']).Last())}' style='color:
{entryColor}'>{entry}</a>";
```

Сценарии использования

```
#:sdk Aspire.AppHost.Sdk@13.0.0
#:package Aspire.Hosting.AppHost@13.0.0

var builder = DistributedApplication.CreateBuilder(args);

var cache = builder.AddRedis("cache")
    .WithDataVolume();

var postgres = builder.AddPostgres("postgres")
    .WithDataVolume()
    .AddDatabase("tododb");

var todoApi = builder.AddProject<Projects.TodoApi>("api")
    .WithReference(cache)
    .WithReference(postgres);

builder.AddNpmApp("frontend", "../TodoApp")
    .WithReference(todoApi)
    .WithReference("api")
    .WithHttpEndpoint(env: "PORT")
    .WithExternalHttpEndpoints();

builder.Build().Run();
```

Полезное

- https://github.com/prtsie/file-based-apps — код из презентации
- https://learn.microsoft.com/en-us/dotnet/core/sdk/file-based-apps — документация Microsoft
- https://github.com/dotnet-script/dotnet-script — неофициальный проект с похожим функционалом