

Development Environment

Clone the "Shaper" main library from GitHub <https://github.com/brayns-it/shaper>

Create a new empty ASP.NET Core Project and simply call "InitializeShaper" and "MapShaperApi".

```
using Brayns.Shaper;

var builder = WebApplication.CreateBuilder(args);
builder.InitializeShaper();

var app = builder.Build();
app.MapShaperApi();

app.Run();
```

"MapShaperApi" will map two path in your web application:

- /api to serve REST request with GET, POST, PUT or DELETE method
- /rpc to serve special JSON request for the client

If you want to use also the web client (not only API) you have to clone "Shaper Web" library from GitHub <https://github.com/brayns-it/shaper-web>

Simply declare the web client and Web Sockets support:

```
app.MapShaperClient();
app.UseWebSockets();
```

"MapShaperClient" will catch all requests from "/client" base URI and serve the default index.html client page.

If you want to redirect also "/" path to default client page, add the following code:

```
app.MapShaperDefault();
```

Enable serving of static files and allows unknown MIME types (for example to enable Let's Encrypt HTTP validation):

```
app.UseStaticFiles(new StaticFileOptions
{
    ServeUnknownFileTypes = true,
    DefaultContentType = "application/other"
});
```

To enable scheduled task:

```
app.UseShaperMonitor();
```

Mark the ASP.NET assembly as Shaper App container within the "AssemblyInfo" file (create it if doesn't exists):

```
[assembly: Brayns.Shaper.Classes.AppCollection]
```

Create the following directory structure:

- **code** (that contains specific project code)
- **var** (that contains configuration and logs)
- **var\resources** (that contains embedded resources)
- **wwwroot** (that contains HTTP served resources)

Complete "program.cs" example

```
using Brayns.Shaper;

var builder = WebApplication.CreateBuilder(args);
builder.InitializeShaper();

var app = builder.Build();
app.MapShaperApi();
app.MapShaperClient();
app.MapShaperDefault();
app.UseWebSockets();
app.UseStaticFiles(new StaticFileOptions
{
    ServeUnknownFileTypes = true,
    DefaultContentType = "application/other"
});
```

```
app.UseShaperMonitor();
```

```
app.Run();
```

Project Configuration

Project Configuration (csproj) must be adapted to:

- Embed PO files with translations
- Deploy **var\resources** folder

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

  ...

  ...

  <ItemGroup>
    <None Remove="**/*.po" />
  </ItemGroup>

  <ItemGroup>
    <EmbeddedResource Include="**/*.po" />
  </ItemGroup>

  <ItemGroup>
    <None Update="var\resources\**">
      <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
    </None>
  </ItemGroup>

  ...

  ...

</Project>
```

Publish Profile

Add the following lines to Publish Profile (pubxml) to preserve "var" directory:

```
<Project>
```

```
...
```

```
...
```

```
<ItemGroup>
```

```
  <Content Update="var\resources" CopyToPublishDirectory="PreserveNewest" />
```

```
  <Content Update="var\*" CopyToPublishDirectory="Never" />
```

```
</ItemGroup>
```

```
..
```

```
..
```

```
</Project>
```

Revision #15

Created 1 December 2023 10:30:42 by Simone Giordano

Updated 20 August 2024 15:50:26 by Simone Giordano