

- [Host and Deploy Narforum Software on Windows Server IIS](#)
- [Requirements](#)
- [Windows Server Requirements](#)
- [Configuring CloudFlare and Domain Name \(Optional\)](#)
- [Configuring Web Api](#)
- [Configuring Admin Blazor WASM \(NarForumAdmin\)](#)
- [Configuring User Blazor WASM Hosted \(NarForumAdmin\)](#)

Host and Deploy Narforum Software on Windows Server IIS

Note: This documentation for windows server. If you know already what you do, you dont have to apply these steps for the installation. If you don't want to use Windows Server, You can try Docker option which doesn't exist the current version of the project. (So you have to make it yourself for now. I'm going to add this option for the next versions.)

Requirements

- Windows Server 2022
- Domain name
- Cloudflare account (Optional)

Note: Cloudflare is optional. I used it because of free SSL and caching features. It has more features that paid or free can be useful for your project. That's the easy part for our usage. It's optional, you don't have to use it for this part.

Windows Server Requirements

Narforum consists of three separate projects. These projects:

- ASP.NET Core Web Api (v8)
- Blazor Webassembly Standalone Application (NarForumAdmin) (.NET 8)
- Blazor Webassembly Interactive Global Application (NarForumUser) (.NET 8)

According to these projects we need to setup Windows Server:

1. Open **Server Manager**.
2. Click **Add Roles and Features** option at the Manage Menu
3. Click Next, and choose Web Server(IIS) role.
4. Click Next, and choose necessary components(The installed ones on NarForum Server) After these installations probably your server will be restarted:
 - Web Server (IIS)
 - Common HTTP Features
 - Default Document
 - Directory Browsing
 - HTTP Errors
 - Static Content
 - Health and Diagnostics
 - HTTP Logging
 - Performance
 - Static Content Compression
 - Security
 - Request Filtering
 - Application Development
 - WebSocket
 - File and Storage Services
 - Storage Services
 - Management Tools
 - IIS Management Console
5. Install **.NET 8 SDK ,Hosting Bundle**. <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>
6. Also install **URL Rewrite** <https://www.iis.net/downloads/microsoft/url-rewrite> to make sure IIS Module Blazor works with Blazor WASM files correctly.
7. Download **PostgreSQL**. NarForum software uses PostgreSQL as relational database management system. It's free and open-source system software.
8. Install PostgreSQL on the server and create server, user and password. We will use them in the **connection string**.
9. Download **Garnet**. (The installed version is win-x64-based-readytorun.zip file for NarForum software) Garnet is a remote cache-store also it's an Redis alternative. There is no installation, we are going to execute the executable file. But it's must always work on the server, if not the api won't work without it.
10. (**Optinal**) Also you can install **Node.js** and **NPM** to install **Redis-Cli** package. This tool provides control on the command line. Why Redis-Cli, because Garnet is also compatible with redis-cli as Redis alternative.

Configuring CloudFlare and Domain Name (Optional)

1. Create an account for Cloudflare if it does not exist.
2. Click **Add a domain** button.
3. Enter an existing domain.
4. Open **DNS Records** and add records like table below:

Type	Name	Content (IPv4)
A	admin	XXX.XXX.XXX.XXX
A	api	XXX.XXX.XXX.XXX
A	@	XXX.XXX.XXX.XXX

5. Cloudflare will give nameservers like below:

Type	Value
NS	xxx.ns.cloudflare.com
NS	xxx.ns.cloudflare.com

6. Since the NarForum domain name is purchased from a different provider, I need to register the nameservers through that provider. I changed default the nameservers value with the nameservers are given by CloudFlare.
7. Open SSL/TLS, and configure SSL/TLS which option is fit for you.
8. Open Caching, and click the Purge Everything in Configuration module.

Configuring Web Api

1. Open NarForum Web Api project from Visual Studio, or any IDE that you used for it.
2. In Program.cs file, add your address as allowed origin for CORS and WebSockets:

```

builder.Services.AddWebSockets(o => {
    ...
    o.AllowedOrigins.Add("https://domain-name.com");
    o.AllowedOrigins.Add("https://admin.domain-name.com");
});

builder.Services.AddCors(options =>
{
    options.AddPolicy("all", builder => builder
        .WithOrigins("...", "...", "https://domain-name.com",
"https://admin.domain-name.com")
        .AllowAnyHeader()
        .AllowAnyMethod()
        .AllowCredentials()
        .SetIsOriginAllowed((host) => true)
    );
    ...
}

```

3. Right click and click Publish option.

4. Publish settings option like below:

Label	Value
Target Location	bin\Release\net8.0\publish\
Delete existing files	false
Configuration	Release
Target Framework	net8.0
Target Runtime	Portable

5. Click the **Publish** button. When publish succeeded, click navigate copy all files from this folder for your server.

6. Create a folder named Publish in your Windows Server Desktop, and create new folder named api.domain-name.com folder inside Publish folder. Paste all files that copied from your local machine.

7. Open appsettings.json file from api.domain-name.com file, and change the two connection string according to PostgreSQL Server that installed Windows Server:

```

{
    ...
    "ConnectionStrings": {
        "ForumDatabaseConnectionString":

```

```
"Host=localhost;Database=db_forum;Username=postgres;Password=pwd;Port=5432;",
    "ForumIdentityDatabaseConnectionString":
    "Host=localhost;Database=db_forum_identity;Username=postgres;
    Password=pwd;Port=5432;"
    },
    ...
```

8. Open IIS Manager and right-click Sites and click Add Website button:

- Enter site name, example: api.domain-name.com
- Choose content directory which is your api files from Publish directory. Choose the api.domain-name.com folder.
- Binding options:
 - Type: http
 - IP Address(IPv4): XXX.XXX.XXX.XXX (your server IP Address for external that also used in Cloudflare)
 - Port: 80
 - Host name: api.domain-name.com
- Uncheck **Start Website** immediately for now.
- Click OK.
- Open Home panel in IIS Manager and click **Server Certificates**
- Click **Create Self-Signed Certificate** from the right panel.
- Give a name to this certificate, for example: CloudFlare-SelfSigned and choose Personal store.
- Click OK.
- Right-Click api.domain-name.com from Sites tree menu.
- Click **Edit Bindings..** button.
- Click binding to add https type of api.domain-name.com:
- Binding options:
 - Type: https
 - IP Address(IPv4): XXX.XXX.XXX.XXX (your server IP Address for external that also used in Cloudflare)
 - Port: 443
 - Host name: api.domain-name.com
 - SSL certificate: Cloudflare-SelfSigned
- Before start this api, we need give permission to IIS Manager from content directory. Go Publish folder in Windows Server Desktop.
- Right-Click Publish folder, click Properties, open Security tab.
- Edit Group or user names, add IIS_IUSRS user.
- Give **Read & Execute** , **List folder contents** and **Read** permissions.

- Go back to IIS Manager, open **Application Pools** and Right-Click api.domain-name.com and click Edit Application Pool.
- Select No Managed Code as .NET CLR version.
- Click OK
- Open IIS Manager Home panel, click **Authentication**
- Enable **Anonymous Authentication**
- Make sure GarnetServer is running on the server. If it does not then click GarnetServer .exe file to run.
- Open IIS Manager, select api.domain-name.com from Sites and click Start button.
- Let's check out <https://api.domain-name.com/swagger>. If you see swagger ui with no error everything is great. Also check if databases are created in PostgreSQL.

Configuring Admin Blazor WASM (NarForumAdmin)

1. Open NarForumAdmin Blazor WASM Standalone project from Visual Studio, or any IDE that you used for it.
2. Right click and click Publish option.
3. Publish settings option like below:

Label	Value
Target Location	bin\Release\net8.0\blazor-wasm\publish\
Delete existing files	false
Configuration	Release
Target Framework	net8.0
Target Runtime	browser-wasm

4. Click the **Publish** button. When publish succeeded, click navigate copy all files from this folder for your server.
5. Create a folder named Publish in your Windows Server Desktop, and create new folder named admin.domain-name.com folder inside Publish folder. Paste all files

that copied from your local machine.

6. Open appsettings.json file from admin.domain-name.com folder, and add your api urls:

```
{
  "ApiBaseUrl": "https://api.domain-name.com/api",
  "HubBaseUrl": "https://api.domain-name.com"
}
```

7. Open IIS Manager and right-click Sites and click Add Website button:

- Enter site name, example: admin.domain-name.com
- Choose content directory which is your admin files from Publish directory. Choose the **admin.domain-name.com** folder.
- Binding options:
 - Type: http
 - IP Address(IPv4): XXX.XXX.XXX.XXX (your server IP Address for external that also used in Cloudflare)
 - Port: 80
 - Host name: admin.domain-name.com
- Uncheck **Start Website** immediately for now.
- Click OK.
- Right-Click admin.domain-name.com from Sites tree menu.
- Click **Edit Bindings..** button.
- Click binding to add https type of admin.domain-name.com:
- Binding options:
 - Type: https
 - IP Address(IPv4): XXX.XXX.XXX.XXX (your server IP Address for external that also used in Cloudflare)
 - Port: 443
 - Host name: admin.domain-name.com
 - SSL certificate: Cloudflare-SelfSigned

- Open **Application Pools** and Right-Click admin.domain-name.com and click Edit Application Pool.
- Select No Managed Code as .NET CLR version.
- Click OK
- Open IIS Manager, select admin.domain-name.com from Sites and click Start button.
- Let's check out <https://admin.domain-name.com>

Configuring User Blazor WASM Hosted (NarForumAdmin)

1. Open NarForumUser project from Visual Studio, or any IDE that you used for it.
2. Right click and click Publish option.
3. Publish settings option like below:

Label	Value
Target Location	bin\Release\net8.0\publish\
Delete existing files	true
Configuration	Release
Target Framework	net8.0
Target Runtime	win-x64
Deployment Mode	Self-contained (????)

4. Click the **Publish** button. When publish succeeded, click navigate copy all files from this folder for your server.
5. Create a folder named Publish in your Windows Server Desktop, and create new folder named **domain-name.com** folder inside Publish folder. Paste all files that copied from your local machine.
6. Open **appsettings.json** file from **domain-name.com** folder, and add your api urls:


```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ApiBaseUrl": "https://api.narforum.com/api",
  "HubBaseUrl": "https://api.narforum.com",
  "BaseUrl": "https://domain-name.com"
}
```

7. Open wwwroot in **domain-name.com** folder and edit appsettings.json file too:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "ApiBaseUrl": "https://api.domain-name.com/api",
  "HubBaseUrl": "https://api.domain-name.com",
  "BaseUrl": "https://domain-name.com",
  "SiteName": "Domain-Name",
  "Locale": "en_EN",
  "HrefLang": "en",
  "ContentLanguage": "en",
  "Description": "Your forum website description"
}
```

8. Open IIS Manager and right-click Sites and click Add Website button:

- Enter site name, example: **domain-name.com**
- Choose content directory which is your admin files from Publish directory. Choose the **domain-name.com** folder.
- Binding options:
 - Type: http
 - IP Address(IPv4): XXX.XXX.XXX.XXX (your server IP Address for external that also used in Cloudflare)
 - Port: 80
 - Host name: domain-name.com

- Uncheck **Start Website** immediately for now.
- Click OK.
- Right-Click admin.domain-name.com from Sites tree menu.
- Click **Edit Bindings..** button.
- Click binding to add https type of domain-name.com:
- Binding options:
 - Type: https
 - IP Address(IPv4): XXX.XXX.XXX.XXX (your server IP Address for external that also used in Cloudflare)
 - Port: 443
 - Host name: domain-name.com
 - SSL certificate: Cloudflare-SelfSigned
- Open **Application Pools** and Right-Click admin.domain-name.com and click Edit Application Pool.
- Select No Managed Code as .NET CLR version.
- Click OK
- Open IIS Manager, select domain-name.com from Sites and click Start button.
- Let's check out https://domain-name.com