

Host SvelteKit project in Microsoft IIS

Asked 1 year, 1 month ago Modified 6 months ago Viewed 2k times



I've been trying to deploy a svelte application into an IIS server my company has provided. I've tried using [this](#) adapter, but i couldn't manage to make it work.

3

Should I keep on trying on maybe suggest some other solution, like express?



Thanks for the help.



javascript

iis

svelte

vite

Share Edit Follow

asked Jul 18, 2023 at 8:45



mvioole

142 2 11

Does the application have backend logic? If not, you don't need any of that. – [brunnerh](#) Jul 18, 2023 at 8:54



It does, it needs to handle a couple of API endpoints. – [mvioole](#) Jul 18, 2023 at 8:55

You should be more specific in how it does not work then. Also, if using IIS is not a strict requirement, I would not bother with it. – [brunnerh](#) Jul 18, 2023 at 8:56

I'll try and push on changing the hosting solution then. Thanks for your help – [mvioole](#) Jul 18, 2023 at 9:02

This link may help you: <https://dev.to/wearahelmet/deploying-svelte-app-to-iis-1kia>. – [samwu](#) Jul 18, 2023 at 9:59

2 Answers

Sorted by:

Highest score (default)



Não encontrou uma resposta? [Pergunte em Stack Overflow em Português.](#)



Deploying Sveltekit App IIS

5

Building App



Sveltekit apps being deployed on IIS need to be built using adapter node



- `npm i -D @sveltejs/adapter-node`



After install open svelte.config and change top line to use adapter-node: `import adapter from '@sveltejs/adapter-node';`

Inside your svelte.config make sure to use this node

```
import adapter from '@sveltejs/adapter-node';
import { vitePreprocess } from '@sveltejs/kit/vite';

/** @type {import('@sveltejs/kit').Config} */
const config = {
  kit: {
    adapter: adapter(),
  },
  preprocess: vitePreprocess()
};

export default config;
```

Build app: `npm run build`

Deploying on server

Node and IISNode must be installed on server

Set up site url

1. Create physical path for built in `C:\inetpub\projectname\wwwroot`
2. Add index.html inside of root with boilerplate

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello World</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

3. Add site:

- Open IIS Manager
- Click on Server Name -> Sites
- Right click on sites -> Add Website
- Give it a name and host name (web address sitedomain)

- Physical path inetpub -> wwwroot

4. Add write permissions for site

- Right click on project folder(inetpub->projectfolder) -> Properties -> security -> edit -> add
- Click on locations
- Select server at top of location window -> Click OK
- In the object names box add the site name (you can see it in the IIS Manager if you forget)
- Click Check Names
- The site name should be added and underlined if entered correctly
- Click OK
- In Permissions box click on site name -> Allow write permissions -> OK

Test URL

On local pc go to C:\Windows\System32\drivers\etc -> host add ip and webaddress

- add domain to host file
 - 10.39.2.50 sitedomain

Navigate to http://sitedomain Should see boilerplate html

Add build to server

Add build folder inside project wwwroot folder

Inside your build folder you need to add two files.

server.cjs file:

```
process.env.SOCKET_PATH = process.env.PORT;  
delete process.env.PORT  
  
import('./index.js')
```

web.config file:

```
<?xml version="1.0" encoding="UTF-8"?>  
<configuration>  
  <system.webServer>  
    <handlers>  
      <add name="iisnode" path="server.cjs" verb="*" modules="iisnode" />  
    </handlers>  
  </system.webServer>  
</configuration>
```

```

</handlers>
<iisnode
  nodeProcessCommandLine="&quot;C:\Program Files\nodejs\node.exe&quot;"
  watchedFiles="web.config;*.js;*.cjs"
/>
<rewrite>
  <rules>
    <!-- All other URLs are mapped to the node.js site entry point -->
    <rule name="node">
      <match url=".*" />
      <action type="Rewrite" url="server.cjs" />
    </rule>
  </rules>
  <!-- Change it back if Location Header got rewritten-->
  <outboundRules>
    <rule name="back">
      <match serverVariable="RESPONSE_Location" pattern="
(.*?)server.cjs" />
      <action type="Rewrite" value="{R:1}" />
    </rule>
  </outboundRules>
</rewrite>
<defaultDocument>
  <files>
    <add value="server.cjs" />
  </files>
</defaultDocument>
</system.webServer>
</configuration>

```

- Copy build files to server build folder
- Copy package.json, package-lock.json, and .env(if there is one) into build folder

Install packages

1. Open cmd as administrator
2. Cd into build folder
3. Run `npm ci --omit dev`

Prisma apps

If you are using prisma you will need to copy the .prisma folder from local pc node-modules into server node-modules. Also if you have more than one database connection you will need to copy each generated prisma node `@generated-db`

Change Site location

After all of these steps are completed go back into IIS Manager and click on the new site -> basic settings and change the physical path from wwwroot to build folder.

Share Edit Follow

answered Aug 11, 2023 at 13:11



Avi4nFLu

328 2 15

Our site is running on IIS test.securesship.ca:8081 But for some reason the `event.getClientAddress()` method returns undefined. Any thoughts? You can see the output that I returned on the console.

– Shamim Hossain Sep 29, 2023 at 17:25

Thanks for the above info. It took me some time to set up. Everything is working as expected. However, it seems like the `.env` variables are not loading automatically. I have installed `dotenv` on the project but don't know how to run this command `node -r dotenv/config build` to load them automatically. Or just copying the `.env` file would be enough here? – Shamim Hossain Sep 29, 2023 at 21:02

I copy the `.env` file into the build directory and it works for me as long as I install `dotenv`. As far as the `event.getClieNdAddress()` I don't have any suggestions sorry. – Avi4nFLu Oct 2, 2023 at 15:35



0



I had a lot of trouble finding this when I was looking so I thought I'd put it here for anyone else looking.

The easiest way now is to use something like <https://github.com/abaga129/sveltekit-adapter-iis> to automatically create the node server and web config for you.



Share Edit Follow



answered Feb 25 at 17:07



iedan

1

- 1 Your answer could be improved with additional supporting information. Please [edit](#) to add further details, such as citations or documentation, so that others can confirm that your answer is correct. You can find more information on how to write good answers [in the help center](#). – Community Bot Feb 25 at 19:36

looks like '@sveltejs/adapter-node' should be replaced with '@sveltejs/vite-plugin-svelte' nowadays – bankangle Jul 12 at 13:58