Direct: +1 888-571-8556
info@mas.systems
https://mas.systems

Installing Express REST Server In Production on Windows Server

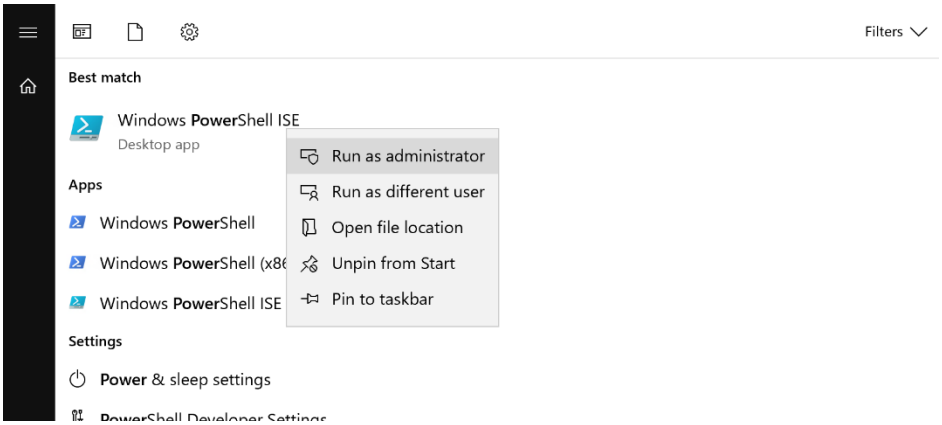
Node.js is the fastest-growing deployment platform for servers in the world. Node.js uses a non-blocking threading model and is one of the fastest servers in the world. It also runs JavaScript and TypeScript natively. Two of the most popular programming languages in the world (what companies are using for new projects).

Node.js, Express.js, pm2, and Nginx run best on Linux. Linux threading model is more in line with Node.js and the server is just a server. Windows Server is dual-mode: it is a server and end-user platform and requires some additional configuration to allow this stack to run. Basically, we have a script that will make Node.js, Express.js, and pm2 see the server as a server first and an end-user platform second. Nginx can run on Windows Server but only if it is running under another WebServer. Linux already has enough web server components as part of the regular Linux install that Nginx runs as-is.

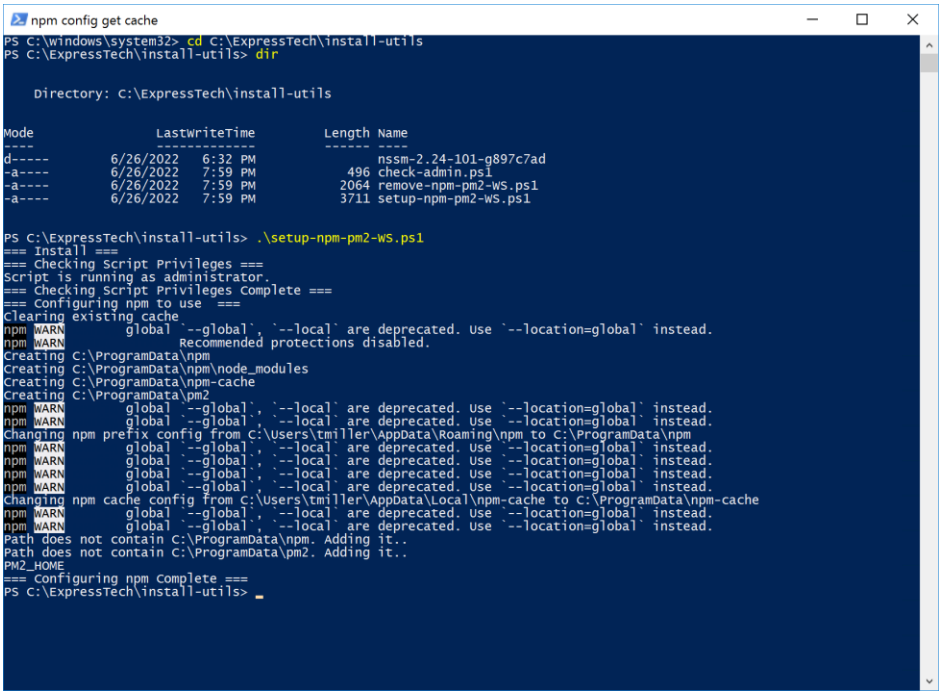
It likely sounds like we are trying to talk you out of using Windows Server. That is not the case, but we also want you to know that if you are looking for high capacity capability (1000 concurrent connections at one time), we would highly recommend Linux. Many will only need 50 - 100 concurrent connections and the setup we describe below will be more than adequate.

Install the Node.js JavaScript Server

- Install Node.js LTS ver 16.15 for Windows Server (do not install optional tools) - download
- Run the installer EmServicesInstall.exe. This installer includes the EM Windows Services. If you have them installed, these will need to be shut down for the installer to run properly.
- Run PowerShell as administrator



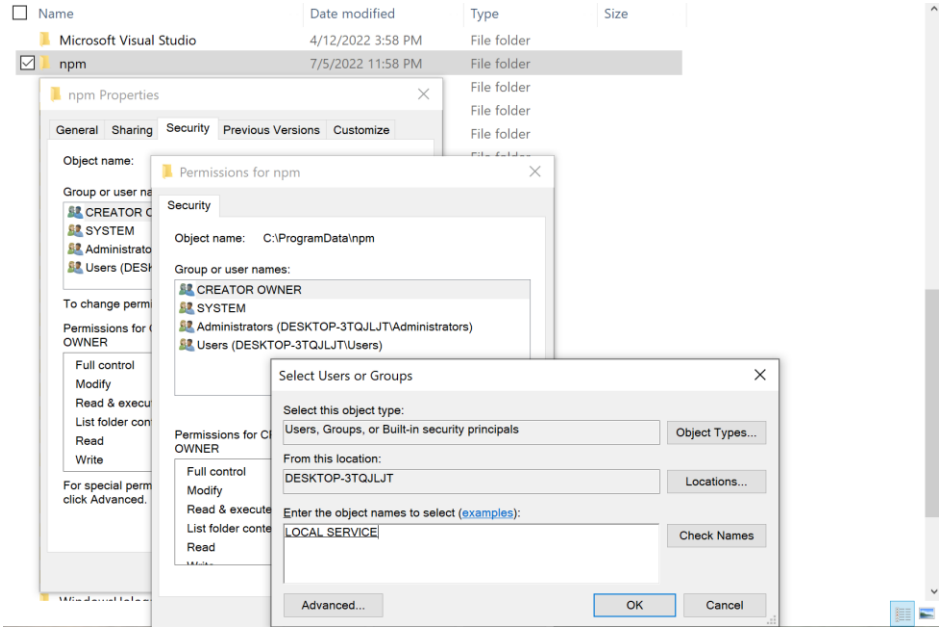
- You will need to change the directory to the environment update script (assumes the default install directory): cd C:\ExpressTech\install-utils *PowerShell doesn't like paths with spaces in it. If you do have spaces, just copy the* ***install-utils*** *directory to the "Temp" folder (C:\Temp\install-utils) and run all the install scripts from there.*
- Run the setup script: .\setup-npm-pm2-WS.ps1



You can ignore the **npm WARN** on the global directive. Even if you do use the new directive, it still warns you. You also may want to **reboot the server** so all the environment changes are picked up properly.

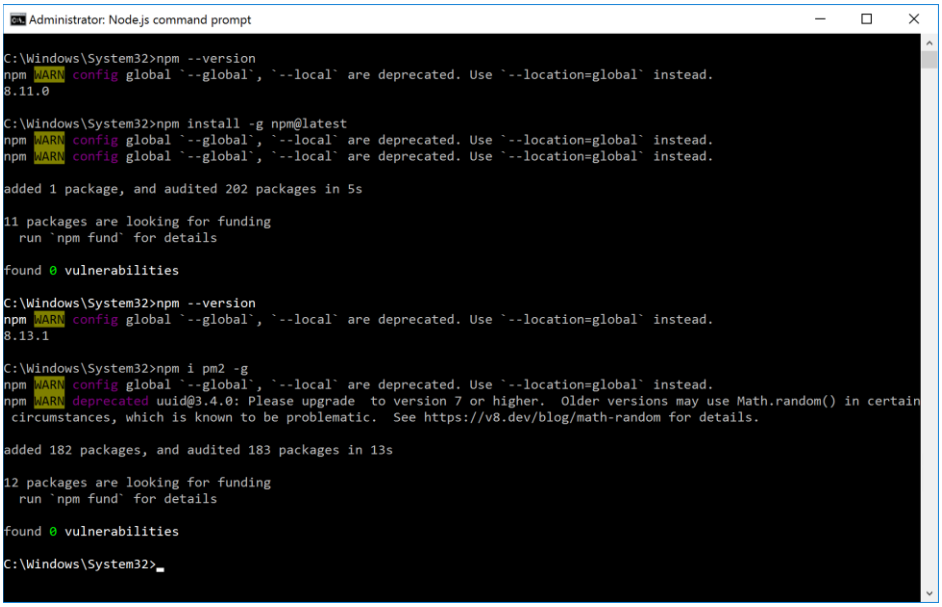
Direct: +1 888-571-8556
info@mas.systems
https://mas.systems

- The script changes the default directory for NPM and PM2 to a server-level directory. This is done with npm settings, environment variables, and search paths. Close the PowerScript window.
- The service will use the **ProgramData\npm** directory and we will make sure it has all the rights it needs to function.
    - Open the directory property option and select the security tab.
    - Click the Edit button.
    - Click the Add Button
    - Enter LOCAL SERVICE and click the Check Name button.
    - Click OK and on the permissions screen select Full Control
    - Click OK again and close out of the properties windows.



- Open Node.js command prompt “As Administrator”
- Run **node --version** (should be v16.15.x)
- Run **npm –-version** (should be v8.11 or something similar)
- Run **npm install -g npm@latest** (will update to the latest version of npm)
- Run **npm i pm2 -g** (will install the latest version of pm2)

Direct: +1 888-571-8556
info@mas.systems
https://mas.systems

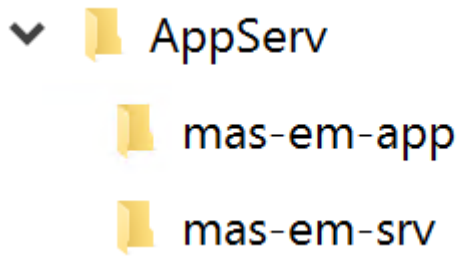


This sets up the server to run our new JavaScript servers. These servers will continue to be developed and expanded. It will support remote APIs, as a data server for Mobile v2.0, ERO v2.0, Dashboard 2.0, and other capabilities not yet announced. ***This setup process does not need to be done again.***

Install / Update Modules for Each Server

Even though you don't need to do the full setup again. You will need to update the modules from time to time. This includes the ones supporting node directly (npm and pm2). We also need to install the modules that support each JavaScript server.

- Open Node.js command prompt "As Administrator" (if not already open)
- Change the directory of each service directory: cd C:\AppServ\nodejs\mas-em-srv
- Run: npm install (this will read the manifest file for the project *package.json* and install all the required modules)



Direct: +1 888-571-8556
info@mas.systems
https://mas.systems

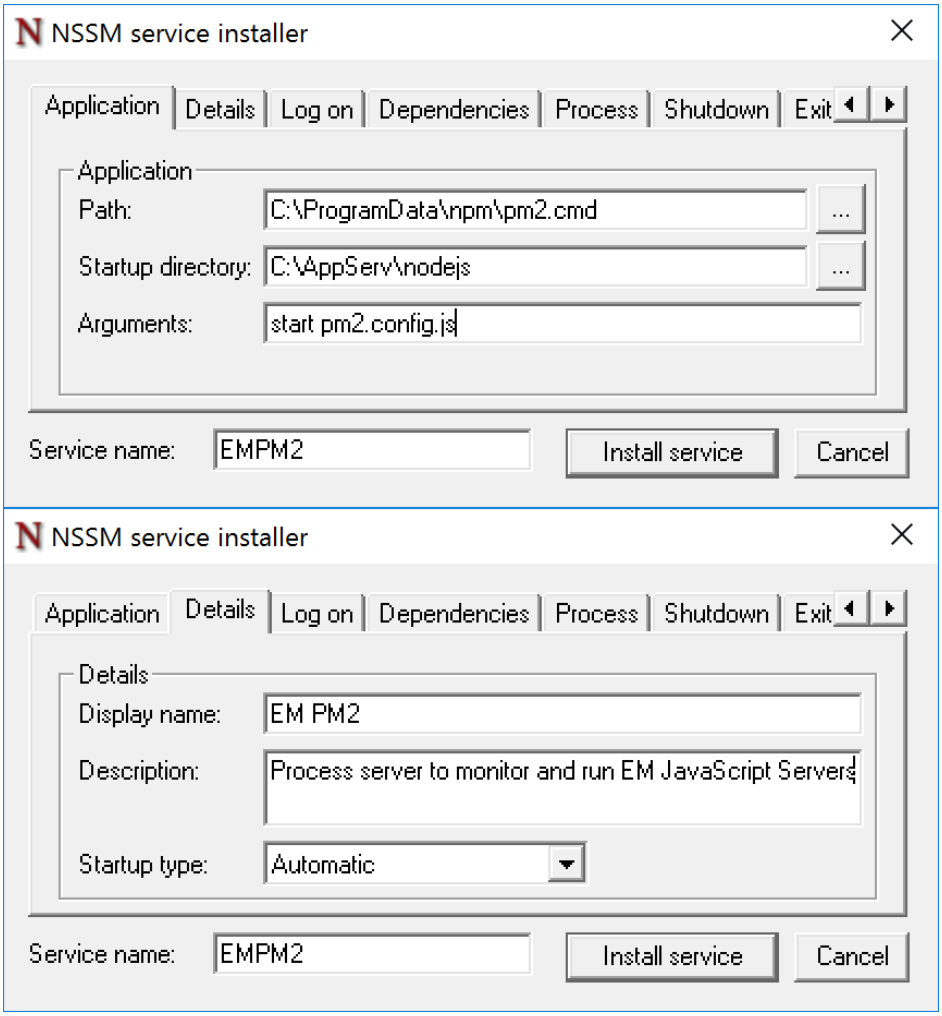
- You should start the service manually one time. This seems to make running it as service run smoother.
- Run: cd C:\AppServ\nodejs
- Run: pm2 start pm2.config.js
- Once the server has started successfully, you should run: pm2 kill

Installing PM2 as a Service

We are using the open-source project ***NSSM***. You can find the commands to manage it here. Let's install ***pm2*** as a service. It is important to use the name ***EMPM2***. This name is a hidden name and takes some work to find it. So better to use this name and be able to look it up here if you ever need to uninstall the service.

- Open Node.js command prompt "As Administrator" (if not already open)
- Run: nssm install EMPM2
- A configuration screen will pop up. Fill in the **Application** and **Details** tabs like below.
- Go to the **Exit action** tab and change the **Restart** setting to ***No Action***.
- Install the service.
- You will need to start the service manually or restart the server (recommended).

Direct: +1 888-571-8556
info@mas.systems
https://mas.systems



PM2 has many capabilities including status and log. You can find other useful pm2 commands here.

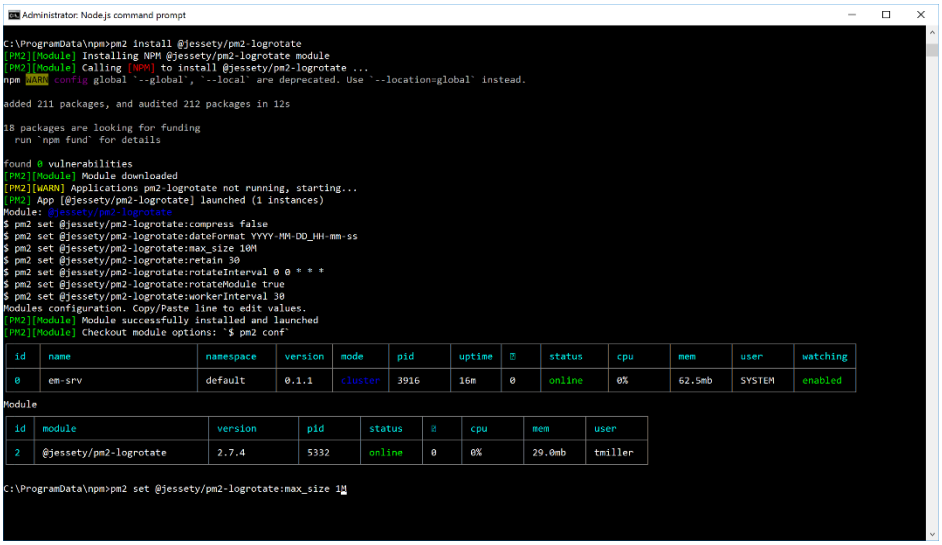
Setting up Log Management

PM2 has great logging in place. The upside is when something goes wrong you have the data you need (and MAS) to locate the problem. The downside is how much hard disk space gets eaten up by the logging. We recommend you install a log management system.

- Open Node.js command prompt "As Administrator"
- Change directory to the C:\ProgramData\npm
- Run: pm2 install @jessety/pm2-logrotate (***yes pm2*** **NOT** *npm*)

It will install and run the log rotate module. Your screen should look something like this:

Direct: +1 888-571-8556
info@mas.systems
https://mas.systems



This also shows all the defaults that are set and you can change. Documentation can be found here. We recommend that you change the default log to 1MB. Run: *pm2 set @jessety/pm2-logrotate:max_size 1M*

Configuring the Database Connection

To configure the DB connectivity, you will need to update the ***multi-tenant-edit.json*** file. This can be found in C:\AppServ\nodejs\mas-em-srv .Then you will call the endpoint **http://localhost:4100/util/proctenant** ({server:port}/util/proctenant). This endpoint will encrypt any newly added configuration (user and password) and then test all the DB connections. After testing is complete, it will copy the file to the ***multi-tenant.json*** file. This will minimize the file locks on the ***multi-tenant.json*** file due to editing it live.

```
[
  {
    "key": "EABC4BBF-E865-4E6F-BEB0-9D59007F1A22",
    "db": "mssql",
    "server": "DESKTOP-SQLSERVER",
    "port": "1433",
    "database": "MyTestDB",
    "username": "admin",
    "password": "123ABCabc789"
  },
  {
    "key": "EABC4BBF-E865-4E6F-BEB0-9D59347F3B68",
    "db": "mdb",
    "server": "localhost",
    "database": "mas-auth-test",
    "username": "admin",
    "password": "123ABCabc789"
  },
```

]

The “key” is the database’s ***dbo.default.UniqueDatabaseID*** field. You can also find this value on the “Connection Info” page in EM. “db” is the database type: “MS SQL Server” or “MariaDB”. The rest of the fields are self-explanatory.

Email Server Job Scheduler

The new nodejs server has a built-in Email Server. This server reads email data from the database and then emails it. It runs every 2.5 minutes. By default, the job is turned off. If you want to use the Email Server, please configure the database connection and Email Server first (found in Express Maintenance in the new API configuration tab). Once these are configured, you can turn on the Email Server Job Scheduler.

To turn on the job, you need to set the EM_EMAIL_JOB environment variable. You can do this by opening the Windows Settings and search for “Edit System”. Select “Edit the System Environment Variables”. On the popup, select the “Environment Variables” button. Find the EM_EMAIL_JOB entry in the “System Variables” section. If it doesn’t exist, you can create it. Set the value to “true” (all lower case). Save and exit the subsystem. You will need to restart the server for the new setting to work.