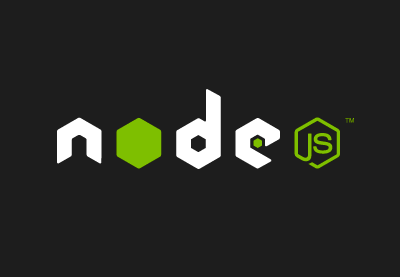
**Como rodar NodeJS em servidor Windows**



Dizem que NodeJS roda melhor em Linux. Como eu não sou lá muito fã de tela preta e queria colocar um projeto no ar de uma vez, acabei optando por rodar em Windows Server mesmo. No entanto deu um pouco mais de trabalho do que eu previa. De qualquer forma, resolvi organizar a trabalheira aqui para ajudar quem estiver nos mesmos apuros ou para eu mesmo me lembrar no futuro, hehehe.

Saliento que minha aplicação está usando [Express](http://expressjs.com/pt-br/), o que pode mudar alguma coisa em relação ao passo-a-passo, dependendo do seu projeto, mas não sei exatamente o quê…

O quê, você não faz a mínima ideia do motivo de eu estar usando NodeJS? Dá uma olhada então nesse [post aqui](http://www.luiztools.com.br/post/por-que-aprender-nodejs/) que você vai entender.

Se o que você estava buscando era um tutorial de como PROGRAMAR em NodeJS, o link certo é [esse aqui](http://www.luiztools.com.br/post/tutorial-nodejs-com-mongodb/).

Os passos para conseguir rodar uma aplicação Node.js no Windows são:

1. [Derrubar o IIS](http://www.luiztools.com.br/post/como-rodar-nodejs-em-servidor-windows/#1)
2. [Desativar Web Deploy](http://www.luiztools.com.br/post/como-rodar-nodejs-em-servidor-windows/#2)
3. [Liberar o Firewall](http://www.luiztools.com.br/post/como-rodar-nodejs-em-servidor-windows/#3)
4. [Configurar a porta 80](http://www.luiztools.com.br/post/como-rodar-nodejs-em-servidor-windows/#4)
5. [Instalar o Node.js](http://www.luiztools.com.br/post/como-rodar-nodejs-em-servidor-windows/#5)
6. [Instalar o Windows Service](http://www.luiztools.com.br/post/como-rodar-nodejs-em-servidor-windows/#6)

**Passo 1: Derrubar o IIS**

Se você tem o IIS instalado no seu servidor, o primeiro passo é parar ele, para que possamos usar a porta 80 do IIS para nossa aplicação NodeJS (afinal, você não vai querer o seu usuário tendo de digitar a porta 3000 ao lado da sua URL, certo?). Você pode pausar o IIS rodando o seguinte comando com permissão de administrador no Executar/Run do Windows (Windows + R).

|  |  |
| --- | --- |
|  | **net stop WAS** |

Ele vai perguntar se você tem certeza, basta confirmar com Y.

**Passo 2: Desativar Web Deploy**

Se você tem instalado Web Deploy no seu servidor (eu tinha), ele também estará ocupando a porta 80, temos de derrubar ele também. Para isso, abra o Executar/Run novamente e digite:

|  |  |
| --- | --- |
|  | **services.msc** |

**I**sso vai abrir o gerenciador de serviços do Windows. Procure pelo serviço Web Deployment Agent Service e dê um Stop nele. Mais que isso, altere para que ele não inicie automaticamente com o Windows, para evitar problemas futuros.

**Passo 3: Liberar o Firewall**

O próximo passo é desbloquear a porta 80 no Firewall do Windows. Na caixa Iniciar do Windows digite Firewall e irá aparecer o ícone correto. Abra o gerenciador do Firewall e clique em Inbound Rules, lá, adicione uma nova regra para permitir acesso à porta 80, configurando a mesma na aba Ports and Protocols. Confirme tudo antes de sair.

**Passo 4: Configurar a porta 80**

Por padrão os apps NodeJS rodam na porta 3000 (ao menos usando Express). Se quiser que ele rode na porta 80, além de preparar o servidor como fiz acima, deve alterar o arquivo www dentro da pasta bin do seu projeto Express. Dentro dele (abra com o notepad) verá que a porta 3000 é citada em uma dada linha, apenas altere para 80.

**Passo 5: Intalar o Node.js**

Baixe e instale no servidor a versão mais recente disponível para Windows no [site oficial do NodeJS](http://www.nodejs.org/).

Depois de instalado, vá novamente no Executar/Run e digite:

|  |  |
| --- | --- |
|  | node –v |

Isso deve lhe devolver a versão do NodeJS instalada no servidor. Se não funcionar, reinicie o servidor e tente novamente.

Publique a sua aplicação NodeJS em uma pasta do servidor. Minha sugestão é C:\domains\nomedoapp\www.

Abra uma janela do CMD com permissão de administrador e navegue usando cd até a pasta da sua aplicação NodeJS. Exemplo:

|  |  |
| --- | --- |
|  | **cd C:\domains\nomedoapp\www** |

Uma vez lá, digite o seguinte comando para instalar as dependências do seu projeto:

|  |  |
| --- | --- |
|  | **npm install** |

Isso deverá ser o suficiente, mas se quiser ter certeza de que tudo que necessita para rodar NodeJS está instalado em seu servidor, execute também os seguintes comandos em sequência:

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9** | **npm install express –g**  **npm install url -g**  **npm install fresh -g**  **npm install cookie -g**  **npm install methods -g**  **npm install crc -g**  **npm install send -g**  **npm install connect -g**  **npm install commander -g** |

Agora finalmente execute seu app NodeJS com o comando abaixo ainda “dentro” da pasta do seu projeto (ou navegue novamente usando cd):

|  |  |
| --- | --- |
|  | **npm start** |

**Npm start**

**Passo 6: Instalar o Windows Service**

Por padrão, o seu site NodeJS ficará executando enquanto a janela do console que iniciou seu site estiver funcionando. Se você fechar ela, ou seu servidor reiniciar (malditos Windows Updates!) já era, seu site vai ficar fora do ar até que você execute o npm start de novo.

Para que isso não aconteça, você deve instalar seu site como um Windows Service. Para fazer isso, primeiro instale o [módulo node-windows](https://github.com/coreybutler/node-windows) globalmente:

|  |  |
| --- | --- |
|  | **npm install -g node-windows** |

Agora rode o seguinte comando (dentro da pasta do seu projeto) para incluir uma referência desse módulo ao seu projeto:

|  |  |
| --- | --- |
|  | **npm link node-windows** |

Depois, dentro do seu projeto NodeJS (na raiz mesmo) crie um arquivo service.js com o seguinte conteúdo:

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16** | **var Service = require('node-windows').Service;**    **// Create a new service object**  **var svc = new Service({**  **name:'Nome da sua Aplicação',**  **description: 'Apenas uma descrição',**  **script: 'C:\\domains\\sitenodejs\\bin\\www'**  **});**    **// Listen for the "install" event, which indicates the**  **// process is available as a service.**  **svc.on('install',function(){**  **svc.start();**  **});**    **svc.install();** |

Troque as propriedades name e description de acordo com seu gosto, mas atente à propriedade script nesse código, que deve conter o caminho absoluto até o arquivo JS que inicia sua aplicação. No meu caso, como estou usando express, estou apontando para o arquivo www que fica na pasta bin do projeto (curiosamente ele não possui extensão, mas é um arquivo).

Se você fez tudo corretamente, vá até Ferramentas Administrativas > Serviços (Administrative Tools > Services ou services.msc no Run/Executar do Windows) e seu serviço vai aparecer lá com o name que definiu ali no script, lhe permitindo alterar suas configurações de inicialização, dar Start, Stop, etc.

Outra coisa bacana desse módulo é que ele manda todas as mensagens para o Event Viewer/Visualizador de Eventos do Windows. Assim, caso sua aplicação NoeJS não esteja funcionando ou seu serviço não esteja inicializando, basta dar uma olhada lá, na seção Windows > Applications para entender o que está acontecendo.

Caso precise remover esse serviço (para instalar uma versão mais atualizada, por exemplo) rode o comando abaixo no cmd:

|  |  |
| --- | --- |
| **1**  **2** | **SC STOP servicename**  **SC DELETE servicename** |

Esse servicename você encontra nas propriedades do Windows Service que deseja excluir.

Espero que tenha funcionado com você também!

O que achou desse artigo?