

Stack Overflow em Português é um site de perguntas e respostas para programadores profissionais e entusiastas. Leva apenas um minuto para se inscrever.



Inscreva-se para participar desta comunidade

Qualquer pessoa pode fazer uma pergunta

Qualquer um pode responder

As melhores respostas recebem votos positivos e sobem para os primeiros lugares

Executar um script em shell com NodeJS

Perguntada 3 anos, 7 meses atrás Ativa 3 anos, 6 meses atrás Vista 3mil vezes



1



Estou precisando criar uma página usando nodejs, que conterà um botão "Liga / Desliga". Ao pressionar este botão pretendo fazer a chamada da mesma rota, porém passando um parâmetro: liga/desliga ou 0/1, etc.. Ao fazer essa chamada, vou tratar o parâmetro recebido e então executar um script chamado "ligar.js" ou "desligar.js". Estes scripts estarão na mesma pasta do servidor. Imaginei que em "File System" haveria algo como a função "exec()" do PHP, mas não encontrei. No PHP, por exemplo, poderia ser feito algo como `exec("/meu_path/meu_script.js $onOff");`. Como faço essa execução de um script no Node?

javascript

node.js

compartilhar melhorar esta pergunta seguir

perguntada 18/07/17 às 2:06



wBB

508 4 23

Acho que isso resolve seu problema udgwebdev.com/node-js-para-leigos-child-process – Rafael Salomão 18/07/17 às 5:50

Rafael, grato pelo retorno. Acho que isso resolve o problema e achei muito útil. Porém acabei fazendo apenas com o `require` como sugeriu o colega Sérgio na outra resposta. Já consegui daquela forma. Obrigado! – wBB 18/07/17 às 13:45

tranquilo, se resolveu fico feliz! – Rafael Salomão 18/07/17 às 13:59

1 Resposta

Ativos	Oldest	Votos
--------	--------	-------



1



Se queres executar um script `.js` o melhor é criares funções nesses scripts e fazer `require` ou `import` desses scripts para poderes correr essas funções quando for preciso.

Se os scripts não forem JavaScript podes fazer assim, uma sugestão que aceita vários comandos e os corre sequencialmente:

```
"use strict;"

var exec = require('child_process').exec;

const commands = [
  'sudo comando1',
  'sudo comando2'
];

function runCommand(cmds, cb){
  const next = cmds.shift();
  if (!next) return cb();
  exec(next, {
    cwd: __dirname
  }, (err, stdout, stderr) => {
    console.log(stdout);
    if (err && !next.match(/\-s$/)) {
      console.log(`0 commando "${next}" falhou.`, err);
      cb(err);
    }
    else runCommand(cmds, cb);
  });
}

runCommand(commands, err => {
  console.log('Script corrido');
});
```

Executar

[Expandir trecho de código](#)

Se quiseres correr somente 1 comando podes simplificar e fazer assim:

```
"use strict;"

var exec = require('child_process').exec;
const cmd = 'sudo comando1';

exec(cmd, {
  cwd: __dirname
}, (err, stdout, stderr) => {
  console.log(stdout);
  if (err) console.log(err);
  else runCommand(cmds, cb);
});
```

```
else runCommand(cmds, cd);  
});
```

Executar

[Expandir trecho de código](#)

compartilhar melhorar esta resposta

editada 19/07/17 às 9:19

respondida 18/07/17 às 8:26

seguir



Sergio ♦

132mil

16

183

275

Sérgio, você tem razão. apenas o `require` já resolve, mas de qualquer forma o que você e o colega Rafael sugeriram sobre o `child_process` é muito legal e útil. Obrigado. – wBB 18/07/17 às 13:47

@wBB ótimo que ajudou. O script que coloquei aqui é para vários comandos. Juntei agora uma versão simplificada para somente 1 comando. – Sergio ♦ 19/07/17 às 9:19
