Path de arquivo Node.js

Todo arquivo no sistema tem um caminho.

No Linux e no macOS, um caminho tem essa cara:

/users/pauloluan/file.txt

enquanto que em computadores Windows é bem diferente, a estrutura é algo assim:

C:\users\pauloluan\file.txt

Você precisa prestar atenção quando está utilizando caminhos na sua aplicação, as diferenças devem ser levadas em consideração.

Para trabalhar com caminhos, inclua esse módulo em seus arquivos:

const path = require('path')

**Obtendo informações de um caminho**

Dado um caminho, você pode extrair informações dele usando esses métodos:

* dirname: obtêm a pasta pai do arquivo
* basename: obtêm a parte do nome do arquivo
* extname: obtêm a parte da extensão do arquivo

Exemplo:

const notes = '/users/pauloluan/notes.txt'

path.dirname(notes) // /users/pauloluan

path.basename(notes) // notes.txt

path.extname(notes) // .txt

Você pode obter o nome do arquivo sem a extensão ao especificar um segundo argumento no basename:

path.basename(notes, path.extname(notes)) // notes

**Trabalhando com caminhos**

Você pode unir duas ou mais partes de um caminho ao utilizar path.join():

const name = 'pauloluan'

path.join('/', 'users', name, 'notes.txt') // '/users/pauloluan/notes.txt'

Você pode obter o cálculo do caminho absoluto de um caminho relativo utilizando path.resolve():

path.resolve('pauloluan.txt') // '/Users/pauloluan/pauloluan.txt' se rodar a partir da minha pasta home

Nesse caso o Nodejs vai simplesmente adicionar /pauloluan.txt na pasta atual. Se você especificar uma pasta como segundo parâmetro, o resolve vai usar o primeiro como base pro segundo:

path.resolve('tmp', 'pauloluan.txt') // '/Users/pauloluan/tmp/pauloluan.txt' se rodar a partir da minha pasta home

Se o primeiro parâmetro começa com uma barra, isso implica que é um caminho absoluto:

path.resolve('/etc', 'pauloluan.txt') // '/etc/pauloluan.txt'

path.normalize() é outra função bem útil, que tentará calcular o caminho real, quando ele contêm especificadores relativos como . ou .., ou barras duplas:

path.normalize('/users/pauloluan/..//test.txt') // /users/test.txt

**Ambos resolve e normalize não irão validar se o caminho existe**. Eles só calculam um caminho baseado nas informações fornecidas.