

O que é process.env no Node.js?

O process.env é uma propriedade do Node.js, não do dotenv. O que o dotenv faz é carregar as variáveis do arquivo .env para dentro do process.env, tornando-as acessíveis dentro do código.

Exemplo de uso:

Se você tem um arquivo .env assim:

```
DATABASE_STRING_CONNECTION=mysql://user:pass@localhost:3306/dbname
```

E carrega ele no seu código assim:

```
require('dotenv').config();  
  
console.log(process.env.DATABASE_STRING_CONNECTION);
```

O dotenv apenas preenche process.env com os valores do .env, mas process.env sempre pertence ao Node.js.

O que é process?

O process no Node.js é um objeto global que representa o processo em execução do Node. Ele fornece informações e métodos para interagir com o ambiente onde o código está sendo executado.

Principais Características de process:

- process.env - Contém variáveis de ambiente do sistema e do próprio Node.js.
- process.argv - Contém os argumentos passados na execução do script.
- process.exit(code) - Encerra o processo com um código de saída.
- process.cwd() - Retorna o diretório de trabalho atual.
- process.pid - Retorna o ID do processo.

Exemplo:

```
console.log(process.env); // Exibe todas as variáveis de ambiente
```

```
console.log(process.cwd()); // Exibe o diretório atual
```

```
console.log(process.pid); // Exibe o ID do processo
```

Ou seja, o process faz parte do próprio Node.js e está sempre disponível sem precisar de require.

Como o process sabe que é do dotenv?

O process.env não sabe que as variáveis vêm do dotenv. O que acontece é que o dotenv simplesmente lê o arquivo .env e adiciona as variáveis a process.env, que já existe no Node.js.

Quando você chama:

```
require('dotenv').config();
```

O dotenv faz o seguinte:

1. Lê o arquivo .env que está no diretório do projeto.
2. Interpreta as variáveis definidas nele (como DATABASE\_STRING\_CONNECTION=mysql://...).
3. Adiciona essas variáveis ao process.env, que já existe no Node.js.

Depois disso, quando você acessa:

```
console.log(process.env.DATABASE_STRING_CONNECTION);
```

O Node.js apenas pega o valor que foi injetado pelo dotenv em process.env, mas o process em si não tem relação direta com o dotenv. O dotenv apenas preenche process.env com novas variáveis.

Se não chamarmos require('dotenv').config();, o process.env ainda existe, mas não terá as variáveis do .env, apenas as que já estiverem definidas no sistema operacional.

O que é `.config()`?

O `.config()` é um método do módulo `dotenv` que lê o arquivo `.env`, carrega suas variáveis e as adiciona ao `process.env`.

Como funciona?

Quando você faz:

```
require('dotenv').config();
```

Isso equivale a chamar o método `.config()` do `dotenv`, que executa os seguintes passos:

1. Lê o arquivo `.env` que está no diretório raiz do projeto.
2. Interpreta as variáveis que estão no formato `CHAVE=VALOR`.
3. Adiciona essas variáveis ao `process.env`, para que possam ser acessadas no código.

Se o arquivo `.env` contiver:

```
DATABASE_URL=mysql://user:pass@localhost:3306/dbname
```

```
PORT=3000
```

Depois de chamar `require('dotenv').config()`, essas variáveis ficam acessíveis via:

```
console.log(process.env.DATABASE_URL); // mysql://user:pass@localhost:3306/dbname
```

```
console.log(process.env.PORT); // 3000
```

Personalização

O `.config()` pode receber um objeto de configuração, como:

```
require('dotenv').config({ path: './meu-arquivo.env' });
```

Isso permite carregar variáveis de outro arquivo .env.

Se você não chamar .config(), o dotenv não faz nada e as variáveis do .env não estarão disponíveis no process.env.

O que é uma variável de ambiente?

Uma variável de ambiente é um valor armazenado no sistema operacional que pode ser acessado por programas em execução. Elas são usadas para configurar e personalizar o comportamento de aplicativos sem alterar o código-fonte.

Exemplos de uso:

1. Configurações sensíveis (senhas, chaves de API, conexões de banco de dados).
2. Definir o ambiente da aplicação (desenvolvimento, teste, produção).
3. Configurar portas de servidores ou caminhos de diretórios.

Como acessar variáveis de ambiente no Node.js:

No Node.js, as variáveis de ambiente ficam dentro de process.env.

```
console.log(process.env.PATH); // Mostra o PATH do sistema
```