Exercício prático: Criando um projeto Node.js com o npm

Passo 1: Configurando o ambiente

Certifique-se de ter o Node.js e o *npm* instalados em seu computador. Você pode verificar a instalação digitando os seguintes comandos em seu terminal ou prompt de comando:

```
node -v
npm -v
```

Passo 2: Criando o projeto

Crie uma nova pasta para o projeto e navegue até ela no terminal:

```
mkdir projeto-npm
cd projeto-npm
```

Passo 3: Inicializando o projeto com o npm

Dentro da pasta do projeto, execute o seguinte comando para inicializar o projeto e criar o arquivo *package.json*:

```
npm init
```

Siga as instruções para preencher as informações necessárias, ou simplesmente pressione *Enter* para aceitar as configurações padrão.

Passo 4: Instalando dependências

Agora, vamos instalar algumas dependências para nosso projeto. Vamos utilizar o pacote *lodash*, uma biblioteca de utilitários *JavaScript* muito popular.

Execute o seguinte comando para instalar o pacote *lodash* como uma dependência do projeto:

```
npm install lodash
```

Passo 5: Criando um arquivo JavaScript

Crie um arquivo chamado *app.js* dentro da pasta do projeto e adicione o seguinte código:

```
// app.js
const _ = require('lodash');

const numbers = [1, 2, 3, 4, 5];
const sum = _.sum(numbers);

console.log('A soma dos números é:', sum);
```

Passo 6: Executando o código

Agora, execute o arquivo *app.js* usando o *Node.js*:

```
node app.js
```

Você deve ver a mensagem "A soma dos números é:" seguida da soma dos números do *array*.

Passo 7: Criando um *script* de execução

Vamos adicionar um *script* no arquivo *package.json* para facilitar a execução do nosso código.

Abra o arquivo *package.json* em um editor de texto e adicione o seguinte código:

```
"scripts": {
  "start": "node app.js"
}
```

Agora você pode executar o *script* usando o seguinte comando:

```
npm start
```

Isso executará automaticamente o arquivo app.js.

Parabéns! Você completou o exercício prático sobre o Node Package Manager (npm). Agora você criou um projeto Node.js, instalou uma dependência usando o npm e executou um arquivo JavaScript dentro do projeto.

O npm é uma ferramenta poderosa para gerenciar dependências e tornar o desenvolvimento de projetos Node.js mais eficiente. Continue explorando e aprendendo mais sobre as funcionalidades do npm!