## **Requisitos e Passo a Passo para Consumir a API da Marvel com Node.js, Express e Nuxt.js**

### **Objetivo**

Desenvolver uma aplicação que utilize a API da Marvel para buscar informações sobre personagens, quadrinhos, etc., utilizando Node.js como backend com o framework Express.js e Nuxt.js para o frontend.

### **Pré-requisitos**

* Conhecimento básico de Node.js, Express.js e Nuxt.js
* Conta na plataforma da Marvel para obter suas chaves API
* Ambiente de desenvolvimento configurado (Node.js, npm ou yarn, editor de código)

### **Passo a Passo**

#### **1. Configuração do Projeto Backend (Node.js + Express)**

**Criar um novo projeto:**Bash:  
 mkdir marvel-app

cd marvel-app

npm init -y

**Instalar as dependências:** Bash:  
 npm install express axios dotenv

* + express: Framework para criar o servidor.
  + axios: Biblioteca para fazer requisições HTTP.
  + dotenv: Para carregar as chaves API do arquivo .env.

**Criar o arquivo .env:** PUBLIC\_KEY=YOUR\_PUBLIC\_KEY

PRIVATE\_KEY=YOUR\_PRIVATE\_KEY

**Criar o arquivo index.js:** JavaScript  
require('dotenv').config();

const express = require('express');

const axios = require('axios');

const app = express();

const port = process.env.PORT || 3000;

const baseUrl = 'http://gateway.marvel.com/v1/public';

// Função para fazer as requisições à API da Marvel

const getMarvelData = async (endpoint, params) => {

const ts = Date.now();

const hash = crypto.createHash('md5').update(ts + process.env.PRIVATE\_KEY + process.env.PUBLIC\_KEY).digest('hex');

const response = await axios.get(`<span class="math-inline">\{baseUrl\}/</span>{endpoint}`, {

params: {

ts,

apikey: process.env.PUBLIC\_KEY,

hash,

...params

}

});

return response.data.data.results;

};

// Exemplo de rota para buscar personagens

app.get('/characters', async (req, res) => {

const characters = await getMarvelData('characters', req.query);

res.json(characters);

});

app.listen(port, () => {

console.log(`Servidor rodando na porta ${port}`);

});

#### **2. Configuração do Projeto Frontend (Nuxt.js)**

**Criar um novo projeto:** Bash  
npx create-nuxt-app nuxt-app

* **Instalar as dependências (opcional):** Dependendo das funcionalidades desejadas, você pode instalar bibliotecas adicionais como Vuex para gerenciamento de estado, Vue Router para roteamento, etc.

#### **3. Consumir a API no Frontend**

**Criar uma página ou componente:** Snippet de código  
<template>

<div v-for="character in characters" :key="character.id">

<h2>{{ character.name }}</h2>

<img :src="character.thumbnail.path + '.' + character.thumbnail.extension" alt="Thumbnail">

</div>

</template>

<script>

export default {

data() {

return {

characters: []

}

},

async mounted() {

const response = await fetch('http://localhost:3000/characters');

const data = await response.json();

this.characters = data;

}

}

</script>

### **Próximos Passos**

* **Implementar mais rotas:** Crie rotas para buscar quadrinhos, eventos, etc.
* **Gerenciar o estado:** Utilize Vuex para gerenciar o estado da aplicação, como os personagens favoritos.
* **Implementar a busca:** Adicione um campo de busca para filtrar os resultados.
* **Criar detalhes dos personagens:** Mostre informações mais detalhadas sobre cada personagem.
* **Implementar autenticação:** Se necessário, implemente um sistema de autenticação para permitir que os usuários salvem seus personagens favoritos.

### **Considerações**

* **Gerenciamento de chaves API:** Evite expor suas chaves API diretamente no código. Utilize variáveis de ambiente ou um serviço de gerenciamento de segredos.
* **Limitação de requisições:** A API da Marvel possui limites de requisições. Implemente mecanismos de cache ou paginação para otimizar o consumo da API.
* **Tratamento de erros:** Implemente um tratamento de erros adequado para lidar com casos como falhas na conexão, dados inválidos, etc.
* **Design da aplicação:** Crie uma interface intuitiva e agradável para o usuário.

**Este é apenas um ponto de partida.** A personalização da aplicação depende das suas necessidades e da sua criatividade. Explore a documentação da API da Marvel e das ferramentas utilizadas para adicionar mais funcionalidades e melhorar a sua aplicação.

**Recursos úteis:**

* **Documentação da API da Marvel:**<http://developer.marvel.com/>
* **Documentação do Express.js:**<https://expressjs.com/>
* **Documentação do Nuxt.js:**<https://nuxtjs.org/>

Lembre-se de adaptar este guia às suas necessidades específicas e explorar as diversas possibilidades que as tecnologias oferecem.