



## Download da API do projeto

### Transcrição

*Começando deste ponto? Você pode fazer o [DOWNLOAD](https://s3.amazonaws.com/caelum-online-public/vue/stages/03-alurapic.zip) (<https://s3.amazonaws.com/caelum-online-public/vue/stages/03-alurapic.zip>) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta `alurapic` e executar o comando `npm install` para baixar novamente todas as dependências da aplicação.*

Nossa aplicação consegue exibir uma lista de fotos fixa, mas sabemos que em uma aplicação real os dados das fotos devem vir de um servidor, ou melhor, de uma API disponibilizada por um servidor. Uma API nada mais é do que um endereço único na Web que é capaz de executar uma série de ações e também receber e enviar dados.

Baixe o arquivo [api.zip](https://s3.amazonaws.com/caelum-online-public/vue/api.zip) (<https://s3.amazonaws.com/caelum-online-public/vue/api.zip>) que contém a API do curso feita em Node.js. Depois de baixar, descompacte o arquivo na sua área de trabalho. Em seguida, abra um **novo terminal**. Neste terminal, levante o servidor que disponibiliza a API utilizada pelo nosso projeto através do comando:

```
npm start
```

[COPIAR CÓDIGO](#)

Se tudo correr bem, será exibida a seguinte mensagem no console:

API escutando na porta: 3000

COPIAR CÓDIGO

Por fim, para verificar se a API esta retornando dados acesse o endereço `http://localhost:3000/v1/fotos` . Isso fará com que uma lista de fotos seja exibida em seu navegador.

Quando usamos o endereço anterior, o navegador automaticamente utiliza o método GET. Nesse sentido, nossa API esta preparada para quando o endereço for este e o método for GET responder com uma lista de fotos. Veremos ao longo do curso que nossa API suporta outros métodos para cadastrar novos fotos, inclusive para apagá-las.

Veja que até o final do curso trabalharemos com dois terminais. O primeiro, que torna acessível nossa aplicação para o navegador e o outro que disponibiliza uma API para ser consumida por essa aplicação.

## Um pouco sobre a separação do cliente e sua API

Quando criamos uma SPA, é prática comum do mercado hospedar o front-end em um servidor em separado da API consumida por esse servidor. Aliás, uma mesma aplicação SPA criada com Vue pode consumir uma ou mais API's criadas nas mais diversas linguagens, sejam elas Python, Java, C# entre outras. O segredo dessa compatibilidade é que essas API's devem seguir o padrão REST, um padrão fortemente baseado no protocolo HTTP, por isso universal.

Essa separação entre front e da API permite que a mesma API seja consumida por outros tipos de clientes que não sejam um navegador como aplicativos feitos em

Android ou até mesmo uma engenhoca construída com Arduino que envia dados de sensores para a API.

Este curso foca a parte de front-end com Vue e não da API. Confira os treinamentos da Alura sobre API REST e escolha aquele com a que você se sentir mais à vontade em programar.