

Este é o cache do Google de <http://www.vuejs-brasil.com.br/crie-rapidamente-um-projeto-vue-com-vue-cli-e-browserify/>. Ele é um instantâneo da página com a aparência que ela tinha em 14 out. 2017 07:50:27 GMT.

A [página atual](#) pode ter sido alterada nesse meio tempo. [Saiba mais](#)

**Versão completa**    [Versão somente texto](#)    [Ver código-fonte](#)

Dica: para localizar rapidamente o termo de pesquisa nesta página, pressione **Ctrl+F** ou **⌘-F** (Mac) e use a barra de localização.



□ MENU

# Crie rapidamente um projeto Vue com vue-cli

# e browserify

28 JUNE 2016 on Ferramentas

Na maioria das vezes, quando nos deparamos com uma nova tecnologia, queremos de imediato testá-la, ver seu funcionamento na prática. Com o Vue isso não será diferente, e através de uma ferramenta chamada *vue-cli*, podemos criar uma estrutura de projeto simples, mas completa para o início de uma app em Vue.

O único requisito para testar o *vue-cli* é o Node.js instalado, na versão 4.X ou superior, incluindo também o *npm*, o gerenciador de pacotes do Node.

## O que é Node? E NPM?

*Node é uma plataforma de desenvolvimento em javascript, que usa a engine do google chamada V8. Basicamente, você consegue executar aplicações javascript no seu computador. O NPM é o gerenciador desses programas, onde é possível instalar através da linha de comando do terminal do seu sistema operacional.*

Para instalar o *vue-cli*, abra um terminal e digite o seguinte comando:

```
$ npm i -g vue-cli
```

O parâmetro `i` é um acrônimo para `install`, e o parâmetro `-g` para global. Isso diz ao Node que está instalado no seu sistema para instalar globalmente o *vue-cli*, de forma que você possa usá-lo em qualquer diretório.

Com o *vue-cli* instalado, podemos analisar como usá-lo. Para criar um projeto vue, execute o seguinte comando:

```
$ vue init <template-name> <project-name>
```

Existem dois parâmetros que precisamos fornecer ao *vue-cli* para que ele possa criar o projeto. O primeiro deles é o `template-name`, que corresponde a um template previamente criado que será usado como *esqueleto* para o seu projeto. O segundo parâmetro é o `project-name`, o nome do seu projeto que corresponde ao diretório que será criado.

Os principais templates de projeto são:

- webpack – Usa o webpack como module bundler, possui o vue-loader com *hot reload*, javascript lint e testes unitários. É o pacote mais completo existente.
- webpack-simple – Usa o webpack, com menos recursos que o primeiro. Não possui *hot reload*, javascript lint e nem testes unitários. É recomendado para quem está começando com o vue.
- browserify – Ao invés do webpack, usa o browserify como module bundler. Possui o vue-loader com *hot reload*, javascript lint e testes unitários.
- browserify-simple – Mais simples que o anterior, sem hot reload, lint ou testes unitários. Usaremos este como exemplo neste artigo.
- simple – Mais simples, impossível. Possui apenas uma única página com o Vue sendo carregado através de um endereço CDN.

## Webpack ou Browserify ?

Uma pergunta comum é: qual usar? Aqui entra a questão de qual melhor se adapta a sua realidade. Basicamente, o Webpack é mais poderoso e configurável, e talvez por isso exija um pouco mais de configuração. Browserify é mais simples de configurar e com um pouco menos de recursos.

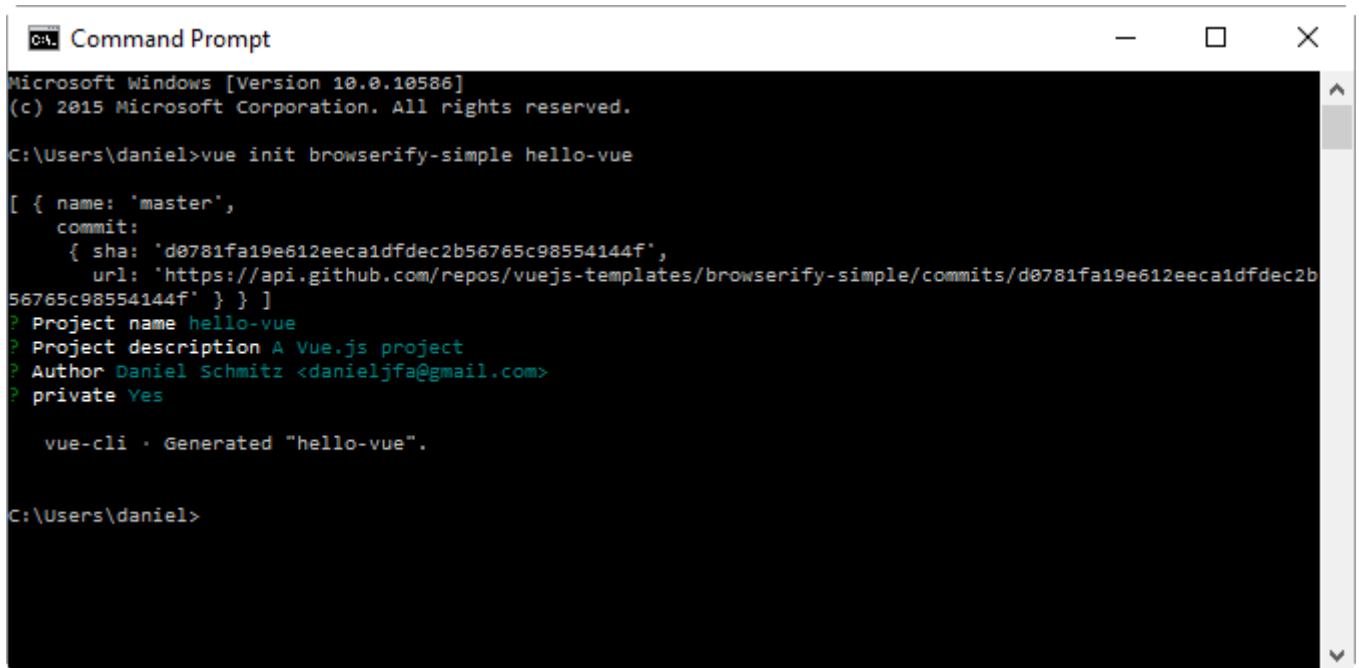
### Alguns termos que você talvez não conheça

- **Module bundler:** Uma forma de agrupar diversos tipos de arquivos em um grupo único, sendo usado no produto final. Por exemplo, é possível criar vários arquivos javascript separados e, quando for usá-los, o bundler irá "juntar" tudo e criar um arquivo único, que poderá inclusive estar minificado.
- **Hot reload:** Usado para atualizar automaticamente o navegador web quando existe alguma alteração no código fonte do projeto
- **Lint:** Usado para validar o seu código javascript, checando por exemplo se existe alguma vírgula onde não deveria, se existe algum ponto-e-vírgula faltando, etc.

Vamos então criar o nosso primeiro projeto Vue. Execute o seguinte comando:

```
$ vue init browserify-simple hello-vue
```

Este comando produzirá o seguinte resultado:



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\daniel>vue init browserify-simple hello-vue

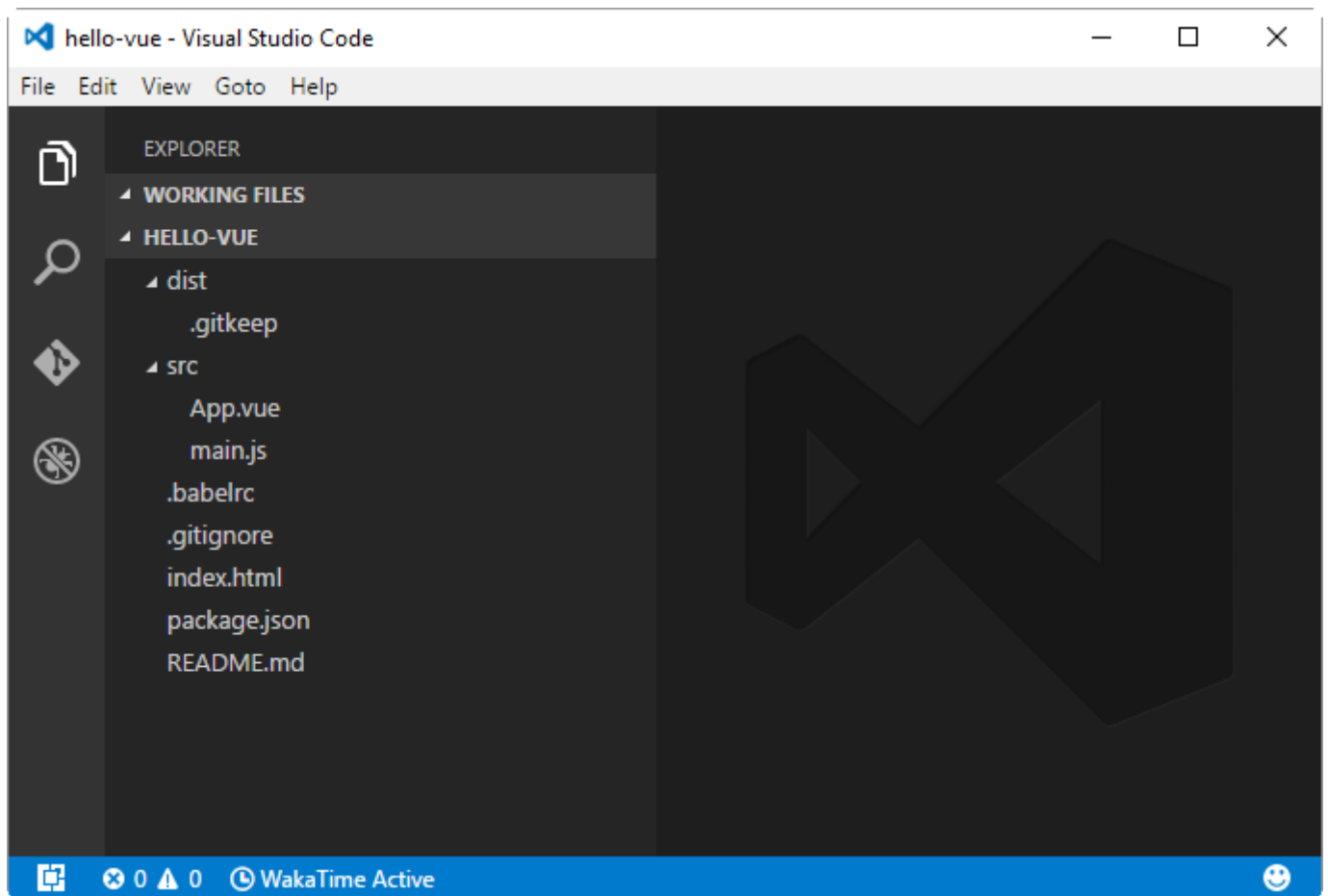
[ { name: 'master',
  commit:
    { sha: 'd0781fa19e612eeca1dfdec2b56765c98554144f',
      url: 'https://api.github.com/repos/vuejs-templates/browserify-simple/commits/d0781fa19e612eeca1dfdec2b56765c98554144f' } } ]
? Project name hello-vue
? Project description A Vue.js project
? Author Daniel Schmitz <danieljfa@gmail.com>
? private Yes

vue-cli - Generated "hello-vue".

C:\Users\daniel>
```

Algumas perguntas são feitas antes do projeto ser criado, tais como o nome do projeto, uma descrição, entre outras.

Após a criação do projeto, o diretório `hello-vue` é criado, com alguns arquivos conforme a imagem a seguir:



Cada arquivo criado possui uma função na configuração do projeto inicial. Vamos comentar cada um deles:

- **dist/** É o diretório onde o arquivo javascript "compilado" irá ficar. Chamamos de "compilado" porque o module bundler escolhido irá "juntar" todos os arquivos javascript em um só, e salvá-lo neste diretório.
- **src/** É o diretório onde a sua aplicação Vue será criada.
- **src/App.vue** É o componente inicial da sua aplicação, carregado pelo arquivo `main.js`
- **src/main.js** É o que chamamos de "bootstrap" da aplicação, ou seja, é onde tudo começa! O *module bundler* começa a "juntar" os arquivos javascript pelo `main.js`.
- **.babelrc** Contém algumas configurações que ajudam o *module bundler* a "transpilar" o código javascript que está

em ES6 para ES5. Isso é necessário porque os navegadores ainda não estão 100% compatíveis com o ES6 (ES2015)

- **.gitIgnore** Usado para ignorar alguns arquivos do controle de versão.
- **index.html** O arquivo html principal da aplicação
- **package.json** O arquivo de configuração do projeto
- **README.md** Um arquivo contendo uma documentação sobre o projeto em si

Após conhecer rapidamente cada arquivo, vamos nos aprofundar em cada um deles. Começando pelo `package.json`, que contém todas as informações do Node/Npm do projeto. Veja:

```
{
  "name": "hello-vue",
  "description": "A Vue.js project",
  "author": "Daniel Schmitz <danieljfa@gmail.com>",
  "private": true,
  "scripts": {
    "watchify": "watchify -vd -p browserify-hmr -e src/main.js -o dist/build.js",
    "serve": "http-server -c 1 -a localhost",
    "dev": "npm-run-all --parallel watchify serve",
    "build": "cross-env NODE_ENV=production browserify src/main.js | uglifyjs -c warnings=false -m > dist/build.js"
  },
  "dependencies": {
    "vue": "^1.0.0"
  },
  "devDependencies": {
```

```
"babel-core": "^6.0.0",
"babel-plugin-transform-runtime": "^6.0.0",
"babel-preset-es2015": "^6.0.0",
"babel-preset-stage-2": "^6.0.0",
"babel-runtime": "^6.0.0",
"cross-env": "^1.0.6",
"babelify": "^7.2.0",
"browserify": "^12.0.1",
"browserify-hmr": "^0.3.1",
"http-server": "^0.9.0",
"npm-run-all": "^1.6.0",
"uglify-js": "^2.5.0",
"vueify": "^8.5.2",
"watchify": "^3.4.0"
},
"browserify": {
  "transform": [
    "vueify",
    "babelify"
  ]
}
}
```

Este arquivo realmente possui muita informação, então vamos quebrá-lo em partes para podermos entender cada pedaço.

Primeiro, temos algumas configurações básicas como nome e autor do projeto:

```
{
  "name": "hello-vue",
  "description": "A Vue.js project",
```



```
"author": "Daniel Schmitz <danieljfa@gmail.com>",  
"private": true,
```

Após estas configurações iniciais, temos os **scripts**, descritos a seguir:

```
"scripts": {  
  "watchify": "watchify -vd -p browserify-hmr -e  
src/main.js -o dist/build.js",  
  "serve": "http-server -c 1 -a localhost",  
  "dev": "npm-run-all --parallel watchify serve",  
  "build": "cross-env NODE_ENV=production browserify  
src/main.js | uglifyjs -c warnings=false -m >  
dist/build.js"  
},
```

Vamos abordar cada um deles, veja:

- **"watchify" : "..."** Usa o pacote *watchify* em conjunto com *browserify-hmr* para recompilar o projeto sempre que houver uma modificação nos arquivos do projeto. Aqui também temos a configuração do arquivo inicial "-e src/main.js" que iniciará todo o processo de bundler (juntar) e como resultado teremos o arquivo final "-o dist/build.js"
- **"serve" : "..."** Usado para instanciar um pequeno servidor web capaz de receber requisições http e respondê-las. Este servidor funciona muito bem para pequenos projetos e testes, onde não há necessidade de processamento pesado no servidor. Pode-se usar este servidor para aprender tudo sobre Vue, e depois usar um servidor mais robusto com o express para processamentos mais complexos.

- **"dev" : "..."** Neste comando usa-se o pacote *npm-run-all* para executar tanto o *watchify* quanto o *server*, em paralelo. Assim o vue estará compilando o seu código fonte sempre que houver atualizações e o servidor web está recarregando a página sempre que um novo `dist.js` for gerado.
- **"build" : "..."** Usado para criar uma versão mais enxuta do arquivo `dist/build.js`, onde ela será minificada e "embaralhada". Nomes de variáveis serão reduzidas e o código ficará ilegível aos olhos humanos. Perceba também que uma variável de ambiente é alterada, o `NODE_ENV`, sendo informado "production", onde pode ser útil em determinadas situações em ambientes de produção.

Estes quatro scripts podem ser executados pelo `npm` através do comando `npm run <nome-do-script>`. Isso significa que, para que você possa testar o vue em funcionamento, basta executar `npm run dev`. Mas não faça isso agora, ainda existe um passo importante a ser executado no projeto.

Continuando com o `package.json`, temos agora a inclusão dos pacotes de dependência do projeto, veja:

```
"dependencies": {
  "vue": "^1.0.0"
},
"devDependencies": {
  "babel-core": "^6.0.0",
  "babel-plugin-transform-runtime": "^6.0.0",
  "babel-preset-es2015": "^6.0.0",
  "babel-preset-stage-2": "^6.0.0",
  "babel-runtime": "^6.0.0",
```

```
"cross-env": "^1.0.6",
"babelify": "^7.2.0",
"browserify": "^12.0.1",
"browserify-hmr": "^0.3.1",
"http-server": "^0.9.0",
"npm-run-all": "^1.6.0",
"uglify-js": "^2.5.0",
"vueify": "^8.5.2",
"watchify": "^3.4.0"
},
```

## Observação

*Na iminência do Vue 2, alguns projetos (como o browserify-simple) já estão com o Vue 2 configurados, então o package.json já está diferente do apresentado. Se você deseja instalar o browserify-simple na versão Vue 1, use:*

```
vue init browserify-simple#1 hello-vue
```

Na categoria "dependencies" temos o "vue" como framework do projeto, e na categoria "devDependencies" temos várias bibliotecas que auxiliam todo o processo de compilação e publicação do projeto. Por exemplo, babel é usado para converter código javascript para algo que o navegador irá compreender (Isso se chama *transpiler*, particularmente eu gosto de resumir para a palavra *compilar*). Veja que temos o *browserify*, o *http-server*, *vueify*, entre outros.

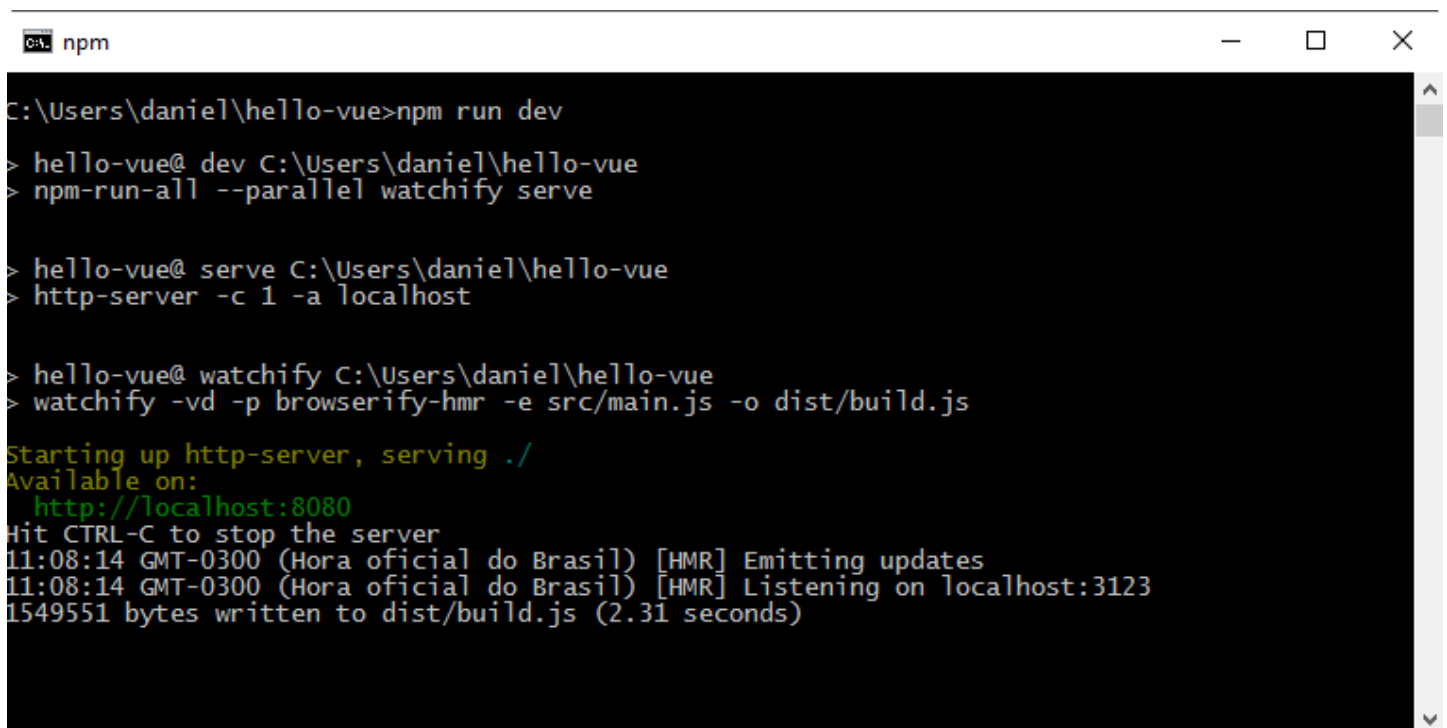
Para que possamos instalar todas as bibliotecas no projeto (pois elas ainda não foram instaladas), devemos executar o seguinte comando:

```
$ npm install
```

Este comando irá realizar o download de todas as bibliotecas e organizá-las na pasta `node_modules`. Após a instalação, veja que o diretório `node_modules` possui diversos pacotes. A partir deste momento, podemos testar a aplicação Vue em ação, bastando usar o seguinte comando:

```
$ npm run dev
```

Este comando irá exibir uma resposta semelhante a figura a seguir:

A terminal window titled 'C:\ npm' with standard window controls. The terminal shows the execution of 'npm run dev' in the directory 'C:\Users\daniel\hello-vue'. It lists the scripts for 'dev', 'npm-run-all', 'serve', and 'watchify'. The 'serve' script starts 'http-server' on port 8080. The 'watchify' script runs 'watchify' with various options. The output shows the server starting on http://localhost:8080 and HMR (Hot Module Replacement) emitting updates and listening on localhost:3123. The file 'dist/build.js' is updated with 1549551 bytes in 2.31 seconds.

```
C:\Users\daniel\hello-vue>npm run dev
> hello-vue@ dev C:\Users\daniel\hello-vue
> npm-run-all --parallel watchify serve

> hello-vue@ serve C:\Users\daniel\hello-vue
> http-server -c 1 -a localhost

> hello-vue@ watchify C:\Users\daniel\hello-vue
> watchify -vd -p browserify-hmr -e src/main.js -o dist/build.js


Starting up http-server, serving ./
Available on:
  http://localhost:8080
Hit CTRL-C to stop the server
11:08:14 GMT-0300 (Hora oficial do Brasil) [HMR] Emitting updates
11:08:14 GMT-0300 (Hora oficial do Brasil) [HMR] Listening on localhost:3123
1549551 bytes written to dist/build.js (2.31 seconds)
```

Perceba que o servidor está disponível no endereço `http://localhost:8080`. Ao acessar esta url no navegador, você verá a seguinte resposta:



O Vue neste momento está funcionando corretamente. É possível, a partir deste momento, alterar o arquivo `src/App.vue` e acompanhar as mudanças no navegador. Se você alterar algo no `<template></template>` a resposta será exibida imediatamente no navegador. Se alterar algo no `<script></script>`, necessita realizar um recar

---

**Daniel Schmitz**

Daniel Schmitz é autor de vários livros sobre programação e desenvolvimento de sistemas.

☐ Juiz de Fora - MG

☐ <http://www.danielschmitz.com.br>

**Share this post**

☐ ☐ ☐



LOG IN WITH

OR SIGN UP WITH DISQUS **Eduardo Borges** • a month ago

Obrigado por compartilhar....

^ | ▾ • Reply • Share ›

**edu Villas** • 2 months ago

```
$ vue init browserify-simple hello-vue
module.js:491
```

```
throw err;
```

```
^
```

```
Error: Cannot find module 'ini'
```

```
->Alguém faz idéia do que eu fiz errado??
```

^ | ▾ • Reply • Share ›

**Paul Sidewalk** • 4 months ago

Obrigado, Daniel. Além de muito bem explicado, fez o que pouquíssimos fazem : explicar o conteúdo e o objetivo de cada arquivo criado pelo vue-cli, importantíssimo.

^ | ▾ • Reply • Share ›

**Fábio Henrique Benatti** • 5 months ago

Estou com uma dificuldade em fazer uso da propriedade 'data' do meu objeto Vue que renderiza a página App.vue:

```
main.js
```

```
-----
```

```
import Vue from 'vue'
import App from './App.vue'
new Vue({
  el: '#app',
  data: {
    msg2: 'Vue js'
  },
  render: h => h(App)
})
```

```
App.vue
```

```
-----
```

```
<div {{ msg2 }} />
```

<!!!>{{ msgZ }}</!!!>

É possível fazer esse bind direto do nosso arquivo .js sem ter que usar o recurso 'export default' da seção script da nossa página .vue?

^ | v • Reply • Share ›



**Fabio Montefusco** • 6 months ago

Ufa, consegui começar! Muito obrigado pela introdução!!

^ | v • Reply • Share ›



**Thauan Santos** • 6 months ago

Ficou muito bom mano! Obrigado pela dedicação em criar algo de qualidade.

^ | v • Reply • Share ›



**Raphael Lacerda** • 6 months ago

ajudou muito

^ | v • Reply • Share ›



**Rodrigo Chinaglia** • 6 months ago

melhor impossivel :)

^ | v • Reply • Share ›



**Victor Ribeiro** • 6 months ago

Excelente post!

^ | v • Reply • Share ›



**Rodrigo Adachi** • 7 months ago

Show, muito bom

^ | v • Reply • Share ›



**Herlon Aguiar** • 8 months ago

Muito bom, valeu!

^ | v • Reply • Share ›



**Arlei F. Farnetani Jr** • a year ago

Top!

^ | v • Reply • Share ›



**Renan Smit** • a year ago

Muito bem explicado.

^ | v • Reply • Share ›



**Daniel Schmitz** • a year ago

Quem estiver com esse erro: Error: Cannot find module 'vue-hot-reload-api' ... vai no seu package.json e troca "vueify": "^8.5.2", por "vueify": "8.5.2", A versão 8.5.3 está com este bug. O

^ | v • Reply • Share ›



**Henrique Lobo (Kico)** • a year ago



**Henrique Lobo** (100) • a year ago

Oi pessoal, muito bacana esta iniciativa: cadastrem o VueJS Brasil no /dev/All pro pessoal poder acompanhar as postagens!  
<http://www.devall.com.br> (a propósito, usei o Vue.JS no site, aqui minha experiência: <http://www.itexto.net/devki...>

^ | v • Reply • Share ›



**Alan Ktquez** • a year ago

Show Daniel, explicadíssimo

^ | v • Reply • Share ›



**vedovelli** Mod • a year ago

Obrigado Daniel! Bem abrangente e bem escrito!

^ | v • Reply • Share ›

#### ALSO ON VUE.JS BRASIL

##### **Vue 2 no mundo mobile com Quasar - Comunidade Brasileira de**

10 comments • 8 months ago

**Leonardo Vilarinho** — Obrigado pelo feedback, pessoalmente ainda prefiro o Ionic por ser mais solido no

##### **Podcast - Falando sobre Vue.JS e Web Components by CODECASTS**

1 comment • a year ago

**Bruno Kilian** — Fala Vinicius, link do Slack aqui no site ta offline. Tão

##### **Filtros no Vue.js 2.0 - Comunidade Brasileira de Vue.js**

3 comments • a year ago

**Vinicius Reis** — Já não eta fã, desde que anunciaram o fim deles eu passei a não usar e acho bem melhor assim.

##### **ACL no Vue 2 - Comunidade Brasileira de Vue.js**

14 comments • 9 months ago

**Leonardo Vilarinho** — Ajuda bastante hehe obrigado pelo feedback

READ THIS NEXT

## Vue-router

A característica mais marcante de uma Single Page Application é a velocidade. Os elementos de interface já estão presentes...

YOU MIGHT ENJOY

## Gravação do 1º Hangout Vue.js Brasil

Hangout realizado em 25 de Junho de 2016....



