



Vuex como apoio arquitetural

Meetup Javascript Joinville

Jackson Ricardo Schroeder



Sobre

Desenvolvedor front-end na FESC
Tecnologia. Apaixonado pelo javascript. Na
área de TI há mais de 20 anos, sendo os
últimos 4 focados em front-end e seu
ecossistema.

Meus principais projetos não estão no github:





Uma segunda profissão



Mas programar é demais!

- Democrático
- Acessível
- Prazeroso
- Mercado aquecido
- **Não tem idade**



Alguns jovens senhores



Donald Knuth
The Art of Computer Programming
81 anos



Anders Hejlsberg
Delphi / .NET / Typescript
58 anos



Uncle Bob
Clean Code / Clean Architecture / Agile
67 anos



Martin Fowler
Refactoring
56 anos

E eles programam mesmo...

Accept new baselines



ahejlsberg committed yesterday ✓



00505ad



Merge pull request [#34212](#) from microsoft/fix34021 ...



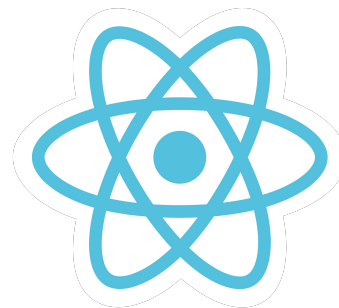
ahejlsberg committed yesterday ✗

Anders dedilhando loucamente...

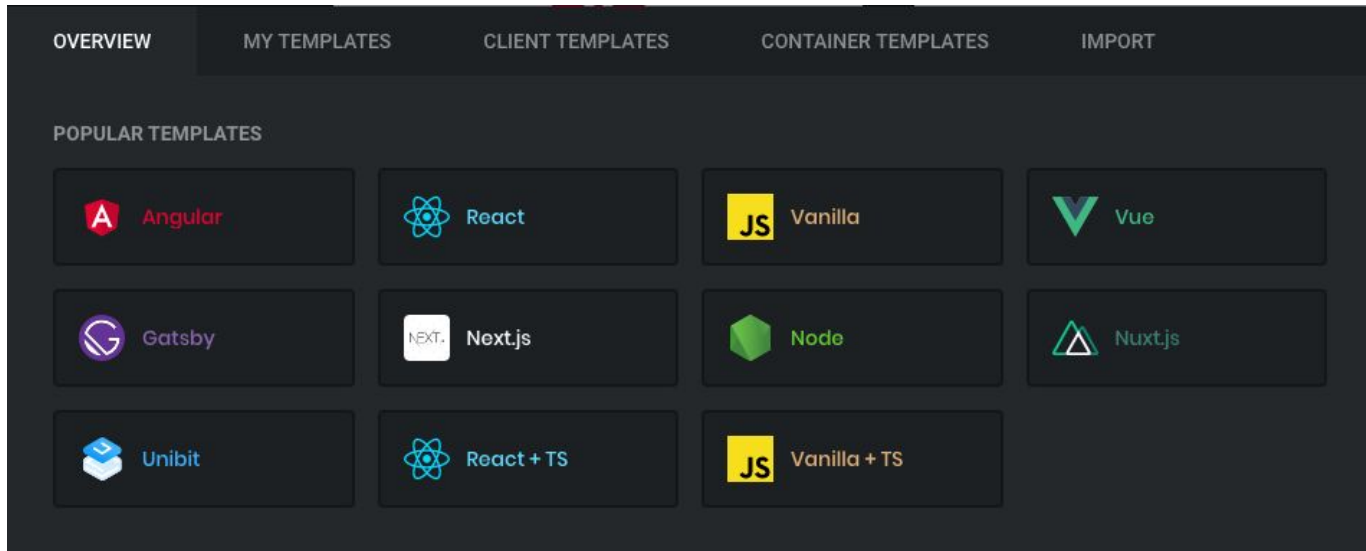




Estamos bem servidos



Codesandbox.io



Dan Abramov

2015: *Smart and dumb components*

2017: *Containers and Presentation*

2019: *Forget it all!*

https://medium.com/@dan_abramov/smart-and-dumb-components-7ca2f9a7c7d0





Tipos de componentes

- **Presentation**

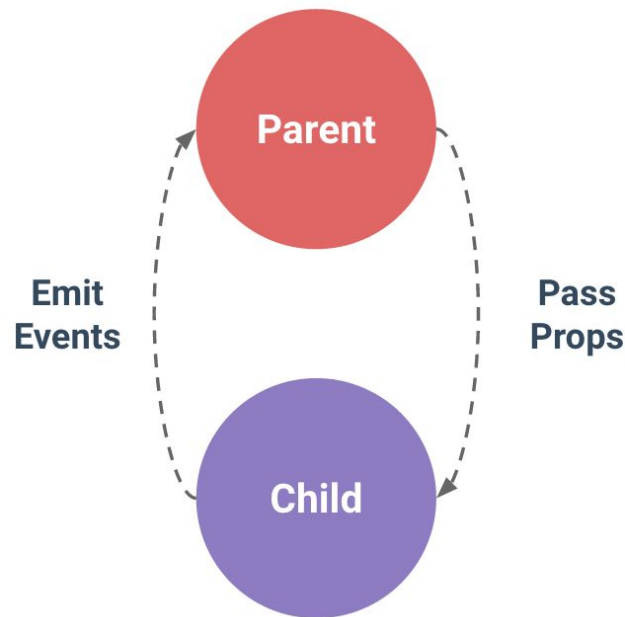
- Dumbs (cruel né!?)
- Agnósticos de estado
- De apresentação
- De design system
- Framework de componentes

- **Containers**

- Composição de componentes
- View
- Páginas
- Rotas
- Telas

Componentes de apresentação

- Comunicação **unidirecional** entre componente pai e filho;
- O pai passa informação para o filho através de props;
- Sem efeito colateral -> reagem apenas às props;
- O filho emite eventos para se comunicar com o pai;
- Componente como funções puras: Renderiza o mesmo conteúdo caso receba os mesmos valores nas props.





Containers / Views / Pages

- Composição de componentes a fim de consolidar um fluxo de dados para o usuário final.
- Uma tela do sistema.
- Em um CRUD:
 - a listagem;
 - a página com o formulário para adição e edição;
 - o próprio formulário;
- Um dashboard;
- Uma rota em um SPA vai renderizá-lo.
- Controla o estado que vai para os componentes.
- Integra com os serviços.
- Faz a ponte entre as libs de gerenciamento de estado (Vuex) e os componentes.

Vuex - Quando e por que usar?

“Flux libraries are like glasses: you’ll know when you need them.”

“Bibliotecas Flux são como óculos: Você saberá quando vai precisar deles.”

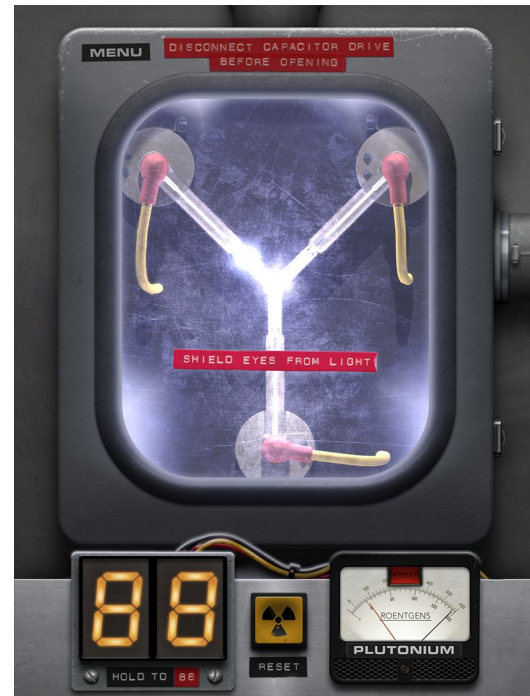


(Dan Abramov)

Facebook, React Team, Co-author of Redux and Create React App

Vuex - Baseado na arquitetura Flux

- Facebook Flux, Redux, Vuex, **MartyJS**, **McFly**, **Delorean** são exemplos de libs baseadas na arquitetura Flux.





O que é Vuex?

- Baseado na arquitetura Flux (nome vem do capacitor de fluxo do filme? 🤪);
- É equivalente ao Redux do React;
- É uma lib + um padrão de gerenciamento de estado da aplicação;



O que é Vuex?

- O estado é ~~mutável~~ **evoluído** de forma previsível e controlada;



O que é Vuex?

- Ele serve como uma base de dados de estados, centralizada;



O que é Vuex?

- Integra com a extensão devtools do Vue.js;
- Oferece fácil navegação na linha de tempo do estado da aplicação (voltar no tempo - **time travel**);

Curiosidade: O que está pegando com o Vue Dev Tools em sua nova versão? (versão 5)



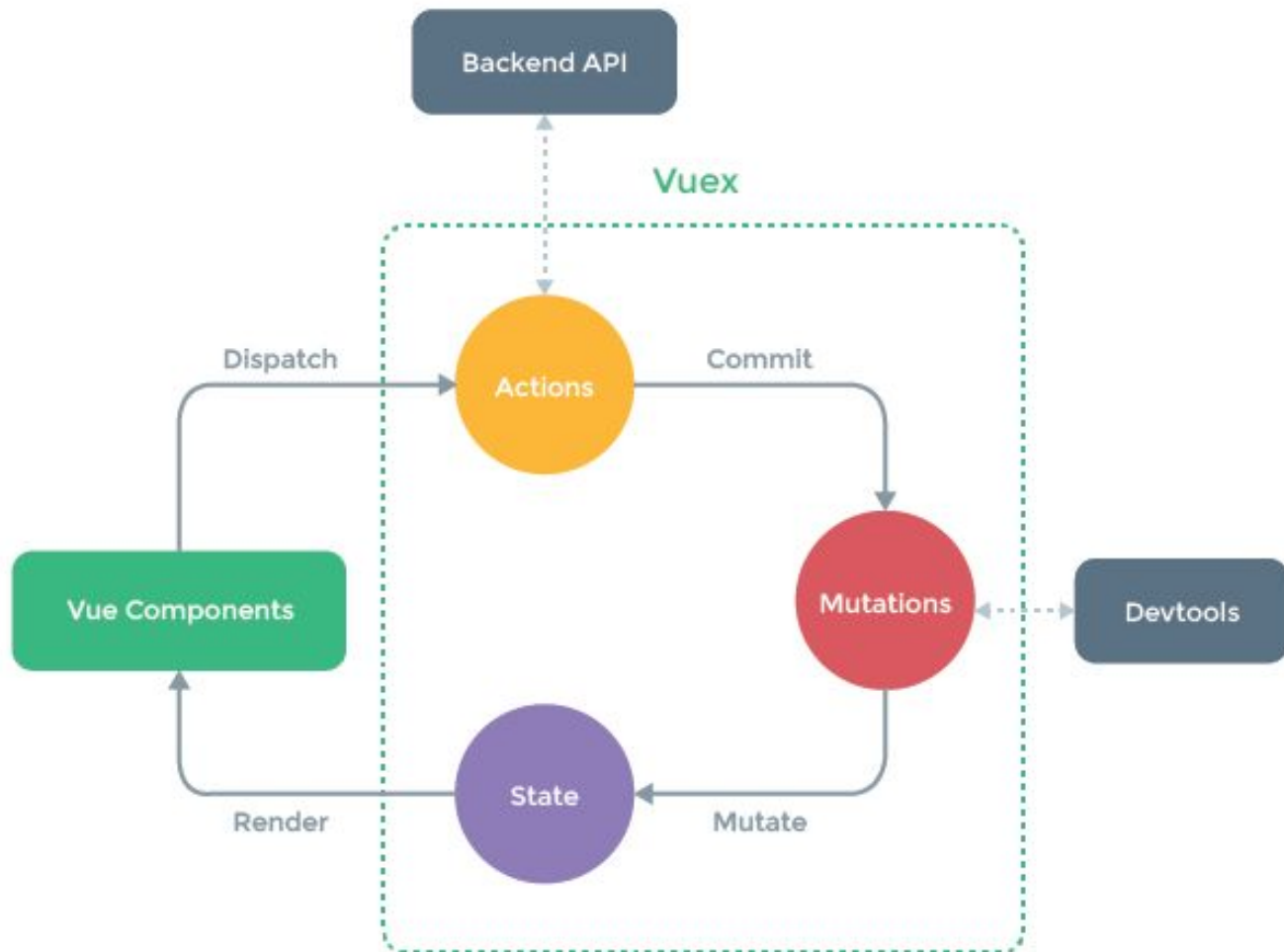
O que é Vuex?

- **Apoia** o isolamento e separação de responsabilidades;
- **Apoia** o isolamento da camada de serviços;
- **Apoia** o encapsulamento de regras de negócio;

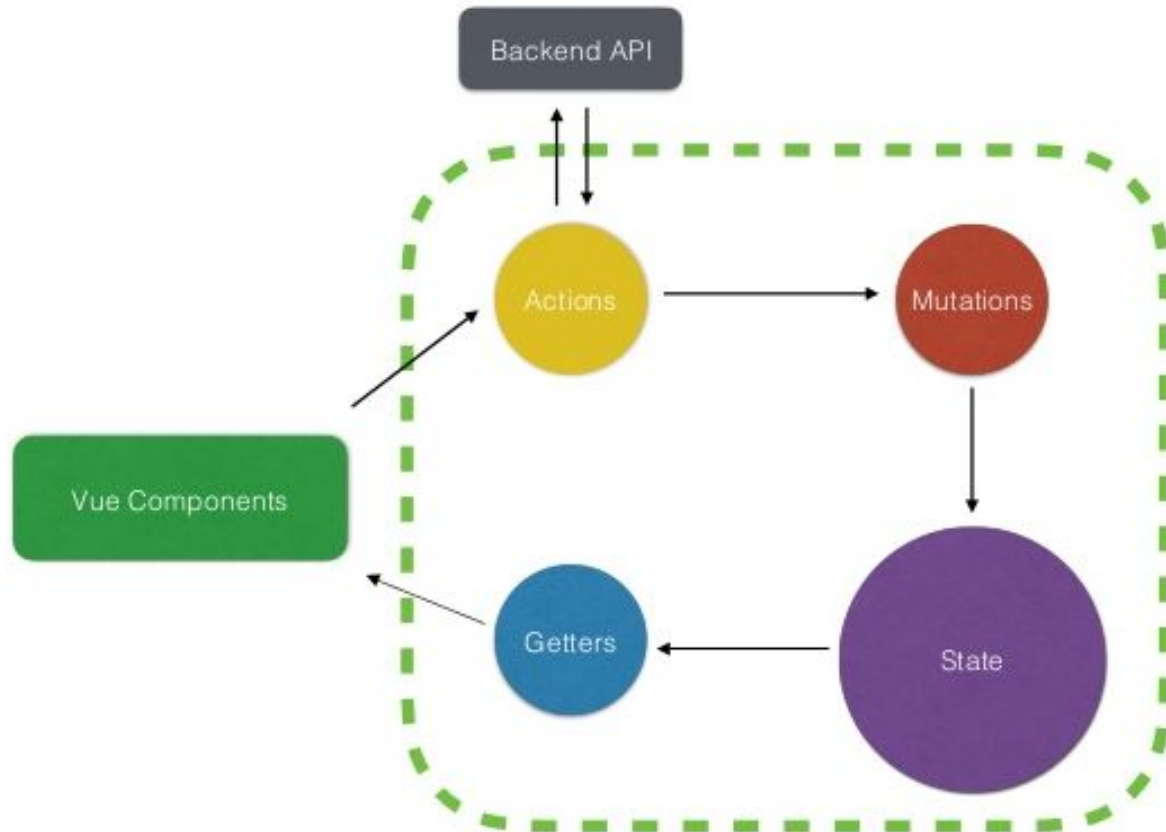


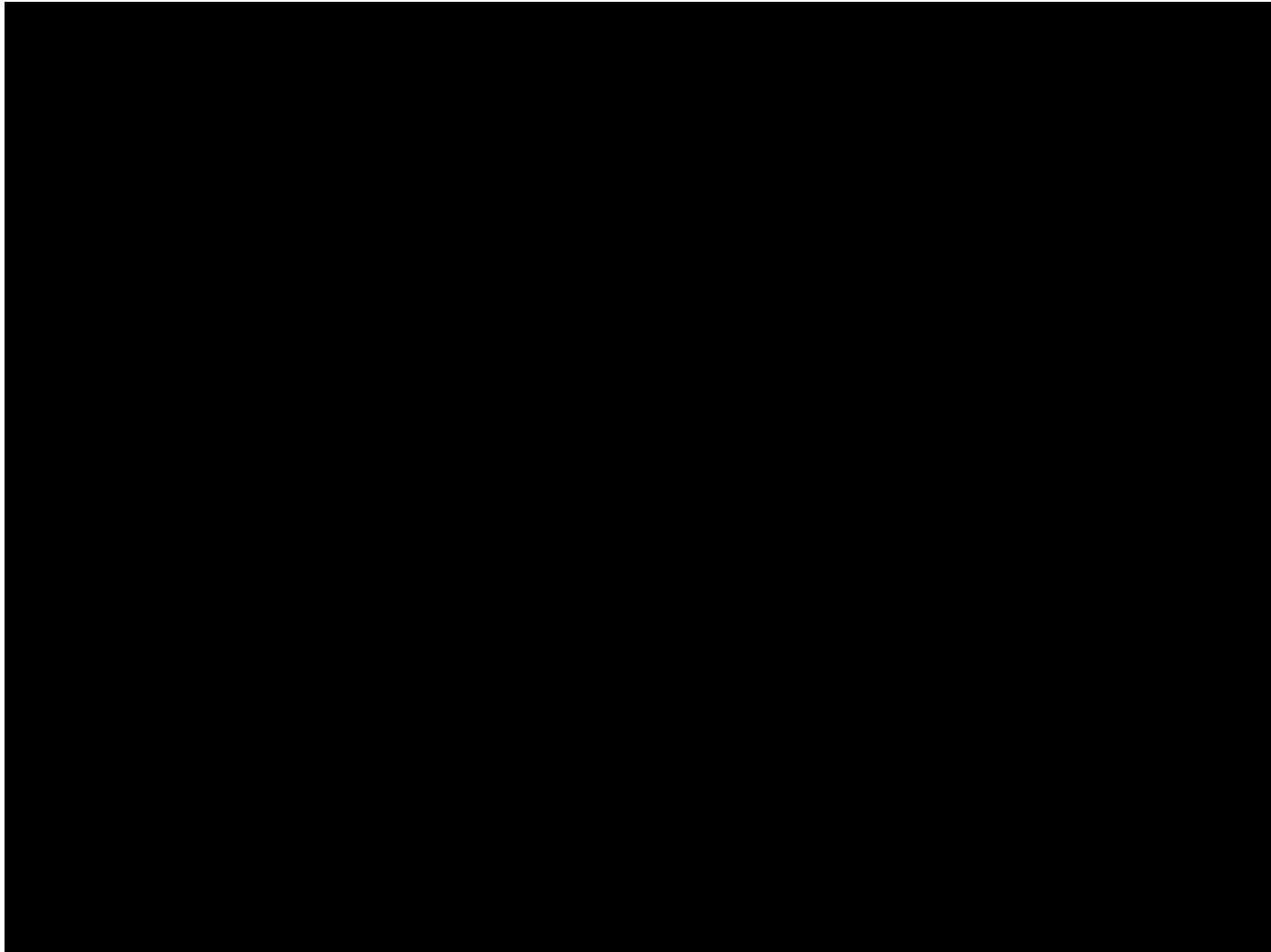
Diagrama e fluxo Vuex

- Promises são resolvidas nas actions do Vuex;
- Com as promises resolvidas, as actions efetivam os dados na store, através das MUTATIONS;
- Transformações de dados são realizadas nos getters do Vuex (ajuste de dados, aplicação de máscara, conversão de data, formatação de valores, novas propriedades, etc...);



Vuex





← → ↺ ↻

localhost:8080/aut/solicitacoes/gerencial

Apps Desenvolvimento Livros Desenvolvimento Material de divul FESC Cursos Emails Programação Fullstack Complete list of

SGSUITE

Módulos

Este módulo

Buscar (CTRL + Espaço)

Administrador

Solicitações em Estudo - Visão Gerencial

Entrada

Em Análise

Em auditoria técnica

Concluído

Número da guia, beneficiário...

Pesquisar

Prazo	Caráter	Nº Guia	Tipo	Unimed	Carteira	Beneficiário	Dt. Solicitação	Dt. Entrada
E		500414...	Interna...	976	0976.7001.00...	JEAN OLIVEIRA DOURADO	20/09/...	03/10/...

<< < 1 > >>

Listando 1 - 1 de 1 registros (p. 1 de 1)

Vue

6

Base State 11:09:27

gerencial

getters

solicitacoes/gerencial/activeTab: Object

solicitacoes/gerencial/activeTabIndex: 3

solicitacoes/gerencial/getTab: f (tabIndex)

solicitacoes/gerencial/isloading: false

solicitacoes/gerencial/pagination: Object

currentPage: 1

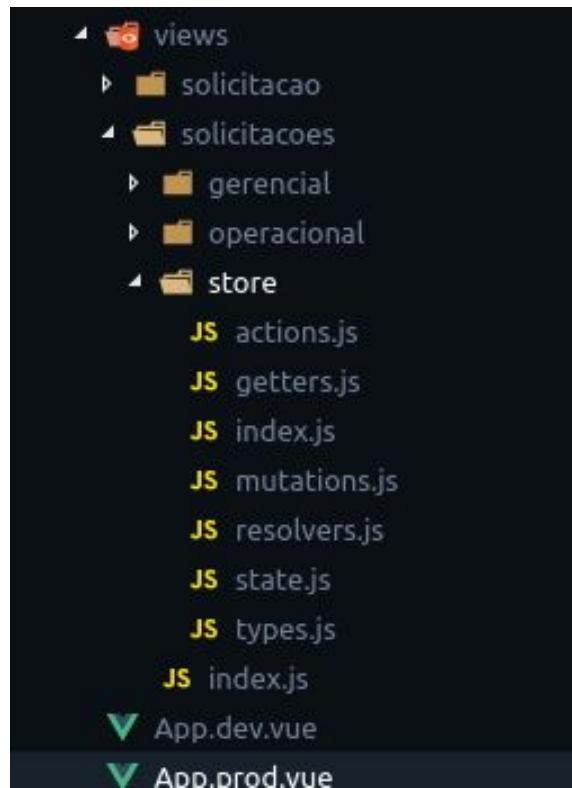
itemsPerPage: 12

totalElements: 1

solicitacoes/gerencial/search: ""



A Store Vuex



Vuex Actions

```
1
2 const someAction = context => {
3   ...
4 }
```

*** recebem 2 argumentos (context, payload)*

```
1 context: {
2   commit,
3   dispatch,
4   getters,
5   state,
6   rootState,
7   rootGetters,
8 };
```



```
1 const loadOne = async ({ commit, dispatch }, id) => {  
2   dispatch('setLoading', true);  
3  
4   const { payload, isSuccess } = await getOne(id);  
5  
6   commit(types.SET_FORM_DATA, payload?.usuarioAutorizador ?? {});  
7  
8   dispatch('setIntegrationError', !isSuccess);  
9   dispatch('setLoading', false);  
10 };  
11
```

Número da guia, carteira, beneficiário, data

Pesquisar

Adicionar Filtro

Exibindo: 15 registros



Nº Guia	Tipo	Caráter	Ud. Solic.	Ud. Benef.	Carteira	Beneficiário	Dt. Solicitação	Situação	Senha
20191011000178	SP/SADT	E	0976	0000	0976	FABIO	23/09/2019	ESTUDO	-
20191011000177	SP/SADT	E	0976	0000	0976	FABIO	23/09/2019	ESTUDO	-
20191011000176	SP/SADT	E	0976	0000	0976	FABIO	23/09/2019	APROVADA	2019101...
20191011000175	SP/SADT	E	0976	0000	0976	FABIO	23/09/2019	APROVADA	2019101...
20191011000174	SP/SADT	E	0976	0000	0976	FABIO	23/09/2019	APROVADA	2019101...
20191011000173	SP/SADT	E	0976	0000	0976	FABIO	23/09/2019	APROVADA	2019101...
20191011000171	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	APROVADA	2019101...
20190923000136	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	APROVADA	2019092...
20190923000135	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	APROVADA	2019092...
20190923000134	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	ESTUDO	-
20190923000133	Internação	E	0634	0976	0976	FABIO	23/09/2019	ESTUDO	-
20190923000132	Internação	E	0634	0976	0976	FABIO	23/09/2019	ESTUDO	-
20190923000130	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	APROVADA	2019092...
20190923000129	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	APROVADA	2019092...
20190923000128	SP/SADT	E	0976	0976	0976	FABIO	23/09/2019	APROVADA	2019092...

Listando 1 - 15 de 1862 registros
(p. 1 de 125)

« < 1 2 3 4 5 6 7 > »

actions



Vuex Getters



```
1 const someGetter = (state, getters, rootState, rootGetters) => { ... }
```

*** recebem até 4 argumentos*



```
1 // chaining operator
2 const entityId = ({ uiControls }) => uiControls?.formData?.id ?? null;
3
4 // lodash get
5 const entityId = ({ uiControls }) => get(uiControls, 'formData.id', null);
6
```

```
1 const hasAdvancedSearch = ({ uiControls }) => {
2   const { advancedSearch } = uiControls;
3   const {
4     numeroCartao,
5     nomeBeneficiario,
6     unimedBeneficiario,
7     unimedSolicitante,
8   } = advancedSearch;
9
10  return !(
11    numeroCartao ||
12    nomeBeneficiario ||
13    unimedBeneficiario ||
14    unimedSolicitante
15  );
16 };
```



```
1 const someGetter = state => ({  
2   ...state,  
3   startDate: dateFormatter(state.startDate),  
4   endDate: dateFormatter(state.endDate),  
5   total: state.items.reduce((acc, cur) => acc + cur.quantity, 0),  
6 });
```



```
1 const ordersQuantity = (state, getters) => getters.someGetter?.length ?? 0;  
2  
3 const ordersQuantity = (state, { someGetter }) => someGetter?.length ?? 0;
```

Número da guia, carteira, beneficiário, data...

Pesquisar

Adicionar Filtro

Exibindo: 15 registros



Nº Guia	Tipo	Caráter	Ud. Solic.	Ud. Benef.	Carteira	Beneficiário	Dt. Solicitação	Situação	Senha
20191011000178	SP/SADT	E	0976	0000	09	FABIO	23/09/2019	ESTUDO	-
20191011000177	SP/SADT	E	0976	0000	09	FABIO	23/09/2019	ESTUDO	-
20191011000176	SP/SADT	E	0976	0000	09	FABIO	23/09/2019	APROVADA	2019101...
20191011000175	SP/SADT	E	0976	0000	09	FABIO	23/09/2019	APROVADA	2019101...
20191011000174	SP/SADT	E	0976	0000	09	FABIO	23/09/2019	APROVADA	2019101...
20191011000173	SP/SADT	E	0976	0000	09	FABIO	23/09/2019	APROVADA	2019101...
20191011000171	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	APROVADA	2019101...
20190923000136	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	APROVADA	2019092...
20190923000135	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	APROVADA	2019092...
20190923000134	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	ESTUDO	-
20190923000133	Interação	E	0634	0976	09	FABIO	23/09/2019	ESTUDO	-
20190923000132	Interação	E	0634	0976	09	FABIO	23/09/2019	ESTUDO	-
20190923000130	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	APROVADA	2019092...
20190923000129	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	APROVADA	2019092...
20190923000128	SP/SADT	E	0976	0976	09	FABIO	23/09/2019	APROVADA	2019092...

Listando 1 - 15 de 1862 registros

(p. 1 de 125)

getters



Vuex Mutations

```
1 // mutation declarada com types
2 [types.SET_LOADING](state, payload) {
3   state.isLoading = payload;
4 },
```

```
1 // mutation declarada direto sem types
2 SET_LOADING(state, payload) {
3   state.isLoading = payload;
4 },
```

```
1 // mutation genérica
2 STATE_UPDATER(state, { prop, payload }) {
3   state[prop] = payload;
4 },
```



E o componente, como fica?

```
1 import { mapActions, mapState, mapGetters } from 'vuex';
2
3 export default {
4   name: 'Container',
5   computed: {
6     ...mapState('stateA', 'stateB'),
7     ...mapGetters('someGetter'),
8   },
9   methods: {
10     ...mapActions('someAction'),
11   }
12 }
```




Revisão Vuex

- **State:** O estado corrente da store;
- **Action:** Contém lógica de negócio, acessa fonte dados e dispara as mutations;
- **Getters:** Recalcular dados da store.

** **Getters** são para a store Vuex, o mesmo que as propriedades computadas (computed) são para uma instância de componente Vue;



Revisão Vuex

- **Mutation**: Sua única responsabilidade é de evoluir o estado da aplicação.
- Helpers **mapState** e **mapGetters**: Converter, automaticamente, para uma propriedade computada do componente, uma propriedade do Vuex state ou um Vuex getter;
- Helper **mapActions**: Instancia uma action Vuex como um método local do componente;
- *Resolvers: Analisam e transformam os dados antes de entregar para os getters;*



Principais benefícios no uso do Vuex

- Mutabilidade controlada;
- Pequenas funções com responsabilidade única;
- Funções puras;
- Reuso dentro de um getter de outros getters pré-existentes;
- Dispatch dentro de uma action de outras actions pré-existentes;
- Separação de responsabilidades;
- Componentes mais enxutos;
- Melhor depuração.



Principais desafios no uso do Vuex

- Curva de aprendizado;
- Mais burocrático;
- Saber onde e quando usar;

Vue Composition API

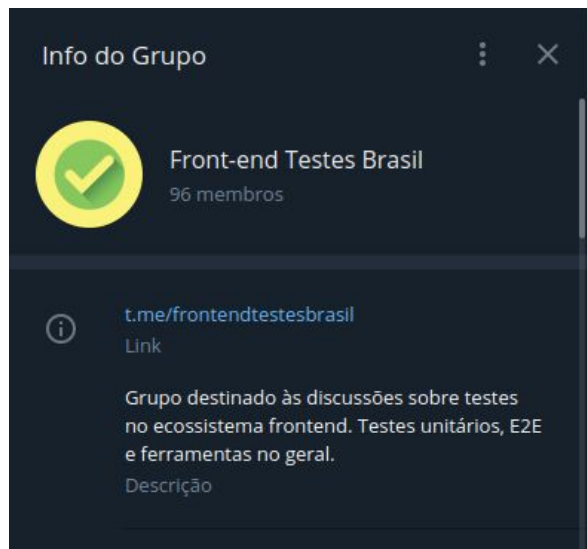
- A RFC era chamada de Functional API RFC
- Já tem suporte para o Vue.js 2
- Reuso de lógicas
- Organização de código
- Funcional
- Explícito
- Melhor suporte para o Typescript
- Mais fácil de testar
- Menos **this** -> menos mágicas -> menos problemas
- Fizeram um hotsite bem completo sobre a Composition API:
<https://vue-composition-api-rfc.netlify.com>



Evan You

creator of the open source JavaScript
framework Vue.js

Telegram





Obrigado!



<https://twitter.com/xereda>