API Options

São propriedades definidas dentro do objeto de configuração de um componente, para construir e controlar seu comportamento:

```
export default {
  name: “”,
  data() {},
  methods: {}
}
```

name: “currentComponentName” – Define o nome do componente.

components: {componente} – Registra componentes filhos que podem ser usados dentro do template do componente atual.

Directives: {

focus: {mounted(el) {el.focus()}},

} - Permite registrar diretivas personalizadas locais que podem ser usadas no template do componente. São usados para definir comportamentos personalizados que podem ser aplicados diretamente a elementos DOM com v-.

inheritAttrs: false – É uma opção booleana (sendo true o padrão) que controla se atributos HTML não explicitamente definidos como props serão automaticamente aplicados ao elemento raiz do componente.

props: {name: String, age} - São os valores que o componente filho recebe do componente pai. Permite passar dados de um componente pai para um filho de forma reativa e controlada.

emits: {} ou emits: [ ] - Declara os eventos personalizados que o componente pode emitir para o componente pai usando $emit. Basicamente documenta quais eventos o componente emite.

data() {

return {variavel: valor,}

} - Define as variáveis que ele usa e que podem mudar com o tempo, ou seja, define todas as variáveis do componente atual.

computed: {func() {}} - Permite criar valores automáticos com base em outros dados do componente, ou seja, faz funções que atualizam quando os valores são atualizados. Uteis para lógica de exibição, formatação, filtros etc. Só recalculam quando suas dependências mudam.

methods: {func() {},} - É onde as funções do componente são definidas.

watch: {func() {}} - Permite observar mudanças em dados reativos (data, props ou computed) e executar uma função quando esses dados mudam.

provide() {return {variavel: valor}} – Serve para fornecer dados a componentes descendentes, sem precisar passar por props em cada nível da hierarquia. Basicamente serve para compartilhar valores com componentes filhos, netos, bisnetos, etc.

inject: [‘variavel’] – É usada para receber dados que foram fornecidos por um componente ancestral (pai) com provide, ou seja, acessa valores vindos de provide, sem precisar receber por props.

template: ` ` - Permite definir conteúdo HTML diretamente como uma string em vez de usar o arquivo .vue com a seção <template>.Define o conteúdo visual do componente.

beforeCreate() {} - Executa antes do vue iniciar a reatividade, injeções (provide/inject) e eventos.

created() {} - Executa quando o vue é criado.

beforeMount() {} - Executa antes do componente ser montado no DOM, ou seja, antes de ser exibido na tela.

mounted() {} - Executa quando o componente é montado no DOM.

beforeUpdate() {} - Executa antes do DOM ser atualizado.

updated() {} - Executa depois que o DOM é atualizado.

activated() {} - Executa quando um componente é ativado novamente, volta a ser exibido, sem ser recriado, mas somente quando esta sendo usado com <keep-alive>.

deactivated() {} - Executa quando um componente é “desativado”, ou seja, removido da tela.

beforeUnmount() {} - Executa antes do componente ser destruido e removido do DOM. Serve para executar limpeza como remover listeners, cancelar timers, etc.

unmounted() {} - Executa depois que o componente foi completamente removido do DOM.