Vue Fundamentals Options API / Composition API Peter Kassenaar – info@kassenaar.com

Vue historically used the 'Options' API.

Since Vue 3, the 'Composition API' is the default.

What the heck is that??

(and why should I bother?)

Traditionally (Vue 1, Vue 2)

- "Option based API", b/c components are written as objects with key/value pairs, the options, defined in Vue documentation
- Code is organized by option and not by functionality
- Data, methods, computed properties are spread all over the component

This is not wrong per se, but has some limitations

Vue 3 – Composition API

Currently: the recommended way of creating Vue Components

```
<script setup>
import {ref} from 'vue';

// reactive state
const count = ref(0)

// function to mutate state and trigger updates
const increment = () => {
  count.value++;
}

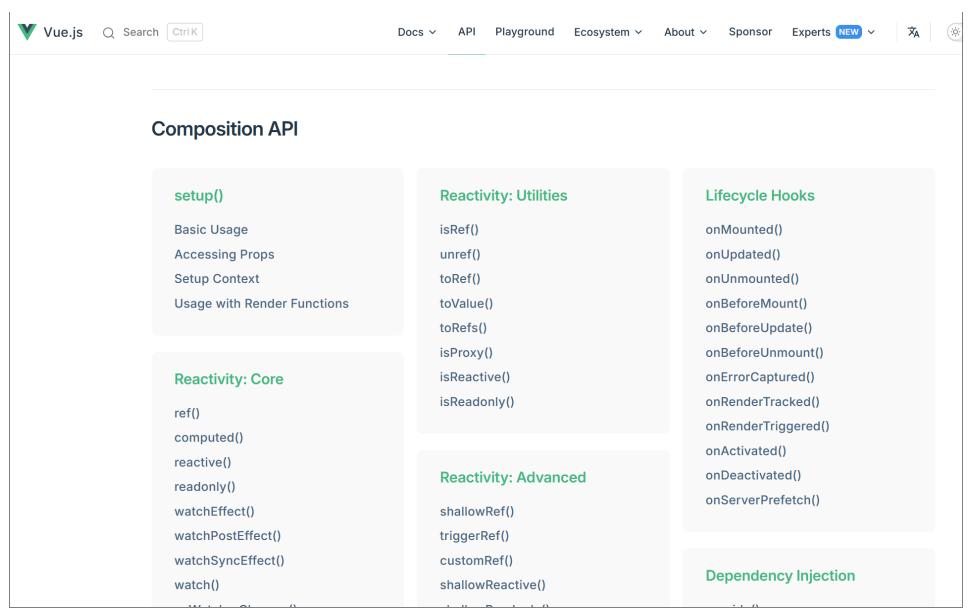
</script>
```

Why Composition API?

- Better reuse of logic
 - Create composable functions and reuse it in applications and projects
- More flexible code organization
 - State, functionality, data, everything has its own unique place instead of scattered all over the component
- Better Type inference
 - Better alignment with current TypeScript practices as a lot of companies do
- Smaller bundle size, less overhead
 - Better performance on the client

Vue 3 uses Composition API by default

```
how component usages
       <script setup>
      defineProns( props: {
        msg: {
 3
 4
          type: String,
 5
          required: true,
 6
 7
        name: {
 8
          type: String,
 9
          required: false
10
11
      })
      </script>
12
13
14
      <template>
        <div class="greetings">
15
          <h1 class="green">{{ msg }}</h1>
16
          <h3>
            You've successfully created a project with
18
            <a href="https://vite.dev/" target="_blank" rel="noopener">Vite</a> +
19
            <a href="https://vuejs.org/" target="_blank" rel="noopener">Vue 3</a>.
20
          </h3>
21
          <h2> --> My name is: {{ name }}</h2>
22
        </div>
23
      </template>
24
25
      <style scoped> PeterKassenaar, 4-12-2024 14:12 • feat: update example to Vite
26
27
      h1 {
        font-weight: 500;
28
```



https://vuejs.org/api /

Checkpoint

- You know the Composition API is the preferred way of creating new Vue applications.
- The Options API is not going away.
 - Existing Vue-applications can still be maintained and updated
- Use the <script setup> block in Composition API
- You know where to find more information on the Composition API.