



# □ Bienvenue sur Vitify

Une interface claire, modulaire et performante pour vos besoins.



Des outils puissants et flexibles

Fonctionnalités intelligentes

Fonctionnalités intelligentes



## Connexion à la plateforme

Simple, sécurisée, efficace

# Performance

Optimisé pour une expérience fluide



# Welcome to Vitify Admin

Presentation for Vitify Admin and new features in Online Monitor

YUE JIN



## ☐☐ Vitify Admin?

Vitify  Vite + Vuetify = Vitify

-



# pnpm

- 包管理工具

- 包管理工具

- patch package

□

Vue□□□□□□□□

pnpm prune

node\_modules □□□

# Vite

- 快速启动
- 快速开发
- 快速部署

config.json

```
VITE v3.0.0 ready in 8873 ms
→ Local:   http://127.0.0.1:9527/
→ Network: use --host to expose
12:10:51 [vite] .env.local changed, restarting server...
```



# vite-plugin-pages

- 自动扫描路由
- 支持路由懒加载
- 支持 vite-plugin-vue-layouts 布局

```
1 src/pages/  
2   └─ users/  
3     └─ [id].vue  
4       └─ index.vue  
5   └─ users.vue
```

```
1  [  
2    {  
3      path: "/users",  
4      component: "/src/pages/users.vue",  
5      children: [  
6        {  
7          path: "",  
8          component: "/src/pages/users/index.vue",  
9          name: "users"  
10       },  
11       {  
12        path: ":id",  
13        component: "/src/pages/users/[id].vue",  
14        name: "users-id"  
15      }  
16    ]  
17  }  
18 ]
```

# Auto Import

❌❌❌❌❌❌ import

- unplugin-vue-components ❌❌ vuetify-loader ❌❌❌❌ Vuetify ❌❌
- unplugin-auto-import import API

without

```
1 import { computed, ref } from 'vue'
2 const count = ref(0)
3 const doubled = computed(() => count.value * 2)
```

with

```
1 const count = ref(0)
2 const doubled = computed(() => count.value * 2)
```

## svg icons

`<svg>`  `<v-icon>` 

- ```
import.meta.glob('src/assets/**/*.svg', { eager: true })
vite-plugin-vue2-svg
vuetify
```

```

1  const svgIcons = Object.fromEntries(
2    Object.entries(import.meta.globEager('@assets/icons/*.svg')).map(
3      ([k, v]) => [
4        k
5          .split(/(\\|\\/)/g)
6          .pop()!
7          .replace(/\.([^./]+)$/ , ''),
8        { component: v.default },
9      ]
10   )
11 )
12 new Vuetify({icon:{values:{... svgIcons}}})

```

# Volar

- `typescript`
- `.vue` `tsconfig.json`
- `tsconfig.json`
- `@volar-plugins/vetur` `vetur` `tsconfig.json`
- `Vuetify`

```
npm run type:vuetify
```



**dschreij** commented 2 days ago



**@kingyue737** I copied and ran your script and it works great in generating the type files! Thanks for it.  
Did you have any luck getting the typings to work again after the problems reported in this thread, or also no luck since?

# TypeScript Project Reference

- `tsconfig.json` (project)
- VSCode `tsconfig.json`
- `tsconfig.json`

vue Tag: kebab-case TS 4.7.4 (takeover) Attr: kebab-case tsconfig.app.json ✓ Prettier

TypeScript TS 4.7.4 (takeover) tsconfig.vitest.json ✓ Prettier





## Before

```
1  <script lang="ts">
2  import { reactive, defineComponent } from 'vue'
3  import BtnCount from './BtnCount.vue'
4  export default defineComponent({
5    components: { BtnCount },
6    setup() {
7      const state = reactive({ count: 0 })
8      function increment() {
9        state.count++
10      }
11      return {
12        state,
13        increment
14      }
15    }
16  })
17 </script>
18 <template>
19   <BtnCount @click="increment">
20     {{ state.count }}
21   </BtnCount>
22 </template>
```

## With <setup script>

```
1  <script setup lang="ts">
2  import BtnCount from './BtnCount.vue'
3  const state = reactive({ count: 0 })
4  function increment() {
5    state.count++
6  }
7  </script>
8  <template>
9    <BtnCount @click="increment">
10      {{ state.count }}
11    </BtnCount>
12  </template>
```

## Before

```
1 import type { PropType } from 'vue'
2 export default defineComponent({
3   props: {
4     dataset: {
5       type: Array as PropType<number[][]>,
6       required: false,
7       default: () => [],
8     },
9     sym: {
10      type: Boolean,
11      default: false,
12    },
13    valueFormatter: {
14      type: Function as PropType<(val: number) => string>,
15      required: false,
16      default: (val: number) => val.toFixed(3),
17    },
18  }
19 })
```

## With <setup script>

```
1 const props = withDefaults(
2   defineProps<{
3     dataset?: number[][]
4     sym?: boolean
5     valueFormatter?: (val: number) => string
6   }>(),
7   {
8     dataset: () => [],
9     sym: false,
10    valueFormatter: (val: number) => val.toFixed(3),
11  }
12 )
```

ES6

Before

```
1 <template>
2   <span> {{ power && power.toFixed(1) }} </span>
3 </template>
```

vue-template-compiler

```
1 <template>
2   <span> {{ power?.toFixed(1) }} </span>
3 </template>
```

# Pinia

- 兼容 options api 和 composition api
- 轻量级 mutation
- 集成 vue devtools

# Portal-Vue

Vue2 与 Vue3 的 <Teleport>

## 使用场景

- header
- footer

z-index

dialog

notification



# I18n

vue-i18n v8    vue-i18n-bridge + unplugin-vue-i18n + i18n-Ally

- Messages
- SFC Custom

```
1 // zh.json
2 {
3   "viewLevel": {
4     "private": "??",
5     "group": "??",
6     "public": "??",
7     "disabledInfo": "?????????????????????"
8   }
9 }
```

```
1 <!-- ViewLevel.vue -->
2 <i18n lang="json" locale="zh">
3 {
4   "private": "??",
5   "group": "??",
6   "public": "??",
7   "disabledInfo": "?????????????????????"
8 }
9 </i18n>
```

# Vue-Echarts

Apache ECharts component for Vue.js

- Auto resize
- Provide/Inject

- `echarts.on` → Component Event listener

```
1 <v-chart autoresize />
```

```
1 const vuetify = useVuetify()  
2 const { locale } = useI18n()  
3 provide(  
4   THEME_KEY,  
5   computed(() => (vuetify?.theme.dark ? 'dark' : undef  
6 )  
7 provide(  
8   INIT_OPTIONS_KEY,  
9   computed(() => ({ locale: locale.value.toUpperCase()  
10 )
```

```
1 <v-chart  
2   @mousemove="onMouseMove"  
3   @mouseout="onMouseout"  
4   @selectchanged="onSelectChanged"  
5 />
```





# Vitest

- Vite 生态
- 测试 watch 模式
- Jest expect 断言



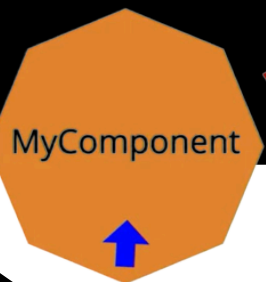
# How to test

tips

□□□

tips

Props



Events



Rendered Output



User Interaction



Lifecycle Methods



- □□□□□□□□
- □□□□□□

## Resources

- `@vue/test-utils` □□
- "Component Tests with Vue.js" by Evan You and Matt O'Connell
- `testing-library` □□
- End-to-End or Component Tests

# Testing Library

- DOM 测试库
- `@vue/test-utils` 测试 Vue 组件

```
1  it('login correctly', async () => {
2    const { getByText, getByLabelText } = renderWithVuetify(loginPage)
3    getByText('登录')
4    const userInput = getByLabelText('用户名')
5    await fireEvent.update(userInput, 'admin')
6
7    const passwordInput = getByLabelText('密码')
8    await fireEvent.update(passwordInput, 'admin')
9
10   const button = getByText('登录')
11   await fireEvent.click(button)
12
13   const store = useUserStore()
14   expect(store.login).toBeCalledWith({ username: 'admin', password: 'admin' })
15 })
```

- 90% 的 Vue 项目都使用了 `@vue/test-utils`

# Component Testing Example

```
1  it('message should show and disappear after seconds', async () => {
2    vi.useFakeTimers()
3    const { getByText, getByTitle, queryByText } = renderWithVuetify(AppMessage)
4    const store = useMessageStore()
5    store.messages = [
6      { text: '????', type: 'info', show: true, time: new Date(), id: 1 },
7    ]
8    getByText('??????')
9    await flushPromises()
10   getByText('???')
11   vi.runAllTimers()
12   await nextTick()
13   expect(queryByText('???')).toBeNull()
14   vi.useRealTimers()
15   const buttonEmpty = getByTitle('??????')
16   await fireEvent.click(buttonEmpty)
17   getByText('??????')
18 })
```

# Store Testing Example

```
1  vi.mock('@api/monitor', () => {
2    return {
3      getLastCorrectData: vi.fn(() => ({
4        data: [{ time_produced: '2022-04-30T13:00:00', id: 4 }],
5      })),
6    }
7  })
8  describe('Monitor Store', () => {
9    beforeEach(() => {
10      setActivePinia(createPinia())
11    })
12    it('Should remove CorrectData more than 24h ago', async () => {
13      const store = useMonitorStore()
14      ;(store.correctList as any) = [
15        { time_produced: '2022-04-20T13:00:00', id: 1 },
16        { time_produced: '2022-04-21T13:00:00', id: 2 },
17        { time_produced: '2022-04-30T12:00:00', id: 3 },
18      ]
19      await store.getLatestData()
20      expect(store.correctList.length).toBe(2)
21    })
22  })
```

# Cypress(WIP)

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

Upcoming

# Why not Vue 2.7?

- 性能提升
- 兼容性

Coming soon!



- 性能提升
- 兼容性 2.7
- 支持 type 检查
- css-bind 支持 sfc 的 API 与 Vue3 的 API



- Do one thing, do it well
- □□□□□□□□
- □□□□□□

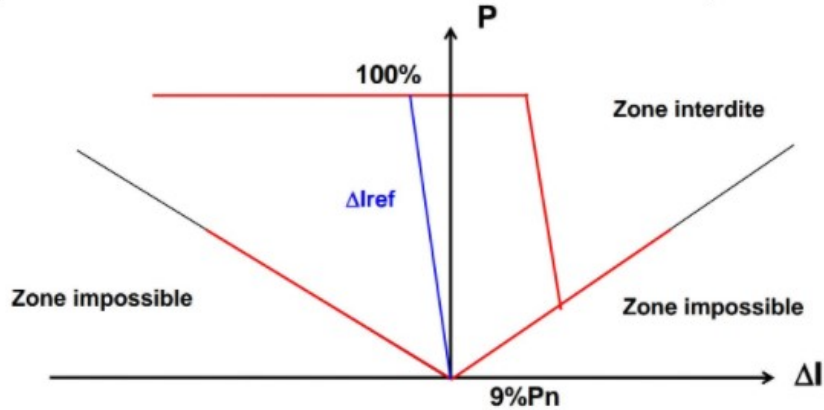
```
1  const weightedDist = computed(() => {
2    + const mesh = burnupMesh.value!
3    return dist.value
4      ? dist.value.map((layer, j) =>
5        -      layer.map((val, i) => (val * burnupMesh.value[j][i]) / activeHeight.value!)
6        +      layer.map((val, i) => (val * mesh[j][i]) / activeHeight.value!)
7      )
8      : undefined
9  })
```





- physical experiments / physics test
- tripleAO  $\rightarrow$  AO
- ladder  $\rightarrow$  trap
- ...

► **Trapèze de fonctionnement 1300 MWe - P,  $\Delta I$**



# Recap

- 项目 Viteiy Admin
- Vite 项目
- Vue3 项目
- 项目 E2E 测试
- 项目 部署
- 项目 部署



Thank You

based on docs.m