



Vue3 + Vite

Vue3, Vite, TypeScript

Vue3 & Vite

Vue3

`vue`

`vite`

```
npm install vue
npm install -D vite @vitejs/plugin-vue
```

`vite.config.js`

```
import { defineConfig } from 'vite'
import Vue from '@vitejs/plugin-vue'

export default defineConfig({
  plugins: [Vue()]
})
```

Vue

`src/main.js`

```
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).mount('#app')
```

`index.html`

```
<body>
  <div id="app"></div>
  <script type="module" src="./src/main.js"></script>
</body>
```

TypeScript

Vite

TypeScript

TypeScript

```
npm install -D typescript
```

JS TS

```
<script>
export default {
  // ...
}
</script>
```



```
<script lang="ts">
import { defineComponent } from 'vue'
export default defineComponent({
  // ...
})
</script>
```

`*.js` → `*.ts`

`tsconfig.json`

```
{
  "compilerOptions": {
    "target": "esnext",
    "module": "esnext",
    "moduleResolution": "node",
    "resolveJsonModule": true,
    "lib": ["esnext", "dom"],
    "jsx": "preserve",
    "strict": true,
    "esModuleInterop": true,
    "isolatedModules": true,
    "skipLibCheck": true,
    "preserveValueImports": true,
    "importsNotUsedAsValues": true,
    "forceConsistentCasingInFileNames": true,
    "baseUrl": "./",
  }
}
```

CSS

CSS / PostCSS Preset Env

Vite CSS

```
# .scss/.sass
npm install -D sass

# .less
npm install -D less

# .styl/.stylus
npm install -D stylus
```

CSS - PostCSS Preset Env

``postcss-preset-env``

```
npm install -D postcss-preset-env
```

``vite.config.ts``

```
import { defineConfig } from 'vite'
import PostCSSPresetEnv from 'postcss-preset-env'

export default defineConfig({
  css: {
    postcss: {
      plugins: [
        PostCSSPresetEnv({
          features: { 'nesting-rules': true }
        })
      ]
    }
  }
})
```

CSS

CSS

Tailwind CSS / Windi CSS / UnoCSS

```
<button class="py-1 px-4 rounded text-white bg-blue-400 active:bg-blue-500">CLICK</Button>
```

CLICK

```
npm install -D unocss
```

```
// vite.config.ts
import { defineConfig } from 'vite'
import UnoCSS from 'unocss/vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ]
})

// main.ts
import 'uno.css'
```

```
// unocss.config.ts
import {
  defineConfig, presetUno, presetIcons,
  transformerDirectives, transformerVariantGroup
} from 'unocss'

export default defineConfig({
  presets: [
    presetUno(),
    presetIcons()
  ],
  transformers: [
    transformerDirectives(),
    transformerVariantGroup()
  ]
})
```

- Vue Router 4.x

API

```
// router.ts
import {
  createRouter, createWebHistory
} from 'vue-router'

export default createRouter({
  history: createWebHistory(),
  routes: [ /* ... */ ]
})
```

```
// main.ts
import router from './router'
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).use(router).mount('#app')
```

```
<script lang="ts">
import { defineComponent } from 'vue'
import { useRouter, useRoute } from 'vue-router'

export default defineComponent({
  setup() {
    const router = useRouter()
    const route = useRoute()
    return { router, route }
  }
})
</script>

<template>
  <p>{{ route.name }}</p>
  <button @click="router.push('/about')">About</button>
</template>
```

- Pinia

pinia:

```
// main.ts
import { createPinia } from 'pinia'
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).use(createPinia()).mount('#app')
```

store

```
// store/counter.ts
import { defineStore } from 'pinia'
export const useCounterStore = defineStore('counter', {
  state: () => ({ count: 0 }),
  actions: {
    increment() {
      this.count++
    }
  }
})
```

store

```
<script lang="ts">
import { defineComponent } from 'vue'
import { useCounterStore } from './store/counter'

export default defineComponent({
  setup() {
    const counterStore = useCounterStore()
    return { counterStore }
  }
})
</script>

<template>
  <button @click="counterStore.increment">
    {{ counterStore.count }}
  </button>
</template>
```

Script Setup

SFC

```
<template>
  <HelloWorld />
  <button @click="increment">{{ count }}</button>
</template>
```

```
<script lang="ts">
import { ref, defineComponent } from 'vue'
import HelloWorld from './components/HelloWorld.vue'

export default defineComponent({
  components: { HelloWorld },
  setup() {
    const count = ref(0)
    function increment() {
      count.value++
    }
    return { count, increment }
  }
})
</script>
```

script setup



```
<script setup lang="ts">
import { ref } from 'vue'
import HelloWorld from './components/HelloWorld.vue'

const count = ref(0)
function increment() {
  count.value++
}
</script>
```


``unplugin-vue-components``

```
<script setup>
import HelloWorld from './components/HelloWorld.vue'
import Foo from './components/Foo.vue'
import Bar from './components/Bar.vue'
</script>
```

```
<template>
  <HelloWorld />
  <Foo />
  <Bar />
</template>
```



```
<template>
  <HelloWorld />
  <Foo />
  <Bar />
</template>
```

``unplugin-vue-components``

```
npm install -D unplugin-vue-components
```

``vite.config.ts``

```
import { defineConfig } from 'vite'
import Components from 'unplugin-vue-components/vite'

export default defineConfig({
  plugins: [
    Components({
      // default
      dirs: "src/components"
    })
  ]
})
```

API

API ``unplugin-auto-import``

```
import { computed, ref } from 'vue'
import { useRouter, useRoute } from 'vue-router'

const count = ref(0)
const doubled = computed(() => count.value * 2)
const router = useRouter()
const route = useRoute()
```



API

```
const count = ref(0)
const doubled = computed(() => count.value * 2)
const router = useRouter()
const route = useRoute()
```

``unplugin-auto-import``

```
npm install -D unplugin-auto-import
```

``vite.config.ts``

```
import { defineConfig } from 'vite'
import AutoImport from 'unplugin-auto-import/vite'

export default defineConfig({
  plugins: [
    AutoImport({
      imports: ['vue', 'vue-router']
    })
  ]
})
```

ESLint / Husky

eslint:

```
npm install -D eslint @yangss/eslint-config-vue
```

```
// .eslintrc.json
{
  "extends": [
    "@yangss/eslint-config-vue"
  ]
}
```

husky & lint-staged:

```
npm install -D husky lint-staged
npx husky install
npx husky add .husky/pre-commit "npx lint-staged"
```

```
// package.json
{
  // ...
  "lint-staged": {
    "*.ts|vue": "eslint --fix"
  }
}
```

Vitest & test-utils

- Vitest Vite
- Test Utils Vue

- VueUse
- UI Element Plus / Ant Design Vue / Vant UI
- vue-i18n

