

Quickstart - Vue 3 + Tailwind CSS (base NASA Explorer) FRANOUX Noé - CARETTE Robin

0. Objectif & contexte

- Projet "side project" : 1 h par seance pour prendre en main un framework UI.
- Binome, rendu avant le **15 octobre** (code + README + quickstart PDF).
- Ce guide montre comment demarrer une interface Vue 3 + Tailwind pour consommer l'API NASA APOD (avec fallback simulation en cas de shutdown).

1. Pre-requis

- Node.js ≥ 18
- npm ≥ 9 (ou pnpm/yarn)
- Editeur conseille : VS Code + extension Volar

```
node -v      # verifier >= 18
npm -v       # verifier >= 9
```

2. Creer le projet avec Vite

```
npm create vite@latest nasa-explorer -- --template vue
cd nasa-explorer
npm install
npm run dev      # http://localhost:5173
```

3. Installer Tailwind CSS

```
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

tailwind.config.js

```
export default {
  content: ["../index.html", "./src/**/*.vue,js,ts,jsx,tsx"],
  theme: { extend: {} },
  plugins: [],
}
```

src/style.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Importer ce CSS dans `src/main.js` :

```
import './style.css'
```

Relancer `npm run dev` pour vérifier que Tailwind est opérationnel.

4. App.vue minimal (layout responsive)

```
<template>
  <div class="min-h-screen bg-gradient-to-b from-slate-950 via-indigo-950 to-slate-950 text-
    <header class="border-b border-cyan-500/30 p-4 flex items-center justify-between">
      <h1 class="text-xl tracking-[0.4em] uppercase text-cyan-300">NASA Explorer</h1>
      <button class="rounded-full border border-cyan-400/50 px-3 py-1 text-xs uppercase tra
    </header>
    <main class="max-w-5xl mx-auto p-6">
      <slot />
    </main>
  </div>
</template>
```

5. Appeler l'API APOD avec Axios

`npm install axios`

`src/api/nasa.js`

```
import axios from 'axios'

const http = axios.create({
  baseURL: 'https://api.nasa.gov',
  timeout: 8000,
})

const API_KEY = import.meta.env.VITE_NASA_API_KEY || 'DEMO_KEY'

export async function fetchApod() {
  try {
    const { data } = await http.get('/planetary/apod', {
      params: { api_key: API_KEY, thumbs: true },
    })
    if (!data?.url) throw new Error('APOD payload incomplet')
    return { data, fallback: false }
  } catch (error) {
    console.warn('APOD indisponible, usage d'un fallback local.', error)
    return {
```

```

    data: {
      title: 'APOD (simulation)',
      explanation: 'Contenu de demonstration affiche pendant une indisponibilite de la NASA',
      url: '/assets/placeholders/apod-fallback.svg',
      media_type: 'image',
    },
    fallback: 'simulation',
  }
}
}

```

6. Composant ApodCard.vue

```

<template>
  <article class="rounded-3xl border border-white/10 bg-white/5 p-6 backdrop-blur">
    <header class="text-xs uppercase tracking-[0.3em] text-cyan-200 mb-4">Astronomy Picture of the Day</header>
    <div class="aspect-video rounded-2xl overflow-hidden border border-white/20 mb-4">
      
      <iframe v-else :src="apod.url" title="APOD Video" class="w-full h-full" allowfullscreen>
    </div>
    <h2 class="text-xl font-semibold">{{ apod?.title }}</h2>
    <p class="text-sm text-slate-300 mt-3">{{ apod?.explanation }}</p>
  </article>
</template>

<script setup>
defineProps({
  apod: { type: Object, default: () => null },
})
</script>

```

7. Utilisation dans App.vue

```

<script setup>
import { onMounted, ref } from 'vue'
import ApodCard from './components/ApodCard.vue'
import { fetchApod } from './api/nasa'

const apod = ref(null)
const fallback = ref(false)

onMounted(async () => {
  const result = await fetchApod()

```

```

    apod.value = result.data
    fallback.value = result.fallback
  })
</script>

<template>
  <div class="min-h-screen bg-slate-950 text-slate-100 p-6">
    <section class="space-y-4 max-w-3xl mx-auto">
      <div class="flex items-center gap-3">
        <h1 class="text-3xl font-semibold">Image du jour</h1>
        <span v-if="fallback" class="rounded-full border border-amber-400/50 px-3 py-1 text-
      </div>
      <ApodCard :apod="apod" />
    </section>
  </div>
</template>

```

8. Configuration .env.local

VITE_NASA_API_KEY=YOUR_KEY_HERE

- Sans cle, la valeur DEMO_KEY declenche automatiquement le fallback (image de simulation).
-

9. Mode responsive & menu mobile

- Ajouter un bouton hamburger (<button class="md:hidden ...">).
 - Quand mobileOpen est true, afficher un panneau plein ecran (position: fixed, backdrop-blur, bg-slate-950/90).
 - Bloquer le scroll : document.body.classList.toggle('overflow-hidden', mobileOpen.value).
 - Exemple de styles : halo cyan, liens uppercase avec tracking-[0.3em], croix de fermeture dediee.
-

10. Checklist finale

- npm run dev → l'APOD reel doit s'afficher si l'API repond.
 - Couper le reseau (DevTools → Offline) → verifier la carte en mode "Simulation".
 - Reduire la largeur (<768 px) → menu hamburger + panneau mobile.
-