

Visualización de Comentarios de Clientes

Estructura del Proyecto

- components/Comentarios.vue: Componente que muestra la lista de comentarios de clientes.
- server/api/comentarios/index.get.ts: Endpoint de la API que devuelve datos simulados de comentarios en JSON.
- stores/comentarios.ts: Store de Pinia que maneja el estado de los comentarios y obtiene los datos de la API.
- pages/index.vue: Página principal de la aplicación que muestra el componente Comentarios.

components/Comentarios.vue

Este componente será responsable de mostrar la lista de comentarios obtenidos del store

componentes > Comentarios.vue > {} template

```
1  <template>
2    <div>
3      <h1>Comentarios de Clientes</h1>
4      <div v-if="getComentarios.length">
5        <ul>
6          <li v-for="comentario in getComentarios" :key="comentario.id">
7            <h2>{{ comentario.title }}</h2>
8            <p>{{ comentario.description }}</p>
9            <strong>Satisfacción: {{ comentario.satisfaction }}</strong>
10           </li>
11         </ul>
12       </div>
13       <div v-else>
14         <p>No hay comentarios disponibles.</p>
15       </div>
16     </div>
17 </template>
18
19 <script setup>
20   // Importa el store configurado para los comentarios
21   import { useComentariosStore } from "~/stores/comentarios";
22
23   // Instancia el store
24   const comentariosStore = useComentariosStore();
25   const { getComentarios } = comentariosStore;
26
27   // Llama a `fetchComentarios` para cargar los datos
28   await comentariosStore.fetchComentarios();
29 </script>
30
```

server/api/comentarios/index.get.ts

Este archivo define el endpoint de la API para devolver comentarios simulados en formato JSON.

```
server > api > comentarios > index.get.ts > ...
1  export default defineEventHandler(async (event) => {
2      const comentarios = [
3          { id: 1, title: 'Comentario 1', satisfaction: 'Muy Satisfecho', description: 'Excelente servicio y atención.' },
4          { id: 2, title: 'Comentario 2', satisfaction: 'Satisfecho', description: 'El producto cumplió mis expectativas.' },
5          { id: 3, title: 'Comentario 3', satisfaction: 'Insatisfecho', description: 'La entrega tardó demasiado.' }
6      ];
7
8      return comentarios;
9  });
10
```

stores/comentarios.ts

Este archivo define el store use ComentariosStore utilizando Pinia para manejar el estado de los comentarios de manera centralizada.

```
stores > comentarios.ts > ...
1  import { defineStore } from "pinia";
2
3  interface Comentario {
4      id: number;
5      title: string;
6      satisfaction: string;
7      description: string;
8  }
9
10 export const useComentariosStore = defineStore("comentarios", {
11     state: () => ({
12         comentarios: [] as Comentario[], // Define `comentarios` como un array de tipo Comentario
13     }),
14     getters: {
15         getComentarios: (state) => state.comentarios,
16     },
17     actions: {
18         async fetchComentarios() {
19             try {
20                 const comentarios = await $fetch("/api/comentarios");
21                 this.comentarios = comentarios;
22             } catch (error) {
23                 console.error("Error al obtener los comentarios:", error);
24             }
25         },
26     },
27 });
28
```

pages/index.vue

Este archivo representa la página principal de la aplicación y utiliza el componente Comentarios para mostrar los comentarios de los clientes.

pages > ▼ index.vue

```
1  <template>
2    <h1>Página Principal</h1>
3    <Comentarios />
4  </template>
5
6  <script setup>
7    // Importa el componente `Comentarios.vue`
8    import Comentarios from '~/components/Comentarios.vue'
9  </script>
10
```

Realizado por: Brandon Joel serrano