TypeScript

- TypeScript is JavaScript with types. Think of it like an **enhanced** version of JavaScript with type checking.

- TypeScript projects have a **tsconfig.json** file at the top. They also require us to setup additional values (ESLint, file references, etc).

TypeScript Conversions

- We can use **interfaces** to define complex object types such as **Job** and our Pinia store states.
- TypeScript can infer many helpful details. It adds up!
- When we convert JavaScript files to TypeScript, we may encounter violations. TS will inform you when it is unsure of what type it’s working with.

The Partial Type

- The **Partial** type accepts a generic argument (type argument). It creates a new type where all the properties of the original type are optional.
  - Partial<Job>

- The **Partial** type can assist with factory functions in TypeScript.

TypeScript and Mocks

- TypeScript does not understand that **vi.mock** replaces an implementation with a Vitest mock function.

- We can use the **as** keyword to tell TS to treat a value as having a different *type*.
  - const axiosGetMock = axios.get as Mock;