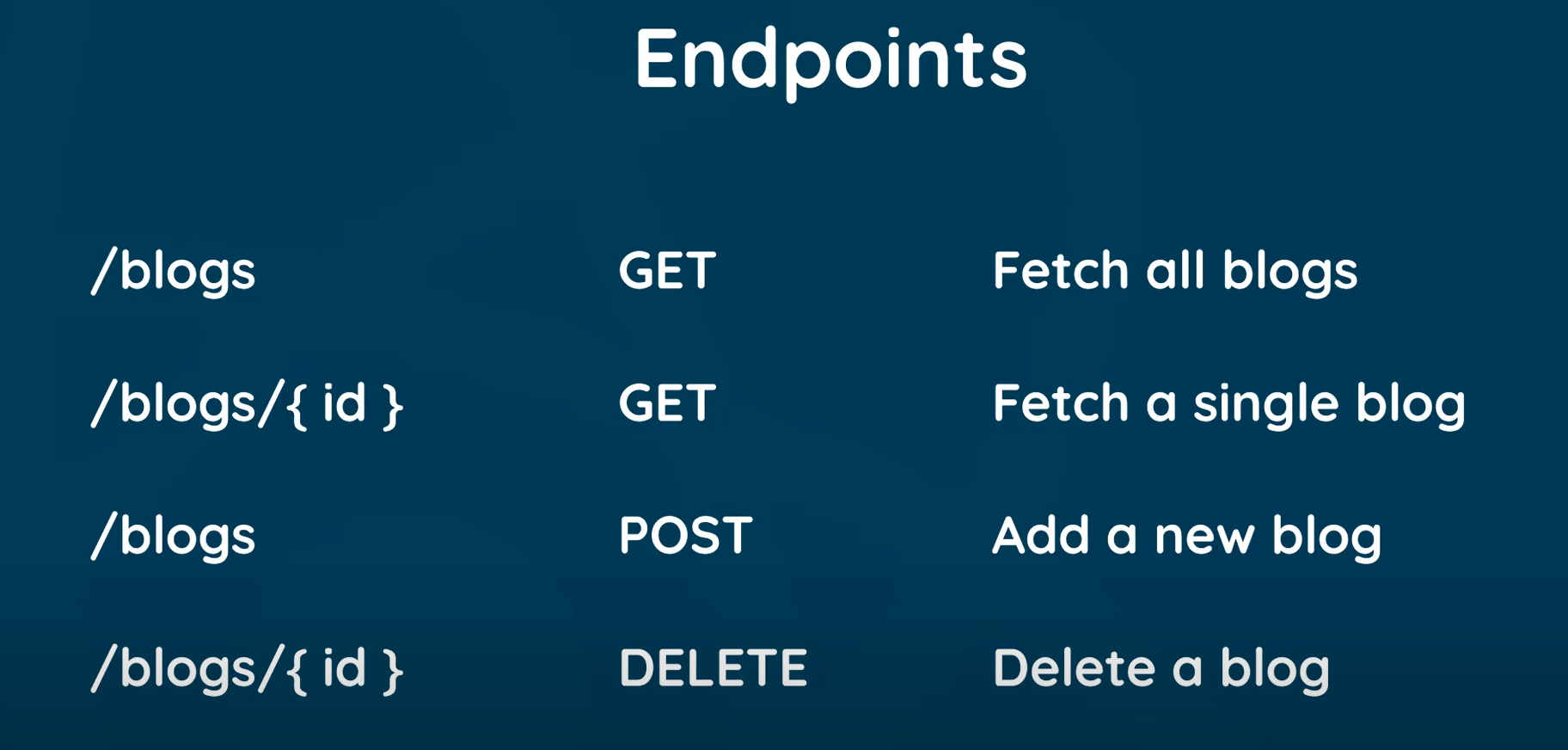
Create React app

|  |  |
| --- | --- |
| Creating a vite project | **npx create-react-app dojo-blog** |
|  | **cd to new pakige** |
|  | **npm start** |
| Install all depandesy (node-moduls folder – need if pushed from GitHub) | **npm install** |
| Instull pacige to VS code | **Simple React Snippets** |
| **Create stateless functional component** | **sfc + tab** |
| Quick nav className="navbar"> | **nav.navbar + enter** |
| duplicate a row in VS Code, you can use a simple keyboard shortcut: | **Shift + Alt + Down Arrow** |

|  |  |
| --- | --- |
| Chrome instanton for React  <https://chromewebstore.google.com/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi> | **react developer tools** |
| To get text in ant tag | **Lorem8** |

|  |  |
| --- | --- |
| used to set up a local ***REST API server*** using a JSON file (data-db.json) as the ***data source***. Here’s what each part of the command means: | **npx json-server --watch data/db.json --port 8000** |
|  | **http://localhost:8000/products** |
|  | **http://localhost:8000/blogs** |



**json-server** essentially creates a **fake (mock) database** using a JSON file. It mimics a real database by providing a fully functional REST API that you can use to perform operations like **GET, POST, PUT,** and **DELETE** requests. The data in your JSON file (**data-db.json**) acts as the source of truth, allowing you to quickly simulate how your application would behave when interacting with a back-end server.