1. **Json Server setup,**

Step 0: <https://www.npmjs.com/package/json-server>

Step 1: npm install -D json-server

Step 2: go to package.json, and add under scripts,

"server": "json-server --watch src/jobs.json --port 8000"

Full dependencies snippet,

  "dependencies": {

    "json-server": "^1.0.0-beta.1",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-icons": "^5.2.1",

    "react-router-dom": "^6.23.1",

    "server": "json-server --watch src/jobs.json --port 8000"

  },

Step 3: npm run server  
  
Code snippet,

  "scripts": {

    "dev": "vite",

    "build": "vite build",

    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",

    "preview": "vite preview",

   "server": "json-server --watch src/jobs.json --port 8000"

  },

1. **UseEffctHooks**

* UseEffect ta mainly API ke tanar jonnoi use hoy
* Er first parameters e thake ekta **arrow function** and second parameters e thake ekta empty array jeta ekbar e **execute** hoar jonno use hoy

**Full Snippet with fetching data from npm-json-server,**

  useEffect(() => {

    const fetchJobs = async () => {

      try {

        const res = await fetch("http://localhost:8000/jobs");

        const data = await res.json();

        console.log(data); // ekhane exactly executed data ta thake

      } catch (error) {

        console.log("Error fetching data");

      } finally {

        setLoading(false);

      }

    };

    fetchJobs(); ///calling the api function

  }, []);

True/False jsx example

  {loading ? (

              <h2>Loading</h2>

            ) : (

              <>

                {jobs.slice(0, 3).map((job, key) => (

                  <JobListing job={job} key={key} />

                ))}

              </>

            )}

1. **React Spinner**

**Step 1:** npm install react-spinners

Step 2: Go tho this link : <https://www.davidhu.io/react-spinners/>

Step 3: For instructions, go to this link, <https://github.com/davidhu2000/react-spinners>

Step 4: Based on the docs, ekta component create kora holo,

Creating a component named **Spinner.jsx** under **components** folder,

import React from "react";

import ClipLoader from "react-spinners/ClipLoader";

const override = {

  display: "block",

  margin: "100px auto",

};

const Spinner = ({ loading }) => {

  return (

    <ClipLoader

      color="#4338ca"

      loading={loading}

      cssOverride={override}

      size={150}

      //   aria-label="Loading Spinner"

      //   data-testid="loader"

    />

  );

};

export default Spinner;

* Ekhane loading er value as a **true** or **false** pass kora hoise as props wise.

Step 5: Ekhon jekoho components er moddhe as a component wise use kora jabe, Code snippet example added here from a **jsx** file,

          {loading ? (

            <Spinner loading={loading}></Spinner>

          ) : (

            <div className="grid grid-cols-1 md:grid-cols-3 gap-6">

              {jobs.slice(0, 3).map((job, key) => (

                <JobListing job={job} key={key} />

              ))}

            </div>

          )}

1. **Using Proxy**

Purpose : Onekta **.env** file er motoi purpose serve kore, for an example, <http://localhost:8000/> ei part tuku ke “/api” likhlei portion ta call hobe.

Step 1: **vite.config.js** e **proxy** configuration thik kore nite hobe

    proxy: {

      "/api": {

        target: "http://localhost:8000/",

        changeOrigin: true,

        rewrite: (path) => path.replace(/^\/api/, ""), // this means, /api/jobs, it will hit http://localhost:8000/jobs

      },

    },

Full code snippet,

import { defineConfig } from "vite";

import react from "@vitejs/plugin-react";

// https://vitejs.dev/config/

export default defineConfig({

  plugins: [react()],

  server: {

    port: 3000,

    proxy: {

      "/api": {

        target: "http://localhost:8000/",

        changeOrigin: true,

        rewrite: (path) => path.replace(/^\/api/, ""), // this means, /api/jobs, it will hit http://localhost:8000/jobs

      },

    },

  },

});

Step 2 : jekono place e call korar time e “/api” evabe call korlei kaj korbe, an example with **useEffect**,

const res = await fetch("/api/jobs");

Full code snippet,

  useEffect(() => {

    const fetchJobs = async () => {

      try {

        const res = await fetch("/api/jobs");

        const data = await res.json();

        setJobs(data);

      } catch (error) {

        console.log("Error fetching data");

      } finally {

        setLoading(false);

      }

    };

    fetchJobs();

  }, []);