

use command --> `npx json-server --watch ./data/userData.json --port 8000`

OR

Using ``npx`` to run ``json-server`` is another convenient way to quickly start a JSON server without installing it as a dependency. Here's how you can use ``npx json-server`` in your React project:

Step 1: Create your React project (if not already done)

```
```bash
npx create-react-app my-app
cd my-app
```
```

Step 2: Create your JSON data file

Create a directory named ``data`` in the root of your project and add a file named ``userData.json`` with your desired data. For example:

```
```json
// data/userData.json
{
 "users": [
 { "id": 1, "name": "John Doe", "email": "john@example.com" },
 { "id": 2, "name": "Jane Doe", "email": "jane@example.com" }
]
}
```
```

Step 3: Start the JSON server using ``npx``

You can start the JSON server with the following command:

```
```bash
npx json-server --watch ./data/userData.json --port 8000
```
```

This will start the JSON server at ``http://localhost:8000``.

Step 4: Fetch data in your React app

In your React component, you can now fetch data from the JSON server. Here's an example:

```
```jsx
import React, { useState, useEffect } from 'react';

function App() {
 const [users, setUsers] = useState([]);

 useEffect(() => {
 fetch('http://localhost:8000/users')
 .then(response => response.json())
 .then(data => setUsers(data));
 }, []);
}
```

```

 return (
 <div>
 <h1>Users</h1>

 {users.map(user => (
 <li key={user.id}>
 {user.name} ({user.email})

))}

 </div>
);
 }
}

export default App;
`);

```

### ### Step 5: Running both servers concurrently

To run both the React development server and the JSON server concurrently, you can use the `concurrently` package. First, install it:

```

`bash
npm install concurrently --save-dev
`);

```

Then, update your `package.json` scripts:

```

`json
"scripts": {
 "start": "concurrently \"npx json-server --watch ./data/userData.json --port 8000\" \"react-scripts start\"",
 "build": "react-scripts build",
 "test": "react-scripts test",
 "eject": "react-scripts eject"
}
`);

```

Now, running `npm start` will start both the React development server and the JSON server concurrently.

### ### Summary

Using `npx json-server` allows you to quickly set up and run a JSON server for your React application without needing to install it globally or as a project dependency. This is a great option for rapid prototyping and development.