JS Fetch CRUD Operations

now let's say i have a json-server at localhostblabla/jobs, i want to fetch it in js and get its data, i know how to do that, but how do I modify it and add elements to it? how do i remove ones and how to change ones (i'm in js)

You can modify data in your `json-server` using JavaScript's `fetch` API with `POST`, `PUT`, `PATCH`, and `DELETE` requests. Here's how you can do it:

1️⃣ Get Data (You already know this)

```js
fetch("http://localhost:3000/jobs")
  .then(res => res.json())
  .then(data => console.log(data))
  .catch(err => console.error(err));
```

2️⃣ Add a New Job (`POST`)

This will create a new job entry in the database.

```js
fetch("http://localhost:3000/jobs", {
  method: "POST",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify({
    title: "Software Engineer",
    company: "TechCorp",
    salary: 5000
  })
})
  .then(res => res.json())
  .then(data => console.log("Added:", data))
  .catch(err => console.error(err));
```

3️⃣ Update an Existing Job (`PUT` or `PATCH`)

- `PUT` replaces the whole object.
- `PATCH` updates only the given fields.

✅ Full update with `PUT` (replaces existing object)

```js
fetch("http://localhost:3000/jobs/1", {
  method: "PUT",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify({
    id: 1,  // You must send the `id` again with `PUT`
    title: "Senior Software Engineer",
    company: "TechCorp",
    salary: 7000
  })
})
  .then(res => res.json())
  .then(data => console.log("Updated:", data))
  .catch(err => console.error(err));
```

✅ Partial update with `PATCH` (modifies only specific fields)

```js
fetch("http://localhost:3000/jobs/1", {
  method: "PATCH",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify({
    salary: 8000  // Only updating the salary
    })
  })
    .then(res => res.json())
    .then(data => console.log("Updated:", data))
    .catch(err => console.error(err));
```

4️⃣ Delete a Job (`DELETE`)

```js
fetch("http://localhost:3000/jobs/1", {
  method: "DELETE"
})
  .then(() => console.log("Deleted job with ID 1"))
  .catch(err => console.error(err));
```

TL;DR:

Action	Method	URL Example	Body
Get All Jobs	`GET`	`/jobs`	❌
Add a Job	`POST`	`/jobs`	`{title, company, salary}`
Replace a Job	`PUT`	`/jobs/:id`	`{id, title, company, salary}`
Update Some Fields	`PATCH`	`/jobs/:id`	`{salary: 8000}`
Delete a Job	`DELETE`	`/jobs/:id`	❌

This is how you **add, update, and delete** elements dynamically from your JSON server. W method? 🚀

```
fetch(`http://localhost:3000/jobs/${toChange}`, {
                method: 'PATCH',
                headers: {
                    'Content-Type': 'application/json'
                },
          body: JSON.stringify({
            titre: jobTitle,
            description: jobDescription,
            salaire: jobSalary,
            'date de création': jobDate

        })})
```

for some reason i get this error:

```
_ctx.fetch is not a function
TypeError: _ctx.fetch is not a function
    at onClick._cache.<computed>._cache.<computed>
(webpack-internal:///./node_modules/babel-
loader/lib/index.js??clonedRuleSet-
40.use[0]!./node_modules/vue-
loader/dist/templateLoader.js??
ruleSet[1].rules[3]!./node_modules/vue-
loader/dist/index.js??
ruleSet[0].use[0]!./src/views/MainPage.vue?
vue&type=template&id=03d9b622&scoped=true:61:14)
    at callWithErrorHandling (webpack-
internal:///./node_modules/@vue/runtime-
core/dist/runtime-core.esm-bundler.js:387:19)
    at callWithAsyncErrorHandling (webpack-
internal:///./node_modules/@vue/runtime-
core/dist/runtime-core.esm-bundler.js:394:17)
    at HTMLButtonElement.invoker (webpack-
internal:///./node_modules/@vue/runtime-
dom/dist/runtime-dom.esm-bundler.js:941:83)
```

Yo, my dude, it looks like Vue is tripping over your use of `fetch`—it's trying to call `_ctx.fetch` which ain't defined. 😅 This sometimes happens in Vue when the context gets a bit messy.

Here are a couple of things to try:

1. **Use the Global Fetch:**
   Instead of just `fetch(...)`, try using `window.fetch(...)` so you're explicitly calling the global fetch function.

   ```js
   ```

```
window.fetch(`http://localhost:3000/jobs/${toChange}`, {
  method: 'PATCH',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    titre: jobTitle,
    description: jobDescription,
    salaire: jobSalary,
    'date de création': jobDate
  })
})
.then(res => res.json())
.then(data => console.log("Updated job:", data))
.catch(err => console.error(err));
```

2. **Check for Conflicts:**
   Make sure you don’t have a property or method in your Vue component called `fetch` that might be shadowing the global function. Sometimes that can lead to confusion in the context (`_ctx`).
3. **Component Setup:**
   If you're using Vue's composition API or similar, verify that you’re not accidentally binding something to `this.fetch` in your setup that could override the global fetch.

Try these out, and if it still gives you grief, hit me up! We're here to get that code to work like a champ. W fr, we got this! 🚀😂

2/2

> ↳ fetch("http://localhost:3000/jobs", { method: "POST", headers: { "Content-Type": "application/json" }, body: JSON.stringify({ title: "Software Engineer", company: "TechCorp", salary: 5000 }) }) .then(res => res.json()) .then(data => console.log("Added:", data)) .catch(err =>...

will this add an id automatically?

Yeah, my G, if you're using `json-server`, it will **automatically generate an** `id` for you when you add a new entry with `POST`. 🚀

How does it work?

- `json-server` uses a simple in-memory database (or a file like `db.json`).

- When you send a `POST` request **without specifying an** `id`, it **automatically assigns one** (incrementing from the highest existing ID).
- Example:

Sending this:

```json
{
  "title": "Software Engineer",
  "company": "TechCorp",
  "salary": 5000
}
```

json-server will return something like:

```json
{
  "id": 1,
  "title": "Software Engineer",
  "company": "TechCorp",
  "salary": 5000
}
```

So **you don’t need to manually add an** `id` unless you want to override it.

💡 **Important:** If you delete all entries and then add a new one, the `id` might not reset to 1 (it keeps incrementing unless you manually reset the database file).

W json-server, W automation. 🔥😂