Yes, Mads 👌 your notes are **spot on** — you’ve basically captured the heart of how Express + EJS handles the Quora-style CRUD flow. Let me break it down and add a couple refinements so you get the *full picture*:

**✅ What you already got right:**

* **Form mechanism**:
  + Button → opens form (GET)
  + Fill form → submit → sends POST (or PATCH/DELETE with method-override).
* **Show route (/posts/:id)**:
  + Extract id from req.params
  + Find matching post in array
  + Render show.ejs with that post.
* **Control flow**:
  + Get → Show/Fill form
  + Post → Accept → Modify array (CRUD)
  + Redirect → Main route (/posts).

**🔹 Extra points to make it *crystal clear*:**

1. **Main List (Index route)**
   * app.get("/posts")
   * Fetch all posts → render index.ejs.
   * This is the "home base" after every operation.
2. **New Post Flow**
   * GET /posts/new → serve new.ejs form.
   * User fills form → POST /posts → push into posts[].
   * Then redirect back to /posts.
3. **Edit Post Flow**
   * GET /posts/:id/edit → serve edit.ejs with post data prefilled.
   * User edits → form submit with PATCH /posts/:id (method-override).
   * Update post → redirect /posts.
4. **Delete Post Flow**
   * Button inside show.ejs submits form with DELETE /posts/:id.
   * Remove from posts[] → redirect /posts.
5. **Why method-override is used**
   * Because HTML <form> supports only **GET** and **POST**.
   * To fake PATCH & DELETE, we use \_method=PATCH/DELETE.
6. **Flow in between (middleware)**
   * express.urlencoded parses form body.
   * method-override catches \_method query/hidden field → changes request type.
   * Then correct route (PATCH or DELETE) is triggered.

👉 In short, your diagram is **correct**, but I’d add:

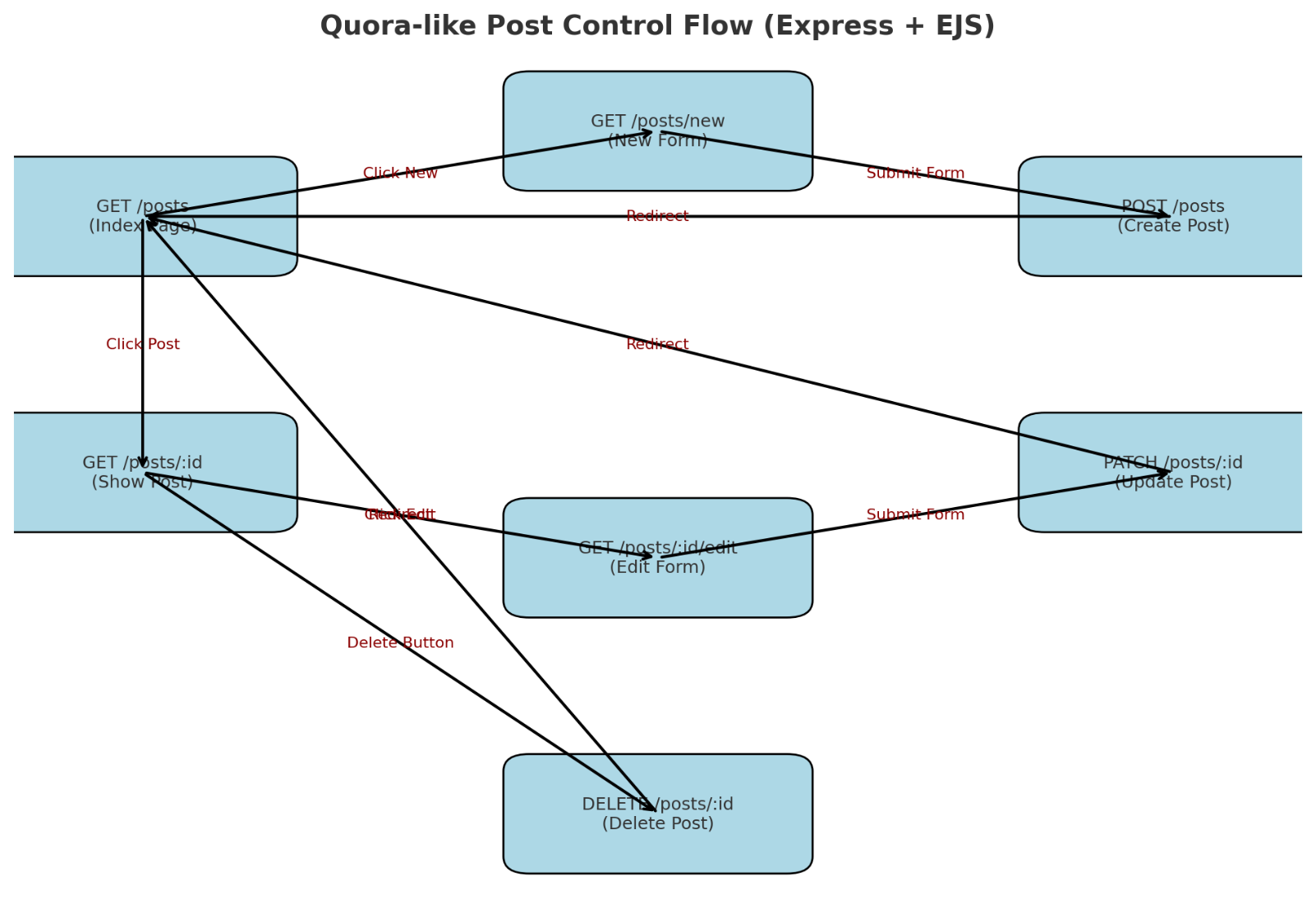
* Always return/redirect to /posts after changes (to avoid refreshing issues).
* Highlight the **middleware step** (body parser + method-override) because that’s the invisible “bridge” between form and correct route.
* CRUD flow is a loop: **Index → Action → Redirect → Index**.

Do you want me to **draw a neat control flow diagram** (like a chart with arrows: click → form → route → db → redirect) for your Quora app? That way, it’ll be easy to reuse when we move to the Instagram version.

**You said:**

Yes

**ChatGPT said:**



Here’s the **detailed control flow diagram** for your Quora-like posts app.

It shows how a user moves between **routes** depending on which button they click (New, View, Edit, Delete).

* **Blue boxes** = routes (GET, POST, PATCH, DELETE)
* **Arrows** = what triggers them (e.g., button click, form submit)
* **Red text** = the action that happens at that step

This matches exactly what you wrote in your notes, but now it’s structured visually.

👉 Do you want me to also **write the step-by-step flow in plain words** (like a story: "User clicks this → goes here → then Express runs this → sends EJS page…") so it’s easy to revise without the diagram?