

Steam Logistics Developer Exercise – Form Submit Component

You are implementing a Form Submit Component. This component lives at the bottom of a form acting as a command bar showing status and action buttons for the form. This will be evaluated for API design, correctness, and UX.

Implement this with a page to demo its behavior. Use Material UI or a similar component/theming library. Use libraries like Zustand, Zod, React Hook Form, and React Query as needed, or substitute your own. No backend is provided for this exercise: mock one as needed.

The component has four key sub-components:

- Status Icon
- Status Message, if any, that can have links embedded
- Submit button
- Reset button

There are several features:

- If there are no changes, display a “no changes” message and make action buttons unclickable.
- If there are unsubmitted changes display the change count.
 - Submit button is clickable.
 - If those changes have validation errors, display the error count instead.
 - The error count is clickable, opening a popup showing all validation errors.
 - Make the submit button unclickable.
- Upon submit button click, run a callback.
 - Status icon/message should show “submitting” with an indeterminate progress spinner.
 - On error, status message becomes “error: failed to save...” with an icon indicating error.
 - Clickable to open a popup with full error info.
 - On success, show a “success” message and icon.
 - Make the action buttons unclickable again.
 - Go to the “unsubmitted changes” state as soon as a new change is made.

The component should be I/O and validation agnostic: its caller is responsible for that logic. Instead, it takes any needed state as props.

Any place the component displays a message or an icon, it should take fragments (or callbacks returning fragments) to allow a high degree of customizability.