DWARVES FOUNDATION

Form & Form Validation

Lap Nguyen
alan@d.foundation

Agenda

1. Native form elements
2. Form in React
3. Manage complex forms
4. Some best practices

Native form elements

- Reference
- Basic elements:
  - <input>
  - <select>
  - <textarea>
  - …
- Built-in validation

Form in React

Concept: Controlled vs uncontrolled input.

Manage complex forms

Management libraries:

- React-hook-form <
- Formik

Validation libraries:

- Zod <
- Yup

Manage complex forms

react-hook-form

- Hook-based & functional approach to form management
- Leverages existing HTML markup (element)
- Light-weight: 8.5kb (Formik 15kb, Redux Form 26.4kb)
- Performant by minimizing re-renders & validate computation
- Built-in support for validation libraries such as Zod or Yup
- Focus on simplicity & great DX

Zod

- TypeScript-first schema declaration and validation library
- Zero dependencies
- Light-weight: 8kb minified + zipped
- Concise, chainable interface
- Functional approach: parse, don't validate

Some best practices

- Use meaningful HTML elements: <form>, <label>, <input>, …
- Label each input with <label>
- Use element attributes to access built-in browser features: type, name, id, for, autocomplete, …
- Prioritize native form controls over custom-built
- Ensure successful form submission with visual cues
- Design forms with UX in mind
    - Accessibility
    - Validation

References

- Validation with Zod
- WAI-ARIA compliant combobox
- Sign-in form best practices

DWARVES FOUNDATION

Thank You

DWARVES FOUNDATION

Q&A