

React Form

2024 년

Login

Email

Password

Submit

React Form

■ 폼을 다루는 방법

- 서버로 요청하기 전에 해당 필드의 값을 어떻게 얻어낼 수 있는지에 대한
 1. State와 Handler 사용하는 방법
 2. Ref를 사용하는 방법
 3. FormData와 브라우저 API 사용
 4. **Custom Hook 사용하기**
 5. React Forms Library

1. State와 Handler 사용하는 방법

■ Exam01.js

2. Ref를 사용하는 방법

3. FormData와 브라우저 API 사용

- 복잡한 form 구조를 다룰 때 사용
- state나 ref를 통해 각 필드에 연결하는 방식은 너무 비효율적인 방법

Signup

Email

Password

Gender

 ▼

Quantity

☐ 1 ☐ 2 ☐ 3

Terms

☐ I agree to the Terms and Conditions

4. Custom Hook 사용하기

- 여러 개의 입력값 상태 및 반복되는 onChange 이벤트를 재사용하기 위해서는 어떻게 해야 할까?
 - ☞ 커스텀 훅 사용

5. React Form Library

1. react hook forms

- `npm install --save react-hook-form`
- `useForm`은 상태와 validation을 관리하기 위해 사용된다.
- `register` 함수는 form의 입력필드를 등록하기 위해 사용되고 각 필드의 props를 리턴한다.
- `handleSubmit` 함수는 `onSubmit`을 넘기며 form 전송을 처리하기 위해 사용된다.

2. Formik

- `npm install --save formik`
- `initialValues` prop은 form 입력값의 초기값을 정의하고
- `validate` 함수는 form validation에 사용된다.
- `onSubmit` 함수는 form이 submit되었을 때 수행된다.

6. 동적 필드 생성

- 동적으로 추가하거나 제거 및 수정이 필요한 폼을 만들어야 하는 경우가 생깁니다.

패턴	의미
a-zA-Z	영어알파벳(-으로 범위 지정)
ㄱ-ㅎ가-힣	한글 문자(-으로 범위 지정)
0-9	숫자(-으로 범위 지정)
.	모든 문자열(숫자, 한글, 영어, 특수기호, 공백 모두! 단, 줄바꿈X)
\d	숫자
\D	숫자가 아닌 것
\w	영어 알파벳, 숫자, 언더스코어(_)
\W	/w 가 아닌 것
\s	space 공백
\S	space 공백이 아닌 것
\특수기호	특수기호

기호	의미
	OR
[]	괄호안의 문자들 중 하나
[^문자]	괄호안의 문자를 제외한 것
^문자열	특정 문자열로 시작(괄호 없음 주의!)
문자열\$	특정 문자열로 끝남
()	그룹 검색 및 분류(match메서드에서 그룹별로 묶어줌)
(?: 패턴)	그룹 검색(분류X)
\b	단어의 처음/끝
\B	단어의 처음/끝이 아님

기호	의미
?	최대 한번(없음 한개)
*	없거나 있거나 (없음 있음): 여러개 포함
+	최소 한개(한개 여러개)
{n}	n개
{Min,}	최소 Min개 이상
{Min, Max}	최소 Min개 이상, 최대 Max개 이하

메서드	의미
("문자열").match(/정규표현식/플래그)	"문자열"에서 "정규표현식"에 매칭되는 항목들을 배열로 반환
("문자열").replace(/정규표현식/, "대체문자열")	"정규표현식"에 매칭되는 항목을 "대체문자열"로 변환
("문자열").split(정규표현식)	"문자열"을 "정규표현식"에 매칭되는 항목으로 쪼개어 배열로 반환
(정규표현식).test("문자열")	"문자열"이 "정규표현식"과 매칭되면 true, 아니면 false반환
(정규표현식).exec("문자열")	match메서드와 유사(단, 무조건 첫번째 매칭 결과만 반환)

플래그	의미
g	Global: 모든 문자 검색(안 쓰면 매칭되는 첫 문자만 검색)
i	Ignore Case: 대소문자 구분 안함
m	Multi line: 여러 행의 문자열에 대해 검색

```
> /^d{4}.d{2}.d{2}$/.test("2018.01.02")
< true
> /^d{3}-d{3,4}-d{4}$/.test("010-1234-5678")
< true
> /^w+@w+\.w+$/.test("qwdqwd@naver.com")
< true
```