



API 참고서 > HOOK >

useActionState

useActionState 는 폼 액션의 결과를 기반으로 State를 업데이트할 수 있도록 제공하는 Hook입니다.

```
const [state, formAction, isPending] = useActionState(fn, initialState, p
```

▣ 중요합니다!

이전 React Canary 버전에서는 이 API가 React DOM에 포함되어 있었고, useFormState 라고 불렸습니다.

- [레퍼런스](#)
 - `useActionState(action, initialState, permalink?)`
- [사용법](#)
 - 폼 액션에서 반환된 정보 사용하기
- [문제 해결](#)
 - 액션이 더 이상 제출된 폼 데이터를 읽을 수 없습니다

레퍼런스

`useActionState(action, initialState, permalink?)`

`useActionState` 를 컴포넌트의 최상위 레벨에서 호출하여 폼 액션이 실행될 때 업데이트되는 컴포넌트 State를 생성하세요. `useActionState` 는 기존의 폼 액션 함수와 초기 State를 전달받고, 폼에서 사용할 새로운 액션을 반환합니다. 또한 최신 폼 State와 액션이 대기 중인지 여부 (`isPending`)도 반환합니다. 이때 최신 폼 State는 `useActionState` 에 전달한 함수에도 함께 전달됩니다.

```
import { useState } from "react";

async function increment(previousState, formData) {
  return previousState + 1;
}

function StatefulForm({}) {
  const [state, formAction] = useState(increment, 0);
  return (
    <form>
      {state}
      <button formAction={formAction}>Increment</button>
    </form>
  );
}
```

폼 State는 폼을 마지막으로 제출했을 때 액션에서 반환되는 값입니다. 아직 폼을 제출하지 않았다면, `initialState` 로 설정됩니다.

서버 함수Server Function와 함께 사용하는 경우, `useActionState` 를 통해 하이드레이션Hydration이 끝나기 전에도 폼 제출에 대한 서버 응답을 표시할 수 있습니다.

[아래 예시를 참고하세요.](#)

매개변수

- `fn`: 폼이 제출되거나 버튼이 눌렸을 때 호출되는 함수입니다. 함수가 호출되면 첫 번째 인수로 폼의 이전 State(처음에는 전달한 `initialState`, 이후에는 이전 반환값)가 전달되고, 그 뒤로는 폼 액션이 일반적으로 받는 인수들이 전달됩니다.
- `initialState`: State가 처음에 가지기를 원하는 값입니다. 이는 직렬화 가능한 값이면 무엇이든 될 수 있습니다. 이 인수는 액션이 처음 호출된 후에는 무시됩니다.

- **optional permalink**: 이 품이 수정하는 고유한 페이지 URL을 포함하는 문자열입니다. 동적 콘텐츠가 있는 페이지(예: 피드)에서 점진적 향상^{Progressive Enhancement}과 함께 사용됩니다. 만약 fn이 [서버 함수](#)이고, 품이 자바스크립트 번들이 로드되기 전에 제출되면, 브라우저는 현재 페이지의 URL 대신 지정된 영구 링크^{Permalink} URL로 이동합니다. React가 State를 전달하는 방법을 알 수 있도록, 동일한 품 컴포넌트가 대상 페이지에 렌더링되도록 해야 합니다. (동일한 액션 fn과 permalink 포함.) 품이 하이드레이션된 후, 이 매개변수는 더 이상 효과가 없습니다.

반환값

`useActionState`는 다음 세 가지 값을 담은 배열을 반환합니다.

1. 현재 State입니다. 첫 렌더링 시에는 `initialState`와 일치하며, 액션이 실행된 후에는 해당 액션이 반환한 값과 일치합니다.
2. `form` 컴포넌트의 `action` Prop이나, 품 내부 `button` 컴포넌트의 `formAction` Prop에 전달할 수 있는 새 액션입니다. 이 액션은 `startTransition` 내에서 수동으로 호출할 수도 있습니다.
3. 현재 Transition이 대기 중인지 알려주는 `isPending` 플래그입니다.

주의 사항

- React 서버 컴포넌트를 지원하는 프레임워크에서 `useActionState`를 사용하면, 클라이언트 자바스크립트 실행 전에도 품과 상호작용할 수 있습니다. 만약 서버 컴포넌트를 사용하지 않는다면, 이는 단순히 컴포넌트 지역 State와 동일하게 동작합니다.
- `useActionState`에 전달된 함수는 첫 번째 인수로 이전 또는 초기 State를 추가로 받습니다. 즉, 직접 품 액션을 사용했을 때와 비교해 함수의 시그니처가 달라질 수 있습니다.

사용법

폼 액션에서 반환된 정보 사용하기

컴포넌트의 최상위 레벨에서 `useActionState`를 호출하면, 품이 마지막으로 제출되었을 때 액션이 반환한 값에 접근할 수 있습니다.

```
import { useActionState } from 'react';
import { action } from './actions.js';
```

```
function MyComponent() {  
  const [ state, formAction ] = useActionState( action, null );  
  // ...  
  return (  
    <form action={ formAction }>  
      {/* ... */}  
    </form>  
  );  
}
```

useActionState 가 반환하는 배열은 다음과 같은 요소를 갖습니다.

1. 폼의 현재 State 는, 처음에는 전달한 초기 State 로 설정되며, 폼이 제출된 후에는 전달한 액션 의 반환값으로 설정됩니다.
2. <form> 의 action Prop에 전달하거나 startTransition 안에서 직접 호출할 수 있는 새로운 액션 입니다.
3. 액션이 처리되는 동안 사용할 수 있는 대기 Pending State 입니다.

폼이 제출되면, 제공한 액션 함수가 호출되며, 해당 함수의 반환값이 새로운 현재 State 로 설정됩니다.

이 액션 함수는 첫 번째 인수로 현재 State 를 추가로 전달받습니다.

처음 제출될 때는 초기 State 가 전달되며, 이후 제출부터는 직전 호출 시 반환된 값이 전달됩니다. 나머지 인수들은 useActionState를 사용하지 않았을 때와 동일합니다.

```
function action( currentState, formData ) {  
  // ...  
  return 'next state';  
}
```

폼 제출 후 정보 표시하기

1. 오류 표시하기 2. 폼 제출 후 구조화된 정보 표시하기



예시 1 of 2: 오류 표시하기

서버 함수 Server Function에서 반환된 오류 메시지나 토스트 메시지를 표시하려면, 해당 액션을 `useActionState`로 감싸주세요.

App.js actions.js

⌚ 새로고침 × Clear ⌂ 포크

```
import { useState } from "react";
import { addToCart } from "./actions.js";

function AddToCartForm({itemID, itemTitle}) {
  const [message, formAction, isPending] = useState(addToCart, null);
  return (
    <form action={formAction}>
      <h2>{itemTitle}</h2>
      <input type="hidden" name="itemID" value={itemID} />
      <button type="submit">Add to Cart</button>
      {isPending ? "Loading..." : message}
    </form>
  );
}
```

▼ 자세히 보기

다음 예시

문제 해결

액션이 더 이상 제출된 폼 데이터를 읽을 수 없습니다

액션을 `useActionState`로 감싸면 첫 번째 인수로 “이전(또는 현재) State”가 추가됩니다. 따라서 일반적인 폼 액션과 달리, 제출된 폼 데이터는 두 번째 인수에서 확인해야 합니다.

```
function action(currentState, formData) {  
  // ...  
}
```

이전

다음

[Hook](#)

[useCallback](#)

 Meta Open Source

Copyright © Meta Platforms, Inc

uwu?

[React 학습하기](#)

[빠르게 시작하기](#)

[설치하기](#)

[UI 표현하기](#)

[상호작용성 더하기](#)

[State 관리하기](#)

[탈출구](#)

[API 참고서](#)

[React APIs](#)

[React DOM APIs](#)

[커뮤니티](#)

[행동 강령](#)

[팀 소개](#)

[더 보기](#)

[블로그](#)

[React Native](#)

문서 기여자

개인 정보 보호

감사의 말

약관

