

useActionState

August 17, 2026

useActionState

Kind: Function**Source:** `src/jsx/dom/hooks/index.ts`**Part of:** `Jsx`

This hook returns the current state and a function to update the state by form action

`useActionState` stores state produced by a form action and returns that state with a function for submitting updates. Pass the returned action function to a form's `action` prop so form data can drive the next state.

Signature

```
function useActionState(fn: Function, initialState: T, permalink: string): [T, Function]
```

Parameters

Name	Type
<code>fn</code>	<code>Function</code>
<code>initialState</code>	<code>T</code>
<code>permalink</code>	<code>string</code>

Returns: `[T, Function]`

Diagram

```
graph LR
  Form[Form submission] --> Action[Action function]
  Action --> Hook[useActionState]
  Hook --> State[Current state]
```

```
Hook --> FormAction[Form action handler]
FormAction --> Form
```

Usage

```
import { useActionState } from "your-library";

type FormState = {
  message: string;
};

async function saveName(
  previousState: FormState,
  formData: FormData,
): Promise<FormState> {
  const name = formData.get("name");

  if (typeof name !== "string" || name.trim() === "") {
    return { message: "Enter a name." };
  }

  await saveProfile({ name });

  return { message: `Saved ${name}.` };
}

export function ProfileForm() {
  const [state, formAction] = useActionState(saveName, {
    message: "",
  });

  return (
    <form action={formAction}>
      <label>
        Name
        <input name="name" />
      </label>

      <button type="submit">Save</button>

      {state.message && <p>{state.message}</p>}
    </form>
  );
}
```

AI Coding Instructions

- Pass an action function that accepts the previous state and submitted form data, then returns the next state.
- Keep the initial state shape aligned with every state returned by the action.
- Attach the returned action function directly to a form's `action` prop.
- Validate `FormData` values before using them, since values may be missing or have an unexpected type.