# React 19 Hooks

## useActionState

Manage an async action’s state + submit + pending in one hook—great with forms and server actions.  
Syntax:

const initial = { error:null, data:null };  
const [state, submit, isPending] = useActionState(loginAction, initial);  
<form action={submit}>…</form>

## useFormStatus

Read the nearest form’s pending status to disable buttons / show spinners without prop-drilling.  
Syntax:

const { pending } = useFormStatus();  
<button disabled={pending}>{pending ? "Submitting…" : "Submit"}</button>

## useOptimistic

Do optimistic UI updates instantly; React reconciles when the real result returns.  
Syntax:

const [optimisticTodos, addOptimistic] = useOptimistic(todos, (s, t) => [...s, t]);  
async function onAdd(fd){  
 addOptimistic({id:"tmp", title:fd.get("title")});  
 await addTodo(fd);  
}

## Passing refs (React 19 – No forwardRef needed)

In React 19, function components can accept `ref` directly as a prop; forwardRef is no longer needed and will be deprecated.  
New way (React 19):

function FancyInput({ ref, ...props }) {  
 return <input ref={ref} {...props} />;  
}  
const inputRef = useRef(null);  
<FancyInput ref={inputRef} />

## Legacy (pre-19) for reference

const FancyInput = forwardRef((props, ref) => <input ref={ref} {...props} />);

## useAPI (custom pattern)

A small custom hook to standardize data / error / loading.  
Syntax:

const { data, error, loading, run } = useAPI(fetchUser);