

File: client/src/components/ui/input-otp.tsx

Type: tsx | Size: 2154 bytes

```
import * as React from "react"
import { OTPInput, OTPInputContext } from "input-otp"
import { Dot } from "lucide-react"

import { cn } from "@/lib/utils"

const InputOTP = React.forwardRef<
  React.ElementRef<typeof OTPInput>,
  React.ComponentPropsWithoutRef<typeof OTPInput>
>(({ className, containerClassName, ...props }, ref) => {
  <OTPInput
    ref={ref}
    containerClassName={cn(
      "flex items-center gap-2 has-[:disabled]:opacity-50",
      containerClassName
    )}
    className={cn("disabled:cursor-not-allowed", className)}
    {...props}
  />
))
InputOTP.displayName = "InputOTP"

const InputOTPGroup = React.forwardRef<
  React.ElementRef<"div">,
  React.ComponentPropsWithoutRef<"div">
>(({ className, ...props }, ref) => {
  <div ref={ref} className={cn("flex items-center", className)} {...props} />
})
InputOTPGroup.displayName = "InputOTPGroup"

const InputOTPSlot = React.forwardRef<
  React.ElementRef<"div">,
  React.ComponentPropsWithoutRef<"div"> & { index: number }
>(({ index, className, ...props }, ref) => {
  const inputOTPContext = React.useContext(OTPInputContext)
  const { char, hasFakeCaret, isActive } = inputOTPContext.slots[index]

  return (
    <div
      ref={ref}
      className={cn(
        "relative flex h-10 w-10 items-center justify-center border-y border-r border-input text-sm transition...
        isActive && "z-10 ring-2 ring-ring ring-offset-background",
        className
      )}
      {...props}
    >
      {char}
      {hasFakeCaret && (
        <div className="pointer-events-none absolute inset-0 flex items-center justify-center">
          <div className="h-4 w-px animate-caret-blink bg-foreground duration-1000" />
        </div>
      )}
    </div>
  )
})
InputOTPSlot.displayName = "InputOTPSlot"

const InputOTPSeparator = React.forwardRef<
  React.ElementRef<"div">,
  React.ComponentPropsWithoutRef<"div">
>(({ ...props }, ref) => {
  <div ref={ref} role="separator" {...props}>
    <Dot />
  </div>
})
InputOTPSeparator.displayName = "InputOTPSeparator"

export { InputOTP, InputOTPGroup, InputOTPSlot, InputOTPSeparator }
```