

File: client/src/components/ui/command.tsx

Type: tsx | Size: 4885 bytes

```
import * as React from "react"
import { type DialogProps } from "@radix-ui/react-dialog"
import { Command as CommandPrimitive } from "cmdk"
import { Search } from "lucide-react"

import { cn } from "@lib/utils"
import { Dialog, DialogContent } from "@components/ui/dialog"

const Command = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive>
>(({ className, ...props }, ref) => (
  <CommandPrimitive
    ref={ref}
    className={cn(
      "flex h-full w-full flex-col overflow-hidden rounded-md bg-popover text-popover-foreground",
      className
    )}
    {...props}
  />
))
Command.displayName = CommandPrimitive.displayName

const CommandDialog = ({ children, ...props }: DialogProps) => {
  return (
    <Dialog {...props}>
      <DialogContent className="overflow-hidden p-0 shadow-lg">
        <Command className="[&_cmdk-group-heading]:px-2 [&_cmdk-group-heading]:font-medium [&_cmdk-group-...
          {children}
        </Command>
      </DialogContent>
    </Dialog>
  )
}

const CommandInput = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive.Input>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive.Input>
>(({ className, ...props }, ref) => (
  <div className="flex items-center border-b px-3 cmdk-input-wrapper=" ">
    <Search className="mr-2 h-4 w-4 shrink-0 opacity-50" />
    <CommandPrimitive.Input
      ref={ref}
      className={cn(
        "flex h-11 w-full rounded-md bg-transparent py-3 text-sm outline-none placeholder:text-muted-foregroun...
        className
      )}
      {...props}
    />
  </div>
))
CommandInput.displayName = CommandPrimitive.Input.displayName

const CommandList = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive.List>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive.List>
>(({ className, ...props }, ref) => (
  <CommandPrimitive.List
    ref={ref}
    className={cn("max-h-[300px] overflow-y-auto overflow-x-hidden", className)}
    {...props}
  />
))
CommandList.displayName = CommandPrimitive.List.displayName

const CommandEmpty = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive.Empty>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive.Empty>
>
```

```

>((props, ref) => (
  <CommandPrimitive.Empty
    ref={ref}
    className="py-6 text-center text-sm"
    {...props}
  />
))

CommandEmpty.displayName = CommandPrimitive.Empty.displayName

const CommandGroup = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive.Group>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive.Group>
>(({ className, ...props }, ref) => (
  <CommandPrimitive.Group
    ref={ref}
    className={cn(
      "overflow-hidden p-1 text-foreground [&_cmdk-group-heading]:px-2 [&_cmdk-group-heading]:py-1.5 [&_c...
      className
    )}
    {...props}
  />
))

CommandGroup.displayName = CommandPrimitive.Group.displayName

const CommandSeparator = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive.Separator>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive.Separator>
>(({ className, ...props }, ref) => (
  <CommandPrimitive.Separator
    ref={ref}
    className={cn("-mx-1 h-px bg-border", className)}
    {...props}
  />
))

CommandSeparator.displayName = CommandPrimitive.Separator.displayName

const CommandItem = React.forwardRef<
  React.ElementRef<typeof CommandPrimitive.Item>,
  React.ComponentPropsWithoutRef<typeof CommandPrimitive.Item>
>(({ className, ...props }, ref) => (
  <CommandPrimitive.Item
    ref={ref}
    className={cn(
      "relative flex cursor-default gap-2 select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none...
      className
    )}
    {...props}
  />
))

CommandItem.displayName = CommandPrimitive.Item.displayName

const CommandShortcut = ({
  className,
  ...props
}: React.HTMLAttributes<HTMLSpanElement>) => {
  return (
    <span
      className={cn(
        "ml-auto text-xs tracking-widest text-muted-foreground",
        className
      )}
      {...props}
    />
  )
}

CommandShortcut.displayName = "CommandShortcut"

export {
  Command,
  CommandDialog,

```

```
CommandInput,  
CommandList,  
CommandEmpty,  
CommandGroup,  
CommandItem,  
CommandShortcut,  
CommandSeparator,  
}
```