import \* as React from "react"

import \* as ToastPrimitives from "@radix-ui/react-toast"

import { cva, type VariantProps } from "class-variance-authority"

import { X } from "lucide-react"

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

const ToastProvider = ToastPrimitives.Provider

const ToastViewport = React.forwardRef<

React.ElementRef<typeof ToastPrimitives.Viewport>,

React.ComponentPropsWithoutRef<typeof ToastPrimitives.Viewport>

>(({ className, ...props }, ref) => (

<ToastPrimitives.Viewport

ref={ref}

className={cn(

"fixed top-0 z-[100] flex max-h-screen w-full flex-col-reverse p-4 sm:bottom-0 sm:right-0 sm:top-auto sm:flex-col md:max-w-[420px]",

className

)}

{...props}

/>

))

ToastViewport.displayName = ToastPrimitives.Viewport.displayName

const toastVariants = cva(

"group pointer-events-auto relative flex w-full items-center justify-between space-x-4 overflow-hidden rounded-md border p-6 pr-8 shadow-lg transition-all data-[swipe=cancel]:translate-x-0 data-[swipe=end]:translate-x-[var(--radix-toast-swipe-end-x)] data-[swipe=move]:translate-x-[var(--radix-toast-swipe-move-x)] data-[swipe=move]:transition-none data-[state=open]:animate-in data-[state=closed]:animate-out data-[swipe=end]:animate-out data-[state=closed]:fade-out-80 data-[state=closed]:slide-out-to-right-full data-[state=open]:slide-in-from-top-full data-[state=open]:sm:slide-in-from-bottom-full",

{

variants: {

variant: {

default: "border bg-background text-foreground",

destructive:

"destructive border-destructive bg-destructive text-destructive-foreground",

},

},

defaultVariants: {

variant: "default",

},

}

)

const Toast = React.forwardRef<

React.ElementRef<typeof ToastPrimitives.Root>,

React.ComponentPropsWithoutRef<typeof ToastPrimitives.Root> &

VariantProps<typeof toastVariants>

>(({ className, variant, ...props }, ref) => {

return (

<ToastPrimitives.Root

ref={ref}

className={cn(toastVariants({ variant }), className)}

{...props}

/>

)

})

Toast.displayName = ToastPrimitives.Root.displayName

const ToastAction = React.forwardRef<

React.ElementRef<typeof ToastPrimitives.Action>,

React.ComponentPropsWithoutRef<typeof ToastPrimitives.Action>

>(({ className, ...props }, ref) => (

<ToastPrimitives.Action

ref={ref}

className={cn(

"inline-flex h-8 shrink-0 items-center justify-center rounded-md border bg-transparent px-3 text-sm font-medium ring-offset-background transition-colors hover:bg-secondary focus:outline-none focus:ring-2 focus:ring-ring focus:ring-offset-2 disabled:pointer-events-none disabled:opacity-50 group-[.destructive]:border-muted/40 group-[.destructive]:hover:border-destructive/30 group-[.destructive]:hover:bg-destructive group-[.destructive]:hover:text-destructive-foreground group-[.destructive]:focus:ring-destructive",

className

)}

{...props}

/>

))

ToastAction.displayName = ToastPrimitives.Action.displayName

const ToastClose = React.forwardRef<

React.ElementRef<typeof ToastPrimitives.Close>,

React.ComponentPropsWithoutRef<typeof ToastPrimitives.Close>

>(({ className, ...props }, ref) => (

<ToastPrimitives.Close

ref={ref}

className={cn(

"absolute right-2 top-2 rounded-md p-1 text-foreground/50 opacity-0 transition-opacity hover:text-foreground focus:opacity-100 focus:outline-none focus:ring-2 group-hover:opacity-100 group-[.destructive]:text-red-300 group-[.destructive]:hover:text-red-50 group-[.destructive]:focus:ring-red-400 group-[.destructive]:focus:ring-offset-red-600",

className

)}

toast-close=""

{...props}

>

<X className="h-4 w-4" />

</ToastPrimitives.Close>

))

ToastClose.displayName = ToastPrimitives.Close.displayName

const ToastTitle = React.forwardRef<

React.ElementRef<typeof ToastPrimitives.Title>,

React.ComponentPropsWithoutRef<typeof ToastPrimitives.Title>

>(({ className, ...props }, ref) => (

<ToastPrimitives.Title

ref={ref}

className={cn("text-sm font-semibold", className)}

{...props}

/>

))

ToastTitle.displayName = ToastPrimitives.Title.displayName

const ToastDescription = React.forwardRef<

React.ElementRef<typeof ToastPrimitives.Description>,

React.ComponentPropsWithoutRef<typeof ToastPrimitives.Description>

>(({ className, ...props }, ref) => (

<ToastPrimitives.Description

ref={ref}

className={cn("text-sm opacity-90", className)}

{...props}

/>

))

ToastDescription.displayName = ToastPrimitives.Description.displayName

type ToastProps = React.ComponentPropsWithoutRef<typeof Toast>

type ToastActionElement = React.ReactElement<typeof ToastAction>

export {

type ToastProps,

type ToastActionElement,

ToastProvider,

ToastViewport,

Toast,

ToastTitle,

ToastDescription,

ToastClose,

ToastAction,

}