```
'use client';


import * as React from 'react';

import * as SeparatorPrimitive from '@radix-ui/react-separator';


import { cn } from '@/lib/utils';


interface SeparatorProps

  extends React.ComponentPropsWithoutRef<typeof SeparatorPrimitive.Root> {

  className?: string;

  orientation?: 'horizontal' | 'vertical';

  decorative?: boolean;

}


/**

 * Separator component that renders a styled visual separator.

 * It can be oriented horizontally or vertically and can be decorative.

 *

 * @param className - Additional class names to apply to the separator.

 * @param orientation - Defines the orientation of the separator, either "horizontal" or "vertical".

 * @param decorative - If true, the separator is purely decorative and not announced by screen readers.

 * @param props - All other props supported by the Radix UI Separator component.

 * @param ref - A ref object to reference the underlying separator element.

 * @returns A separator element with the applied styles and props.

 */
```

```
const Separator = React.forwardRef<
  React.ElementRef<typeof SeparatorPrimitive.Root>,
  SeparatorProps
>(
  (
    { className, orientation = 'horizontal', decorative = true, ...props },
    ref,
  ) => (
    <SeparatorPrimitive.Root
      ref={ref}
      decorative={decorative}
      orientation={orientation}
      className={cn(
        'shrink-0 bg-border',
        orientation === 'horizontal' ? 'h-[1px] w-full' : 'h-full w-[1px]',
        className,
      )}
      {...props}
    />
  ),
);

Separator.displayName = SeparatorPrimitive.Root.displayName;

export { Separator };
```