

Framer Motion HOC Explanation

Vous avez dit :



```
import { motion } from "framer-motion";

import { styles } from "../styles";
import { staggerContainer } from "../utils/motion";

const SectionWrapper = (Component, idName) =>
function HOC() {
  return (
    <motion.section
      variants={staggerContainer()}
      initial="hidden"
      whileInView="show"
      viewport={{ once: true, amount: 0.25 }}
      className={` ${styles.padding} max-w-7xl mx-auto relative z-0`}
    >
      <span className="hash-span" id={idName}>
        &nbsp;
      </span>

      <Component />
    </motion.section>
  );
};

export default SectionWrapper;
```

ChatGPT a dit :



The code you provided defines a higher-order component (HOC) called `SectionWrapper`. This HOC wraps any component passed to it with a `motion.section` from Framer Motion, allowing the wrapped component to have animations and other properties.

Here's a brief explanation of the code:

1. Imports:

- `motion` from `framer-motion`: Used for animations.
- `styles` from `../styles`: Presumably contains the styling classes.
- `staggerContainer` from `../utils/motion`: A utility function to create staggered animations.

2. SectionWrapper HOC:

- `SectionWrapper` is a function that takes a `Component` and an `idName`.

- It returns a new function (HOC) that renders the `Component`` inside a `motion.section``.

3. `motion.section``:

- The `motion.section`` element wraps the `Component`` and applies the animation variants from `staggerContainer()``.
- `initial="hidden"``: Sets the initial state of the animation to "hidden."
- `whileInView="show"``: The animation plays when the component is in view.
- `viewport={{ once: true, amount: 0.25 }}``: The animation triggers when 25% of the component is in view and only once.

4. Styling and ID:

- The `className`` applies padding from the `styles`` and centers the component.
- The `span`` with `id={idName}`` provides a way to link to this section via the ID.

5. Export:

- `SectionWrapper`` is exported as the default export, so it can be used to wrap other components.

This HOC is useful for adding consistent section-level styling and animations to multiple components in your application. You can use it by wrapping your components with `SectionWrapper`` and passing an `idName`` for linking purposes.

ChatGPT peut faire des erreurs. Envisagez de vérifier les informations importantes.