Awesome Receipt App Style Guide

This list is inspired by AirBnb’s reasonable approach to React and JSX

Basic Rules

Only include one React component per file.

Always use JSX syntax.

Do not use React.createElement unless you’re initializing the app from a file that is not JSX.

Function components are preferred over Class components

Do not use mixins

Why? Mixins introduce implicit dependencies, cause name clashes, and cause snowballing complexity. Most use cases for mixins can be accomplished in better ways via components, higher-order components, or utility modules.

Naming

Use .jsx extension for React components

Use PascalCase for filenames. E.g., ReservationCard.jsx.

Use PascalCase for React components and camelCase for their instances

Component Naming: Use the filename as the component name. For example, ReservationCard.jsx should have a reference name of ReservationCard. However, for root components of a directory, use index.jsx as the filename and use the directory name as the component name:

Higher-order Component Naming: Use a composite of the higher-order component’s name and the passed-in component’s name as the displayName on the generated component. For example, the higher-order component withFoo(), when passed a component Bar should produce a component with a displayName of withFoo(Bar).

Why? A component’s displayName may be used by developer tools or in error messages, and having a value that clearly expresses this relationship helps people understand what is happening.

Props Naming: Avoid using DOM component prop names for different purposes.

Why? People expect props like style and className to mean one specific thing. Varying this API for a subset of your app makes the code less readable and less maintainable, and may cause bugs.

Declaration

Do not use displayName for naming components. Instead, name the component by reference.

Alignment

If props fit in one line then keep it on the same line

If props are too long indent and have a prop per line

Quotes

Always use double quotes (") for JSX attributes, but single quotes (') for all other JS

Why? Regular HTML attributes also typically use double quotes instead of single, so JSX attributes mirror this convention.

Spacing

Always include a single space in your self-closing tag

Props

Always use camelCase for prop names, or PascalCase if the prop value is a React component.

Omit the value of the prop when it is explicitly true

Always include an alt prop on <img> tags. If the image is presentational, alt can be an empty string or the <img> must have role="presentation"

Do not use words like "image", "photo", or "picture" in <img> alt props

Why? Screenreaders already announce img elements as images, so there is no need to include this information in the alt text.

Use only valid, non-abstract ARIA roles

Do not use accessKey on elements

Why? Inconsistencies between keyboard shortcuts and keyboard commands used by people using screen readers and keyboards complicate accessibility.

Always define explicit defaultProps for all non-required props.

Why? propTypes are a form of documentation, and providing defaultProps means the reader of your code doesn’t have to assume as much. In addition, it can mean that your code can omit certain type checks.

Use spread props sparingly.

Why? Otherwise you’re more likely to pass unnecessary props down to components. And for React v15.6.1 and older, you could pass invalid HTML attributes to the DOM.