React Native with NativeWind Project Structure

NativeWind (Tailwind CSS) styling. Use className instead of StyleSheet objects.

#react-native #expo #nativewind #tailwind #css #styling

PNG PDF Copy Prompt

Project Directory

```
my-app/
  app.json
  package.json
  tsconfig.json
  babel.config.js  NativeWind pres…
  tailwind.config.js  Tailwind config…
  metro.config.js  CSS support
  global.css  @tailwind direc…
  nativewind-env.d.ts  Type declaratio…
  .gitignore
> app/
    _layout.tsx  Import global.c…
    index.tsx
  > (tabs)/
      _layout.tsx
      index.tsx
      profile.tsx
> components/
    Button.tsx  className-styled
    Card.tsx
    Avatar.tsx
> lib/
    utils.ts  cn() helper for…
> hooks/
    useColorScheme.ts
> constants/
    theme.ts
> assets/
    images/
```

Why This Structure?

NativeWind v4 brings Tailwind CSS to React Native. Write `className="flex-1 bg-blue-500 p-4"` instead of StyleSheet objects. Same mental model as Tailwind web. Supports dark mode, arbitrary values, and most Tailwind utilities.

Key Directories

tailwind.config.js - Standard Tailwind config, custom colors/spacing

global.css - @tailwind base/components/utilities directives

app/_layout.tsx - Import global.css at root

lib/utils.ts - cn() helper for conditional classes

NativeWind Component

```tsx
// components/Button.tsx
import { Pressable, Text } from 'react-native';

export function Button({ title, onPress }) {
  return (
    <Pressable
      onPress={onPress}
      className="bg-blue-500 px-4 py-2 rounded-lg active:b
    >
      <Text className="text-white font-semibold text-cente
        {title}
      </Text>
    </Pressable>
  );
}
```

Getting Started

1. `npx create-expo-app@latest my-app`
2. `npx expo install nativewind tailwindcss`
3. Create `tailwind.config.js` and `global.css`
4. Configure `babel.config.js` with NativeWind preset
5. Import `global.css` in root `_layout.tsx`

When To Use This

- Team knows Tailwind from web development
- Want consistent styling between web and mobile
- Prefer utility classes over StyleSheet
- Building design systems with Tailwind patterns
- Rapid UI development with familiar utilities

Trade-offs

Setup complexity - Babel/Metro config required

Not all utilities - Some web-only Tailwind features unavailable

Bundle size - CSS processing adds overhead