

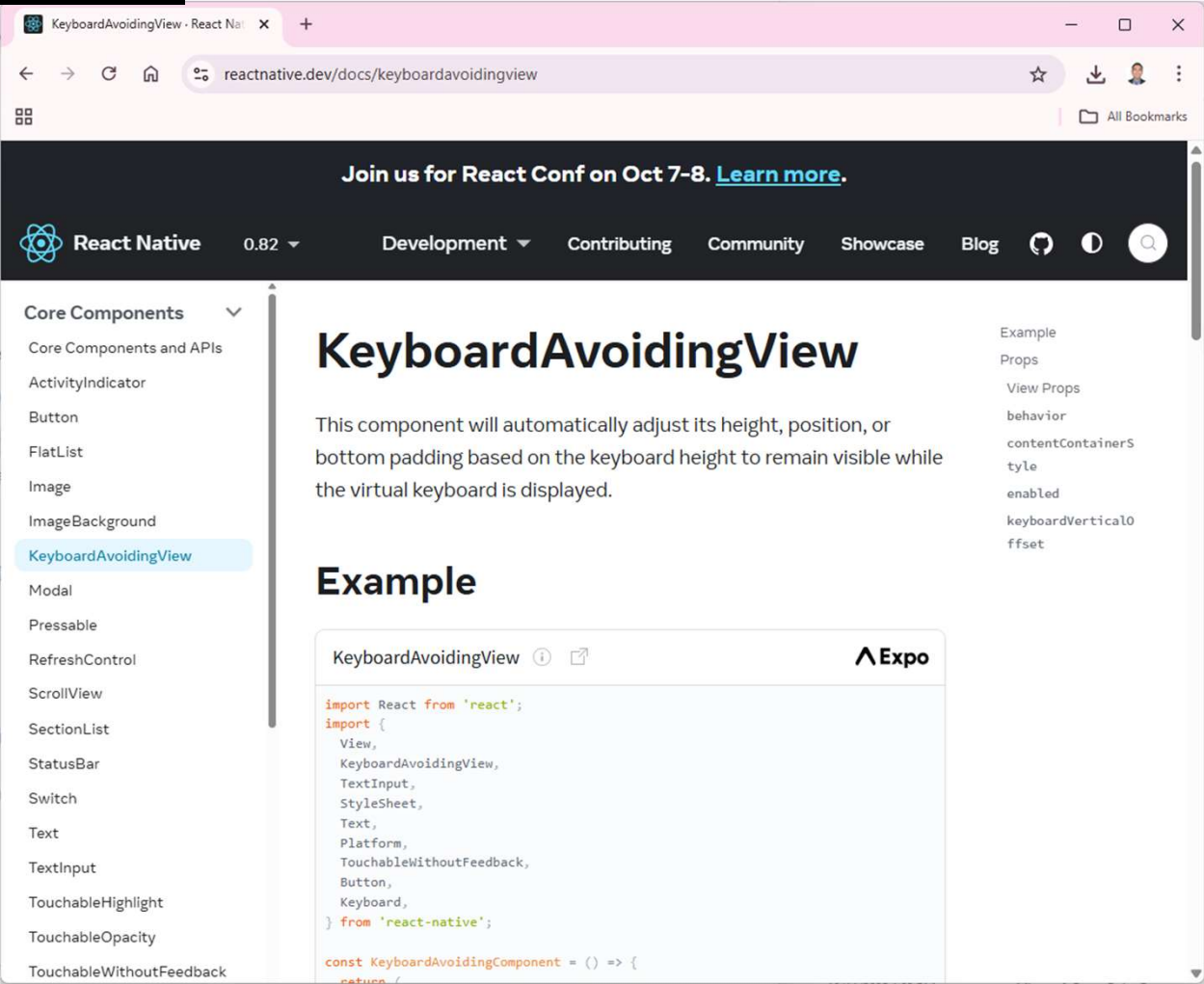
Handling input + keyboard overlap

Keyboard Avoiding View

Example – “Login with Keyboard Avoiding View”

Reference:
<https://reactnative.dev/docs/keyboardavoidingview>

This component will automatically adjust its **height**, **position**, or **bottom padding** based on the **keyboard** height to remain visible while the virtual keyboard is displayed.



Example – “Login with Keyboard Avoiding View”

```
import React, { useState } from 'react';
import {View, TextInput, Button, Text,KeyboardAvoidingView, Platform} from
'react-native';

export default function App() {
  const [email, setEmail] = useState('');
  const [pass, setPass] = useState('');

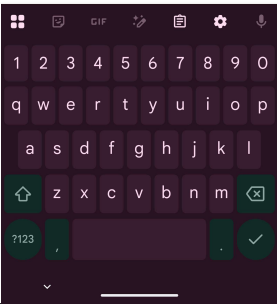
  const handleLogin = () => alert(`Email: ${email}`);

  return (
    <KeyboardAvoidingView
      behavior={Platform.OS === 'ios' ? 'padding' : 'height'}
      style={{ flex: 1, justifyContent: 'center', padding: 30 }}>
      <TextInput
        placeholder="Email"
        style={{ borderWidth: 1, padding: 10, marginBottom: 10 }}
        onChangeText={setEmail}
      />
      <TextInput
        placeholder="Password"
        secureTextEntry
        style={{ borderWidth: 1, padding: 10, marginBottom: 10 }}
        onChangeText={setPass}
      />
      <Button title="Login" onPress={handleLogin} />
    </KeyboardAvoidingView>
  );
}
```

When the keyboard appears on mobile devices, it can **cover up input fields and buttons**, making it impossible for users to see what they're typing or to access the submit button.

9:56 4G+ • • •

LOGIN



- How it works step-by-step:
- 1. User taps on a `TextInput` → Keyboard starts to appear
 - 2. `KeyboardAvoidingView` detects this → Automatically adjusts the layout
 - 3. Content shifts up → Input fields and button remain visible above the keyboard
 - 4. Keyboard dismissed → View returns to original position