



Introducing Universal RSC in Expo Router

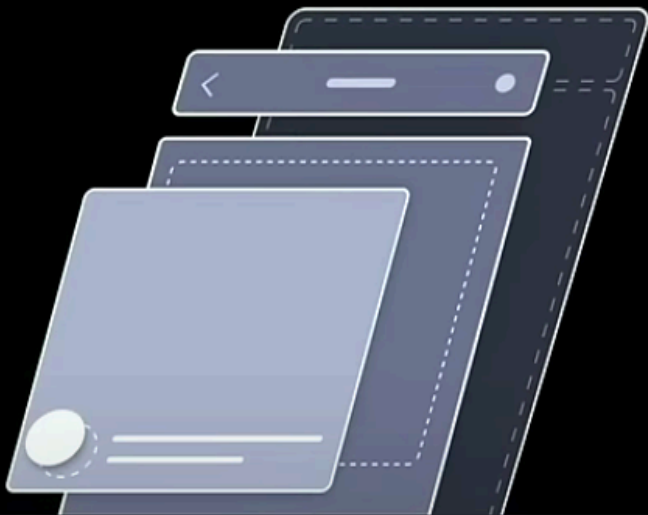


Expo Router

File System-based Routing for React Native

▼ app

- home.tsx
- ▼ (stack)
- new.tsx
- [post].tsx



This module runs on-device

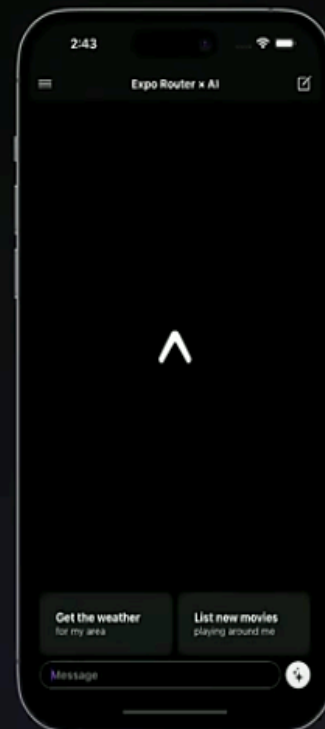
React Server Action (POST request)

"use client"

```
import { ScrollView, TextInput, View } from 'react-native';

export function Chat({ sendMessage }) {
  const [messages, setMessages] = useState([]);
  return (
    <ScrollView>
      {messages.map((message) => (
        <View key={message.id}>{message.ui}</View>
      ))}
    </ScrollView>

    <TextInput
      onSubmitEditing={async ({ nativeEvent }) => {
        // Invoke server action with client prompt
        const response = await sendMessage(nativeEvent.text);
        setMessages((messages) => [...messages, response]);
      }}
    />
  );
};
```



```
import { Text } from 'react-native';

// React Server Action
export function sendMessage(text: string) {
  "use server";
  const stream = askLLMAsync(text);
  return (
    <Suspense>
      <RecursiveText buffer={stream.getReader()} />
    </Suspense>
  )
}

// Async React Server Component
async function RecursiveText({ buffer }) {
  const { done, value } = await buffer.read();

  if (done) return null;

  const text = new TextDecoder().decode(value);

  return (
    <Text>
      {text}
      { /* Ensure stream remains open. */ }
    <Suspense>
      <RecursiveText buffer={buffer} />
    </Suspense>
  </Text>
);
}
```



Expo Router x AI

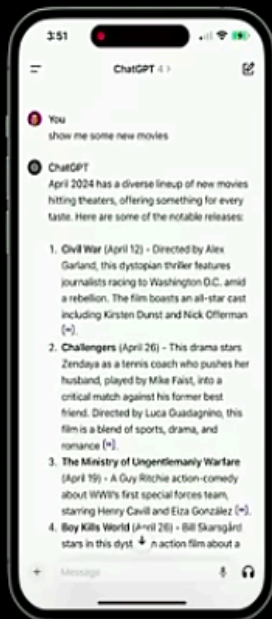


Should native apps be server rendered?

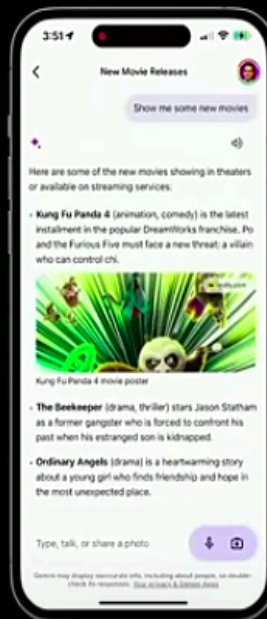


Native apps are typically not server-rendered

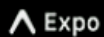
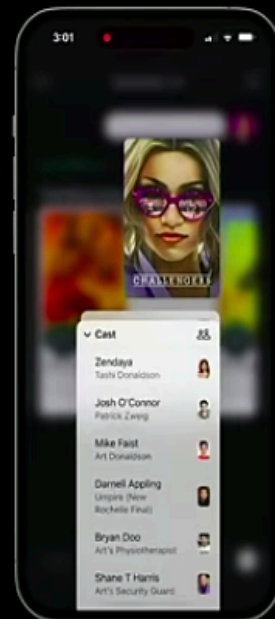
ChatGPT 4



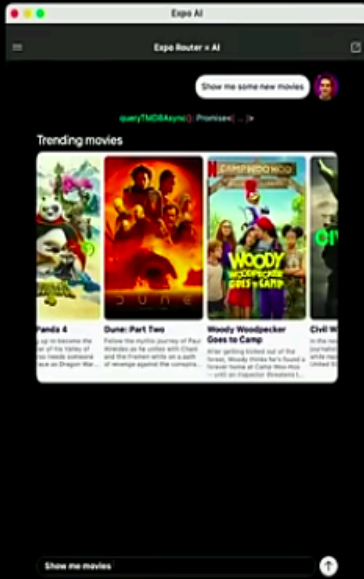
Google Gemini



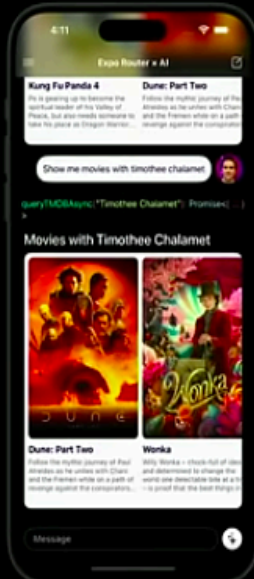
Expo Router



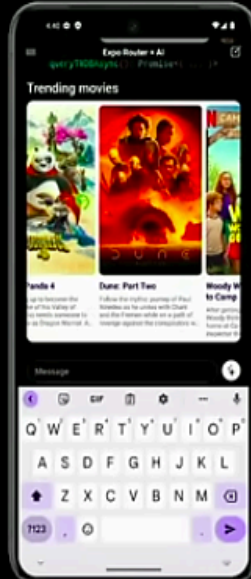
Apple Desktop



iOS



Android



Web

