

useSignInWithEmailAndPassword

Overview

The `useSignInWithEmailAndPassword` hook allows users to securely sign in with their email and password. This hook integrates with Replyke's authentication system and returns access token, refresh token, and user details upon successful registration.

Usage Example

```
import { useSignInWithEmailAndPassword } from "@replyke/react-js";

function SignInForm() {
  const signIn = useSignInWithEmailAndPassword();

  const handleSignIn = async () => {
    try {
      const result = await signIn({
        email: "user@example.com",
        password: "securePassword123",
      });

      console.log("User signed in:", result.user);
    } catch (error) {
      console.error("Sign-in failed:", error.message);
    }
  };

  return <button onClick={handleSignIn}>Sign In</button>;
}
```

Parameters & Returns

Parameters

The hook returns a function that accepts an object with the following fields:

Parameter	Type	Required	Description
email	string	Yes	The user's email address.
password	string	Yes	The user's password.

Returns

The returned function resolves with an object containing:

Property	Type	Description
accessToken	string	The access token for the authenticated user. It should be stored in memory.
refreshToken	string	The refresh token for the user session. React Native apps must store this securely, whereas React apps do not need to store it manually as it is managed via cookies.
user	object	The authenticated user's details.

Last updated on May 7, 2025