`useSignUpWithEmailAndPassword`

Overview

The `useSignUpWithEmailAndPassword` hook enables you to securely sign up a user 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 { useSignUpWithEmailAndPassword } from "@replyke/react-js";

function SignUpForm() {
  const signUp = useSignUpWithEmailAndPassword();

  const handleSignUp = async () => {
    try {
      const result = await signUp({
        email: "user@example.com",
        password: "securePassword123",
        name: "John Doe",
        username: "johndoe",
        avatar: "https://example.com/avatar.jpg",
        bio: "Just another user",
        location: { latitude: 40.7128, longitude: -74.0060 },
        birthdate: new Date("1990-01-01"),
        metadata: { favoriteColor: "blue" },
        secureMetadata: { ssn: "123-45-6789" },
      });

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

  return <button onClick={handleSignUp}>Sign Up</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.
`name`	`string`	No	The user's full name.
`username`	`string`	No	The user's chosen username.
`avatar`	`string` (URL)	No	URL to the user's profile picture.
`bio`	`string`	No	A short bio or description for the user.
`location`	`object`	No	The user's geographic location (latitude, longitude).
`birthdate`	`Date`	No	The user's birthdate.
`metadata`	`Record<string, any>`	No	Custom metadata associated with the user.
`secureMetadata`	`Record<string, any>`	No	Metadata about the user that would not be fetched when the user is referenced (e.g., in comments or entities).

Returns

The returned function resolves with an object containing:

Property	Type	Description
`accessToken`	`string`	The access token for the newly created 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

Property	Type	Description
		store it manually as it is managed via cookies.
`user`	`object`	The registered user's details.

Last updated on May 7, 2025

© Replyke 2025