Using Built-in Authentication

Replyke's built-in authentication makes it straightforward for developers to implement user sign-up, sign-in, and sign-out functionality in their applications. This is achieved through the `useAuth` hook, which provides access to the authentication context of the `AuthProvider`.

Accessing Authentication Methods

To interact with authentication features, use the `useAuth` hook. It exposes several key functions:

Sign-up

This function allows developers to create a new user account using an email and password. It also supports additional user attributes to provide a more personalized experience.

Parameters:

```
signUpWithEmailAndPassword: (props: {
  email: string;
  password: string;
  name?: string;
  username?: string;
  avatar?: string;
  bio?: string;
  location?: {
    latitude: number;
    longitude: number;
  };
  birthdate?: Date;
  metadata?: Record<string, any>;
  secureMetadata?: Record<string, any>;
}) => Promise<void>;
```

Example:

```
const { signUpWithEmailAndPassword } = useAuth();
```

```
await signUpWithEmailAndPassword({
  email: "user@example.com",
  password: "securePassword123",
  name: "John Doe",
  username: "johndoe",
  bio: "Loves coding and coffee",
});
```

Sign-in

This function handles user login using an email and password.

Parameters:

```
signInWithEmailAndPassword: (props: {
  email: string;
  password: string;
}) => Promise<void>;
```

Example:

```
const { signInWithEmailAndPassword } = useAuth();

await signInWithEmailAndPassword({
  email: "user@example.com",
  password: "securePassword123",
});
```

Sign-out

This function signs out the currently authenticated user.

Parameters:

```
signOut: () => Promise<void>;
```

Example:

```
const { signOut } = useAuth();

await signOut();
```

Change-password

This function lets an authenticated user change their password.

Parameters:

```
changePassword: (props: {
  password: string;
  newPassword: string;
}) => Promise<void>;
```

Example:

```
const { changePassword } = useAuth();

await changePassword({
  password: "12345678",
  newPassword: "abcdefgh",
});
```

Summary

By leveraging these three functions, developers can easily implement authentication workflows in their projects. Whether it’s creating new accounts, logging users in, or logging them out, Replyke’s `useAuth` hook simplifies the process, enabling a seamless integration into your application.

Last updated on May 7, 2025