

Understanding the User Object and useUser Hook

Replyke provides a simple way to access and manage user information using the `useUser` hook. This hook is automatically available as part of the `AuthProvider`, which is added under the hood of `ReplykeProvider`, so there is no need to wrap your app with an additional context provider.

The useUser Hook

The `useUser` hook provides access to the user context and exposes two key properties:

1. `user`: An object containing information about the authenticated user.
2. `updateUser`: A function to update the user's details.

Updating User Information

The `updateUser` function allows developers to update user details. It accepts an object with the following properties:

```
export type UpdateUserParams = {
  name?: string | null; // Update the user's name
  username?: string | null; // Update the user's username
  avatar?: string | null; // Update the user's avatar
  bio?: string; // Update the user's bio
  birthdate?: Date | null; // Update the user's birthdate
  location?: {
    latitude: number;
    longitude: number;
  } | null; // Update the user's location
  metadata?: Record<string, any>; // Update custom metadata
  secureMetadata: Record<string, any>; // Update custom secure metadata
};
```

Example Usage of updateUser

Here are examples of how to use the `updateUser` function to modify user details:

Updating Basic User Information

```
const { updateUser } = useUser();

const updateName = async () => {
  try {
    await updateUser({
      name: "John Doe",
      username: "johndoe123",
    });
    console.log("User updated successfully!");
  } catch (error) {
    console.error("Error updating user:", error);
  }
};

updateName();
```

Updating User Metadata

```
const { updateUser } = useUser();

const updateMetadata = async () => {
  try {
    await updateUser({
      metadata: {
        favoriteColor: "blue",
        interests: ["coding", "gaming"],
      },
    });
    console.log("Metadata updated successfully!");
  } catch (error) {
    console.error("Error updating metadata:", error);
  }
};

updateMetadata();
```

Updating User Location

```
const { updateUser } = useUser();

const updateLocation = async () => {
  try {
```

```
    await updateUser({
      location: {
        type: "Point",
        coordinates: [-73.935242, 40.73061], // New York City coordinates
      },
    });
    console.log("Location updated successfully!");
  } catch (error) {
    console.error("Error updating location:", error);
  }
};

updateLocation();
```

Summary

The `useUser` hook is a powerful tool for accessing and managing user data in Replyke. By understanding the structure of the user object and how to use `updateUser`, developers can create seamless and personalized user experiences. If you have any additional metadata or user-specific details to handle, the `metadata` property provides ample flexibility.

Last updated on May 7, 2025