Authentication & Session Management

Table des matières

1. Overview

Trips & Roads uses a multi-layered authentication system. The core is powered by the **CakePHP Authentication Component**, managing user sessions, while external login is handled via **Google OAuth2**.

2. Authentication Mechanisms

A. Standard Session Authentication

Most of the application relies on stateful sessions.

- **Login Logic:** The login() action verifies credentials against the Users table using a secure password hasher.
- **Persistence:** Upon successful login, the user's identity is stored in the session.
- **Unauthenticated Access:** Specific actions (like index, view, and accessibility) are explicitly allowed for non-logged-in users via addUnauthenticatedActions().

B. Google OAuth2 Integration

The platform allows users to sign in using their Google account.

- **Provider:** Uses the League\OAuth2\Client\Provider\Google library.
- **Flow:** 1. The user is redirected to Google's consent screen (loginGoogle). 2. After approval, Google redirects back to callbackGoogle. 3. The system checks if the email exists in our database; if not, a new user is created automatically.

3. Security & Authorization

A. Password Policy

Passwords are never stored in plain text. They are automatically hashed using the **bcrypt** algorithm through CakePHP's DefaultPasswordHasher.

B. Access Control (RBAC)

Access is restricted at the Controller level:

- **Global Protection:** The initialize() method in AppController requires authentication by default.
- **Manual Overrides:** Controllers like FriendshipsController and MessagesController verify that the userId in the request matches the identity of the logged-in user to prevent vulnerabilities.

C. Cookie-Based Preferences

Beyond authentication, we use secure cookies to store user accessibility preferences (e.g., Dark Mode, Daltonism settings) even for unauthenticated users.

4. Developer Commands

To clear all active sessions during development, you can use:

```
Bash

$ rm -rf tmp/sessions/*
```