WeImg Server

This is the README.md from github project https://github.com/lzwjava/weimg-server.

---

weimg-server

WeImg is your ultimate destination for discovering the most hilarious memes, adorable pets in sweaters, mind-blowing science facts, hidden video game Easter eggs, and everything else that makes the internet so entertaining. Get ready to add a whole new level of fun to your phone!

Welcome to weimg-server! This repository contains the backend components for powering a dynamic web application. Below is a brief overview of the directory structure and key components of the project:

Directories:

- **cache**: Contains cached files used for optimizing performance.
- **config**: Stores configuration files for various aspects of the application such as database settings, routes, and constants.
- **controllers**: Houses PHP controllers responsible for handling incoming requests and generating responses.
- **core**: Contains core PHP classes and controllers fundamental to the application's functionality.
- **helpers**: Stores PHP helper functions and utilities used throughout the application.
- **hooks**: Placeholder directory for implementing custom hooks and callbacks.
- **id**: [No description provided]
- **language**: Contains language files for internationalization support, currently only supporting English.
- **libraries**: Stores custom PHP libraries and third-party dependencies used in the application.
- **logs**: Placeholder directory for storing application logs.
- **models**: Houses PHP models representing data entities and interacting with the database.
- **third_party**: Placeholder directory for third-party libraries or modules.

Files:

- **index.html**: Default landing page for the server project.
- **test.php**: A PHP script for testing purposes.
- **welcome_message.php**: PHP script generating a welcome message for the application's homepage.

How to Use:

1. Ensure that PHP is installed on your server environment.

2. Configure the settings in the `config` directory, especially `config.php` and `database.php`, according to your environment.
3. Utilize the controllers in the `controllers` directory to define application logic and handle HTTP requests.
4. Interact with the database using the models defined in the `models` directory.
5. Customize and extend the functionality of the application by adding new controllers, models, libraries, and helpers as needed.
6. Refer to the `views` directory for HTML templates and error pages.

Feel free to explore the project further and contribute enhancements or report any issues you encounter. Happy coding!