

## DEVELOPER NOTES:

This document provides a **simplified explanation** of the UserController class in **Laravel**, focusing on its **main flow and logic** rather than deep framework internals.

It helps team to understand **how data is handled, validated, stored, and displayed** in a RESTful API.

### UserController – Workflow Summary

#### 1. Incoming Request

A client sends an API request to create or retrieve users.

#### 2. Validation Phase

Input data is validated to ensure correct format, unique email, and secure password rules.

#### 3. Database Operation

- o For store: a new user record is created with a hashed password.
- o For index: users are retrieved with optional search, sort, and pagination.

#### 4. Transformation Phase

User data is formatted before being returned.

Additional attributes (like can\_edit or orders\_count) are added for clarity.

#### 5. Authorization Check

The system verifies if the current user has permission to edit or view specific data (based on role: admin, manager, or self).

#### 6. Response Generation

The controller returns a clean, structured JSON response containing the requested or newly created user information.

#### 7. Optional Notification

A notification or welcome email can be sent after successful user creation.

## **Model – Summary**

### **1. User Model (User.php)**

- This model represents the users table.
- Table columns: id, name, email, password, role, active, remember\_token, created\_at, updated\_at.

### **2. Order Model (Order.php)**

- This model represents the orders table.
  - Table columns: id, user\_id, created\_at, updated\_at.
- 

## **API – Summary :**

- These routes follow RESTful API conventions.
- All responses are JSON formatted.
- The /user route is protected and requires a valid Sanctum token for authentication.
- Use proper request validation in the controller to ensure input integrity.

## **ORIGINAL URL :**

<https://github.com/gumuruuhsspj/php-code-test>

## **DOWNLOAD LINK (ZIP):**

<https://github.com/gumuruuhsspj/php-code-test/archive/refs/heads/main.zip>

## **NOTES :**

The email sending function in this system is currently in **simulation mode (inactive)**.

Email implementation can be reactivated once the final configuration for one of the following options is determined:

1. SMTP using the company's domain email
2. Integration with a third-party mailing service (e.g., Mailgun, SendGrid, etc.)
3. Alternative methods as defined by the system policy