Project Installation Guide

This guide provides detailed instructions to set up and start the project.

---

1. Initial Setup

Extract the RAR file with the project to your development or production directory and then follow the steps below:

> ***Note:*** *The documentation folder is not required for the application to function on the server. Including this item is optional.*

1.1. Modifying the `.htaccess` File

Ensure that the server is configured to use PHP version 7.4. Below is an example configuration that can be inserted into the `.htaccess` file located at the root of the project:

```
# Define PHP version 7.4 for this project on your project (server: hostinger)
# Check with your server provider for the method to declare the PHP version used
<FilesMatch "\.(php4|php5|php3|php2|php|phtml)$">
    SetHandler application/x-lsphp74
</FilesMatch>

<Files "*">
    Order Deny,Allow
    Deny from all
</Files>

<FilesMatch "\.(eot|otf|ttf|woff|woff2|svg)$">
    Order Allow,Deny
    Allow from all
</FilesMatch>

<FilesMatch "\.(html|css|js|jpg|jpeg|png|gif|svg|webp|ico|bmp|tiff)$">
    Order Allow,Deny
    Allow from all
</FilesMatch>

<Files "index.php">
    Order Allow,Deny
    Allow from all
</Files>

Options -Indexes
RewriteEngine On

# Used in production
RewriteBase /

## Used in development
# RewriteBase /l5-test/
```

```
RewriteCond %{REQUEST_FILENAME} -f
RewriteRule ^ - [L]

RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]

RewriteRule ^front-end/view/ - [L]
RewriteRule ^(.*)$ index.php [QSA,L]
```

> ***Note 1:*** *Confirm with the server administrator if the PHP version configuration in* `FilesMatch` *is compatible.*
>
> ***Note 2:*** *In a development environment, there is no need to use* `FilesMatch` *for the PHP version, as IDEs generally handle this.*
>
> ***Note 3:*** *The* `Files` *and* `FileMatch` *configurations were necessary for the security of the online application; in a local development environment, these configurations are not needed and should not be applied.*
>
> ***Note 4:*** *Replace* `RewriteBase` *according to your project's directory.*
>
> *For any additional questions, refer to the htaccess documentation:*
> *https://httpd.apache.org/docs/2.4/howto/htaccess.html*

1.2. Modifying the `util.js` File

Update the project's base URL in the `util.js` file, located in `front-end/view/assets/js/utils`:

```
window.globalSiteUrl = 'http://localhost/l5-test/api/';
```

Replace the values according to the development or production environment.

---

2. Database

2.1. Creating the Database

1. Create a database with the desired name.
2. Import the SQL file available in the `documentation` folder to set up the tables and initial data.

> ***Note:*** *After importing the Dump (file* `empty-db-dump.sql`*), you also have the option to import 1000 fake records, available in the same directory in the file* `fake-data-to-db.sql`*.*
> *These data will be used for better testing of the API.*

2.2. Database Structure

Tables

1. **`api_logs`**

Stores records of API calls made.

Field	Type	Attributes	Description
`id`	INT(20)	NOT NULL, PRIMARY KEY, AUTO_INCREMENT	Unique identifier.
`register_date`	DATETIME	NOT NULL	Date and time of the record.
`request_method`	VARCHAR(10)	NOT NULL	HTTP request method (e.g., GET, POST).
`endpoint`	VARCHAR(255)	NOT NULL	Accessed endpoint.
`response_code`	INT(3)	NOT NULL	HTTP response code.
`user_ip`	VARCHAR(50)	NOT NULL	User's IP address.

2. `users`

Stores information about users who access the restricted API area.

Field	Type	Attributes	Description
`id`	INT(5)	NOT NULL, PRIMARY KEY, AUTO_INCREMENT	Unique identifier.
`name`	VARCHAR(150)	NOT NULL	User's name.
`api_key`	CHAR(20)	NOT NULL, UNIQUE KEY	Unique authentication key.

Indexes

- **Table `api_logs`:**
  - Primary Key: `id`
- **Table `users`:**
  - Primary Key: `id`
  - Unique Key: `api_key`

Special Configurations

- All tables use:
  - Charset: `utf8mb4`
  - Collation: `utf8mb4_general_ci`
  - Engine: **InnoDB**
- **AUTO_INCREMENT**:
  - `users`: Starts at `2`.

---

3. Configuring the `Config.php` File

Edit the `Config.php` file, located in the `system` folder, with the database and URL settings:

3.1. Database Settings

Adjust the settings according to the database access credentials created in the previous step.

```
/**
 * Database connection settings.
 * These constants define the necessary credentials and configuration for connecting to the database.
 */
define('DB_HOST', 'localhost');
define('DB_PORT', '3306');
define('DB_NAME', 'l5transactions');
define('DB_USERNAME', 'root');
define('DB_PASSCODE', '');
```

3.2. System URLs

Update the URLs according to your project's directory.

```
/**
 * Base URLs for the production and development environments.
 * These constants define the base URLs for different environments to ensure proper routing.
 */
define('PRODUCTION_URL', 'https://your-production-site-url.here');
define('DEVELOPMENT_URL', 'http://localhost/l5-test'); // Adjust for your development directory structure

/**
 * URLs for the site in different environments.
 * These constants define the base URLs for site routing depending on the environment.
 */
define('URL_PRODUCTION', '/');
define('URL_DEVELOPMENT', '/l5-test/'); // Adjust for your development directory structure
```

---

4. Integration with The Movie Database API

To display movie posters, you need to register for The Movie Database API.
Registration is free and can be done at the link:

https://api.themoviedb.org/

After registering, insert the generated key in the `secureConfig.php` file located in the `system` folder.

```
define('FILM_IMAGE_API_KEY', 'Insert your API key here');
```

> ***Note 1:*** *For sensitive information, such as passwords and access data, it is recommended to use `.env` files or similar techniques.*
> *In this project, the use of external tools was limited by the requirements, so to avoid potential non-compliance, I chose to use a separate file, called `secureConfig.php`, just to avoid versioning the API key on GitHub.*

> ***Note 2:*** *It is necessary to keep this global variable for the app to function. If you choose not to create a key, leave the variable empty.*

---

5. Finalization

After following all the steps, the project will be set up and ready to use. Run tests to ensure everything is functioning correctly. ```