Money Lover API

Thành viên nhóm:

Nguyễn Đức Thịnh

Nguyễn Trọng Minh Dũng

Lê Anh Sơn

<https://github.com/thinhnd580/MoneyLoverAPI>

**Mục lục :**

Contents

[Phần I : Overview 2](#_Toc449645258)

[1. Giới thiệu 2](#_Toc449645259)

[2. Môi trường phát triển 2](#_Toc449645260)

[3. Reference 2](#_Toc449645261)

[Phần II : Setup Enviroment 3](#_Toc449645262)

[1. Enviroment 3](#_Toc449645263)

[2. Database Import 3](#_Toc449645264)

[Phần III : API Specification 3](#_Toc449645265)

[1. Overview 3](#_Toc449645266)

[2. APIs 3](#_Toc449645267)

[I. Login API 3](#_Toc449645268)

[II. Sign Up API 4](#_Toc449645269)

[III. Get User Info 4](#_Toc449645270)

[IV. Create Transaction 5](#_Toc449645271)

[V. Get All Transaction 5](#_Toc449645272)

[Phần IV : Project Organization 6](#_Toc449645273)

[1. File Organize 6](#_Toc449645274)

[2. Coding 7](#_Toc449645275)

[Phần IV : Database 8](#_Toc449645276)

[1. Mapping 8](#_Toc449645277)

[2. Entity in database 9](#_Toc449645278)

[I. User 9](#_Toc449645279)

[II. Indentities 9](#_Toc449645280)

[III. Transaction 10](#_Toc449645281)

[IV. Category 10](#_Toc449645282)

[Phần V : Security 10](#_Toc449645283)

[1. Authorization 10](#_Toc449645284)

[2. Token base 10](#_Toc449645285)

[3. Password hash 11](#_Toc449645286)

[Phần VI: Test 11](#_Toc449645287)

[1. Trước khi test 11](#_Toc449645288)

[2. Unit test 11](#_Toc449645289)

# Phần I : Overview

1. Giới thiệu

* Money Lover API được xây dựng dựa trên php với framework Symfony 3.x, cung cấp các hàm api cho các chức năng : login, sign up, get user info, create transaction, get all transaction.
* Api được thiết kế để cho việc sử dụng ứng dụng trên di động.
* Sản phầm do nhóm làm gồm có 3 thành viên : Nguyễn Đức Thịnh, Nguyễn Trọng Minh Dũng, Lê Anh Sơn.
* **Sản phẩm của nhóm tại đường link :** <https://github.com/thinhnd580/MoneyLoverAPI>

1. Môi trường phát triển

* Framework : Symfony 3.x
* OS : Windows
* Server : Xampp 7
* Testing: Phpunit

1. Reference

* Symfony : <http://symfony.com>
* Xampp : <https://www.apachefriends.org/download.html>
* Phpunit : <https://phpunit.de/>
* MariaDB : <https://mariadb.org/>

# Phần II : Setup Enviroment

1. Enviroment

* Setup Xampp 7
* Pull source code từ github : <https://github.com/thinhnd580/MoneyLoverAPI>
* Add php variable to windows path
* Chạy thử bằng các công cụ hỗ trợ post như postman

1. Database Import

* Import file data.sql vào cơ sở dữ liệu để chạy

# Phần III : API Specification

1. Overview

* API được thiết kế chỉ hoạt động với phương thức POST với 1 url duy nhất: <http://localhost/MoneyLoverAPI/web/app_dev.php/api>
* API sư dụng kiểu dữ liệu Json để response
* Các phương thức khác POST sẽ không được đáp ứng
* Khi sử dụng phương thức nào, ta gửi lên với parameter “method” và gắn giá trị của phương thức cần sử dụng rồi gửi lên kèm theo các parameter khác.
* Ví dụ : Phương thức Get User Info

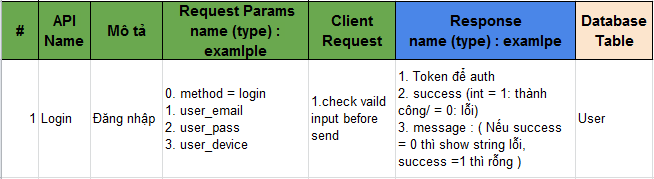


* Success Request : Status Code = 200
* Fail Request : Status Code = 4xx hoặc 5xx

1. APIs
2. Login API

Request

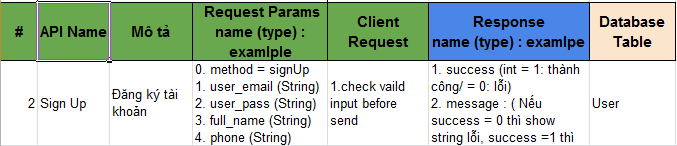
|  |  |
| --- | --- |
| Method | URL |
| POST | <http://localhost/MoneyLoverAPI/web/app_dev.php/api> |



1. Sign Up API

Request

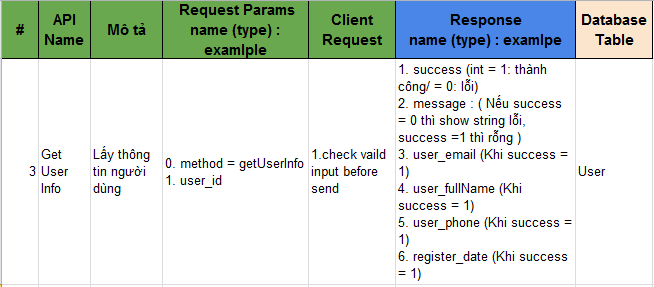
|  |  |
| --- | --- |
| Method | URL |
| POST | <http://localhost/MoneyLoverAPI/web/app_dev.php/api> |



1. Get User Info

Request

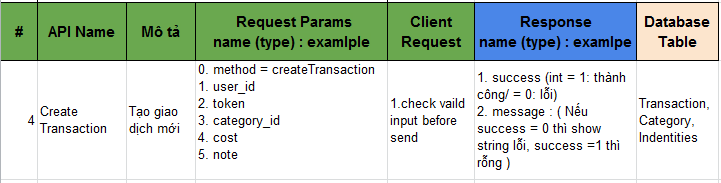
|  |  |
| --- | --- |
| Method | URL |
| POST | <http://localhost/MoneyLoverAPI/web/app_dev.php/api> |



1. Create Transaction

Request

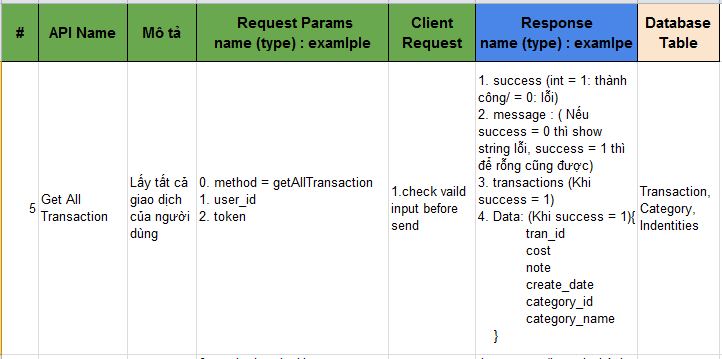
|  |  |
| --- | --- |
| Method | URL |
| POST | <http://localhost/MoneyLoverAPI/web/app_dev.php/api> |



1. Get All Transaction

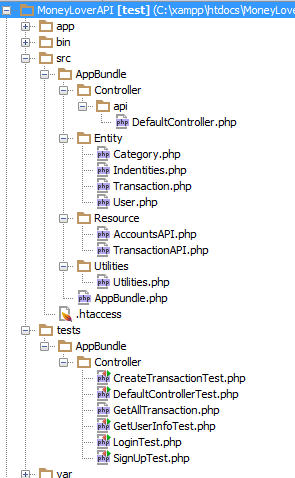
Request

|  |  |
| --- | --- |
| Method | URL |
| POST | <http://localhost/MoneyLoverAPI/web/app_dev.php/api> |



# Phần IV : Project Organization

1. File Organize



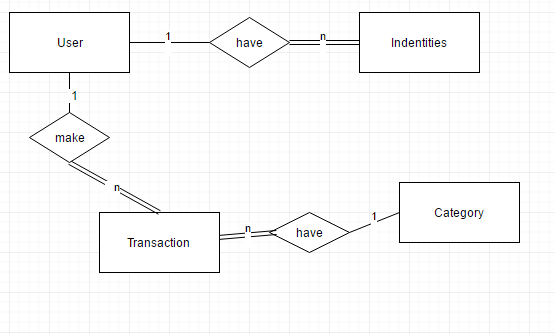
* Src : All class for api
* Tests : All unit test

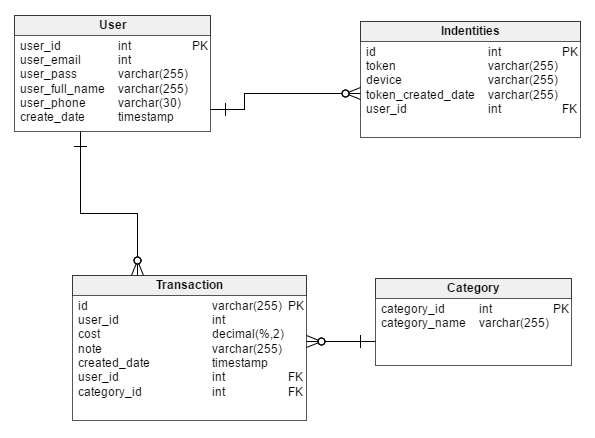
1. Coding

* DefaultController: Controller để điều hướng và điểu khiển các request
* Các entity: User, Indentities, Transaction, Category
* AccoutAPI : cung cấp các hàm xử lý các request liên quan đến acount gồm Login, Signup và Authorization . Các hàm này trả về json response là kết quả của request
* Utilities: Hỗ trợ các hàm static hỗ trợ việc tái sử dụng code
* Các file trong thư mục tests gồm các unit test

# Phần IV : Database

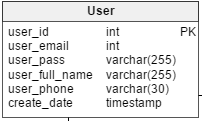
1. Mapping



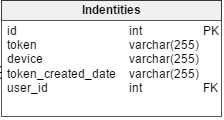


* User - Indentities : one to many
* User – Transaction: one to many
* Transaction – category: many to one

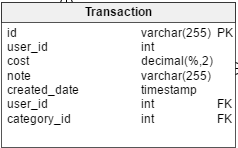
1. Entity in database
2. User



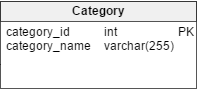
1. Indentities



1. Transaction



1. Category



# Phần V : Security

1. Authorization

* Người dùng sau khi đăng nhập sẽ được cấp một token và token đó sẽ được lưu trong database
* Sau lần đăng nhập đó, mỗi request gửi lên sẽ phải gửi kèm theo token để xác minh. Nếu token sai hoặc hết hạn sẽ không thể thực hiện được request.

1. Token base

* Token được mã hóa bằng hàm md5 với các parameter truyền vào là user\_email, user\_pass và device\_id của thiết bị sử dụng để đăng nhập. Ở đây Api được thiết kế để cho việc sử dụng ứng dụng trên di động.
* Sau một thời gian nhất định, token sẽ hết hạn, người dùng bắt buộc phải đăng nhập lại

1. Password hash

* Mật khẩu của người dùng được bảo mật và lưu trong cơ sở dữ liệu dưới dạng đã được mã hóa.

# Phần VI: Test

1. Trước khi test

* Bật cửa số cmd với đường dẫn là thư mục đến project
* Chạy 2 câu lệnh sau trước khi test:
* $ php bin/console doctrine:database:drop --force
* $ php bin/console doctrine:database:create
* Import cơ sở dữ liệu data.sql
* Gõ lệnh : phpunit

1. Unit test

* Tổng các hàm unit test : **21**
* Tổng assertion : **139**
* Passed : **100%**

