Technical Design Document

Prepared for **Fun Academy**
by Germán González Rodríguez

SUMMARY

Along this document I'll try to explain the technical decisions made for developing the Fun Academy backend developer test. It is also documented all the api endpoints. Built with Lumen. All the Part 1 and 2 features were developed. The extra features were unimplemented due to lack of time.

INSTALLATION

Server Requirements

As part of Lumen requirements:

- PHP >= 5.6.4
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension

For the proposed solution it is also required:

- Mysql
- Composer

All of these requirements are satisfied by the Laravel Homestead, so it was used as local development environment.

Deployment

For deploying the app. In a terminal window:

1. Execute **composer install** in the provided source root directory to install the dependencies.
2. Create a new database in Mysql.
3. Rename .env.example file to .env and update the database variables.
4. Migrate the database. Execute **php artisan migrate** in the root directory.
5. Seed the database using **php artisan db:seed** command.

Testing

For your evaluation, a beta environment was set up at:
http://funacademy.gergonzalez.com/
A postman export file with all the api requests is provided to facilitate the testing using this chrome extension.

DEVELOPEMENT

Coding Style

Standards defined in the PSR-0, PSR-1, PSR-2 and PSR-4 documents were followed.

The API

A RESTful API was implemented. Due to Lumen is a light and quick version of Laravel is a perfect fit for developing web services.

App Directory Structure

The structure used is the default Laravel/Lumen structure. But 3 more folders were added. One named **app/Services** stores the custom guard implemented (Check **Auth** for more information). The second one is named **packages** and it stores my local package Fractal. The last one is **app/Http/Transformers** and it stores the data transformers (Check **Transformers**).

Database

Mysql was used, no other technology was considered because it is a skill requirement for the job offer.
For designing the database, due to the requirement of different kind of users and after considering different solutions, I decided to implement **Polymorphic relationships** for the association between users table and admins, website_users, providers and retailers. Even with its disadvantages, this solution offers a clean, readable and structured database. It also makes the database easily extensible for future user types updates. An intermediate table was created to handle the relationship between providers and retailers.
The simplified EER diagram of the implemented Database structure:

Users
Id (PRIMARY)
email (UNIQUE)
userable_id
userable_type
…

Orders
id (PRIMARY)
user_id (FK users)
product_id(FK products)
…

Products
id (PRIMARY)
name
…

Provider-Retailer
id (PRIMARY)
provider_id (FK providers)
retailer_id (FK retailers)
accepted

Admin
id (PRIMARY)
name

Website Users
id (PRIMARY)
name
…

Providers
id (PRIMARY)
company_name
…

Retailers
id (PRIMARY)
name
…

Controllers
9 controllers were implemented. These controllers are in charge of handling the logic of related requests in a single class.
The customers controllers (**WebsiteUserController, ProviderController, RetailerController, AdminController**) handle the creation and update of those users, adding providers, accept retailers, etc.
The **UserController** manages the users information flow and the account activation.
The **OrderController** is in charge of store the orders, update the status and delete them and the **ProductController** show all stored products.
Finally, **AuthController** responds to the login and token refresh requests.

Models
7 models were implemented. Each table has one Model and it allows to easily interact with that table.
They implement methods to handle the relationships between the tables and some helpers, for example **User** Model has some user type checks, like **isAdmin**, or the **Retailer** Model, one to calculate the amount of an order. About this method, **orderTotalAmount**, the method could fail if you add a provider and you already ordered products, but due to the scope of the test I decided to don't implement a more complex logic to handle this.

Auth
For the user authorization is used JSON Web Tokens (JWT) conforming to RFC 7519. Lumen implements out of the box token authorization, but it is a really simple implementation and lacks some needed features, like token expiration. There are well tested packages to implement JWT on Lumen, like https://github.com/tymondesigns/jwt-auth, but it is still on beta, so its use is not recommended in any production environment.
Because of that, I implemented a very simple custom **JWTGuard**, you can check it at app\Services\Auth folder. It implements the test required features but there is no token blacklisting or token invalidation.

Transformers
For presenting the data output, I used data transformers. Think of this as a view layer for JSON. I created a package (**Gergonzalez\Fractal**) that is a wrapper of **League Fractal package** (http://fractal.thephpleague.com).
Then, I implemented 8 data transformers that are in charge of presenting the data before sending it using JSON.

Exceptions
I'm intercepting the most common exceptions and responding with custom json error messages. Check app/Exception/Handler for more information.

Routes
The api endpoints.

Public:

HTTP METHOD	URL
POST	/login
POST	/refresh
POST	/website-users
POST	/providers
POST	/retailers

Authenticated:
email: admin@gergonzalez.com
password: admin

HTTP METHOD	URL
GET	/users
GET	/users/{id}
POST	/users/activate/{id}
PATCH	/website-users/{user_id}
PATCH	/providers/{user_id}
POST	/providers/{user_id}/discount
POST	/providers/{provider_user_id}/accept-retailer/{retailer_user_id}
PATCH	/retailers/{user_id}
POST	/retailers/{retailer_user_id}/add-provider/{provider_user_id}
DELETE	/retailers/{retailer_user_id}/remove-provider/{provider_user_id}
GET	/products
GET	/orders
POST	/orders
PATCH	orders/{order_id}
DELETE	orders/{order_id}

NOTES:
For the endpoints that require authentication, you can add the token in the header (**Key**: *Authorization*, **Value:** *Bearer eyJ0eXAiOiJKV1),* as a query *(/?token=eyJ0eXAiOiJKV1)* or as a parameter for post methods (**Key**: *token* **,** **Value:** *eyJ0eXAiOiJKV1*).

Reference
POST /Login
Returns the authorization token.

PARAMETERS
email: *required*
password: *required*

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/login

EXAMPLE HEADER
```
Authorization
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJGdW5BY2FkZW15VGVzd
CIsImlhdCI6MTUwMTQzNDk1OCwiZXhwIjoxNTAxNDM1MDE4LCJzdWIiOjF9.dstMUQ
3wfOvc8rsaYIkxOB+vNxcjkjO44pwJmJ48Z0Y=
```

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "data": {
        "token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJGdW5BY2FkZW15VGVz
dCIsImlhdCI6MTUwMTQzNDk1OCwiZXhwIjoxNTAxNDM1MDE4LCJzdWIiOjF9.dstMU
Q3wfOvc8rsaYIkxOB+vNxcjkjO44pwJmJ48Z0Y=",
        "user": {
            "id": 1,
            "email": "admin@gergonzalez.com",
            "type": "Admin",
            "activated": 1,
            "createdAt": "Sat, Jul 29, 2017 7:59 PM",
            "info": {
                "name": "Admin"
            }
        }
    }
}
```

EXAMPLE ERROR RESPONSE
```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "Incorrect Credentials",
        "token": []
    }
}
```

POST /refresh
Refresh attached token

PARAMETERS

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/refresh

EXAMPLE HEADER
```
Authorization
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJGdW5BY2FkZW15VGVzd
CIsImlhdCI6MTUwMTQzNDk1OCwiZXhwIjoxNTAxNDM1MDE4LCJzdWIiOjF9.dstMUQ
3wfOvc8rsaYIkxOB+vNxcjkjO44pwJmJ48Z0Y=
```

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "data": {
        "token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJGdW5BY2FkZW15VGVz
dCIsImlhdCI6MTUwMTQ4Nzc2NiwiZXhwIjoxNTAxNDg3ODI2LCJzdWIiOjF9.jEK3s
pDEtAPA7u8Gxw3dp3sVUUojY2RcnFUFAFK7rjw=",
        "user": {
            "id": 1,
            "email": "admin@gergonzalez.com",
            "type": "Admin",
            "activated": 1,
            "createdAt": "Sat, Jul 29, 2017 7:59 PM",
            "info": {
                "name": "Admin"
            }
        }
    }
}
```

EXAMPLE ERROR RESPONSE
```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "Token error",
        "token": []
    }
}
```

POST /website-users

Register a website user.

PARAMETERS

email(required)
password(required)
name(required)
mobilePhone(required)
createdAt(required)

EXAMPLE REQUEST

POST http://funacademy.gergonzalez.com/website-users

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 2,
    "email": "ger@gergonzalez.com",
    "type": "Website User",
    "activated": 0,
    "createdAt": "Sat, Jul 29, 2017 8:04 PM",
    "info": {
        "name": "Ringo Starr",
        "mobilePhone": "616060184"
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 422 UNPROCESSABLE ENTITY
{
    "error": {
        "message": "The given data failed to pass validation.",
        "data": {
            "email": [
                "The email has already been taken."
            ]
        }
    }
}
```

POST /providers
Register a provider.

PARAMETERS
email(required)
password(required)
companyname(required)
phone(required)
IBAN(required)
companyDescription
createdAt(required)

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/providers

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "id": 3,
    "email": "ger1@gergonzalez.com",
    "type": "Provider",
    "activated": 0,
    "createdAt": "Sat, Jul 29, 2017 8:06 PM",
    "info": {
        "companyname": "John and Yoko",
        "phone": "616060184",
        "IBAN": "123456789019284",
        "companyDescription": "",
        "discount": 0
    }
}
```

EXAMPLE ERROR RESPONSE
```
HTTP 1.1 422 UNPROCESSABLE ENTITY
{
    "error": {
        "message": "The given data failed to pass validation.",
        "data": {
           "IBAN": [
                "The i b a n field is required."
            ]
        }
    }
}
```

POST /retailers
Register a retailer

PARAMETERS
email(required)
password(required)
name(required)
retailerResponsibleName(string,required)
phone
mobilePhone(required)
address(required)
IBAN(required)
provider: provider_user_id
createdAt(required)

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/retailers

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "id": 4,
    "email": "gers34ddssad4@gergonzalez.com",
    "type": "Retailer",
    "activated": 0,
    "createdAt": "Sat, Jul 29, 2017 8:09 PM",
    "info": {
        "name": "Paul McCartney",
        "retailerResponsibleName": "The Beatles",
        "phone": "123456789",
        "mobilePhone": "616060184",
        "address": "Strawberry Fields Square",
        "IBAN": "123456789019284",
        "providers": {
            "data": [
                {
                    "id": 3,
                    "email": "ger1@gergonzalez.com",
                    "activated": 0,
                    "createdAt": "Jul 29, 2017",
                    "type": "Provider",
                    "accepted": 0,
                    "data": {
                        "companyname": "John and Yoko",
                        "phone": "616060184",
                        "IBAN": "123456789019284",
                        "companyDescription": "",
                        "discount": 0
                    }
                }
            ]
        }
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "Incorrect Credentials",
        "token": []
    }
}
```

GET /users
Get all registered users. Authorization required. Only Admin.

PARAMETERS
No parameters

EXAMPLE REQUEST
GET http://funacademy.gergonzalez.com/users

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "data": [
        {
            "id": 1,
            "email": "admin@gergonzalez.com",
            "type": "Admin",
            "activated": 1,
            "createdAt": "Sat, Jul 29, 2017 7:59 PM",
            "info": {
                "name": "Admin"
            }
        },
        {
            "id": 2,
            "email": "ger@gergonzalez.com",
            "type": "Website User",
            "activated": 0,
            "createdAt": "Sat, Jul 29, 2017 8:04 PM",
            "info": {
                "name": "Ringo Starr",
                "mobilePhone": "616060184"
            }
        },
        {
            "id": 3,
            "email": "ger1@gergonzalez.com",
            "type": "Provider",
            "activated": 0,
            "createdAt": "Sat, Jul 29, 2017 8:06 PM",
            "info": {
                "companyname": "John and Yoko",
                "phone": "616060184",
                "IBAN": "123456789019284",
                "companyDescription": "",
                "discount": 0
            }
        },
        {
            "id": 4,
            "email": "gers34ddssad4@gergonzalez.com",
            "type": "Retailer",
            "activated": 0,
            "createdAt": "Sat, Jul 29, 2017 8:09 PM",
            "info": {
                "name": "Paul McCartney",
                "retailerResponsibleName": "The Beatles",
                "phone": "123456789",
                "mobilePhone": "616060184",
                "address": "Strawberry Fields Square",
                "IBAN": "123456789019284",
                "providers": {
                    "data": [
                        {
                            "id": 3,
                            "email": "ger1@gergonzalez.com",
                            "activated": 0,
                            "createdAt": "Jul 29, 2017",
                            "type": "Provider",
                            "accepted": 0,
                            "data": {
                                "companyname": "John and Yoko",
                                "phone": "616060184",
                                "IBAN": "123456789019284",
                                "companyDescription": "",
                                "discount": 0
                            }
                        }
                    ]
                }
            }
        }
    ]
}
```

GET /users/{user_id}

Get specified user. Authorization required. Only Admin.

PARAMETERS

No parameters

EXAMPLE REQUEST

GET http://funacademy.gergonzalez.com/users/1

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 1,
    "email": "admin@gergonzalez.com",
    "type": "Admin",
    "activated": 1,
    "createdAt": "Sat, Jul 29, 2017 7:59 PM",
    "info": {
        "name": "Admin"
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No query results for model [App\\User] 5"
    }
}
```

PATCH /website-users/{user_id}
Update specified user. Authorization required. Only Admin.

PARAMETERS
email
password
name
mobilePhone
createdAt
active

EXAMPLE REQUEST
PATCH http://funacademy.gergonzalez.com/website-users/2

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "id": 2,
    "email": "ger@gergonzalez.com",
    "type": "Website User",
    "activated": 0,
    "createdAt": "Sat, Jul 29, 2017 8:04 PM",
    "info": {
        "name": "Ringo Starr",
        "mobilePhone": "616060185"
    }
}
```

EXAMPLE ERROR RESPONSE
```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No user with id 14"
    }
}
```

PATCH /providers/{user_id}
Update specified user. Authorization required. Only Admin.

PARAMETERS
email
password
companyname
phone
IBAN
companyDescription
createdAt
active

EXAMPLE REQUEST
PATCH http://funacademy.gergonzalez.com/providers/3

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "id": 3,
    "email": "ger1@gergonzalez.com",
    "type": "Provider",
    "activated": "1",
    "createdAt": "Sat, Jul 29, 2017 8:06 PM",
    "info": {
        "companyname": "Sgt. Pepper's Lonely Hearts Club Band",
        "phone": "616060184",
        "IBAN": "123456789019284",
        "companyDescription": "",
        "discount": 0
    }
}
```

EXAMPLE ERROR RESPONSE
```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No user with id 14"
    }
}
```

PATCH /retailers/{user_id}

Update specified user. Authorization required. Only Admin.

PARAMETERS
email
password
name
retailerResponsibleName
phone
mobilePhone
address
IBAN
createdAt
active

EXAMPLE REQUEST
PATCH http://funacademy.gergonzalez.com/retailers/4

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 4,
    "email": "ger3@gergonzalez.com",
    "type": "Retailer",
    "activated": "1",
    "createdAt": "Sat, Jul 29, 2017 8:09 PM",
    "info": {
        "name": "Paul McCartney and Wings",
        "retailerResponsibleName": "The Beatles",
        "phone": "123456789",
        "mobilePhone": "616060184",
        "address": "Strawberry Fields Square",
        "IBAN": "123456789019284",
        "providers": {
            "data": [
                {
                    "id": 3,
                    "email": "ger1@gergonzalez.com",
                    "activated": 1,
                    "createdAt": "Jul 29, 2017",
                    "type": "Provider",
                    "accepted": 0,
                    "data": {
                        "companyname": "Sgt. Pepper's Lonely
Hearts Club Band",
                        "phone": "616060184",
                        "IBAN": "123456789019284",
                        "companyDescription": "",
                        "discount": 0
                    }
                }
            ]
        }
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No user with id 14"
    }
}
```

POST /users/activate/{user_id}

Activate specified user. Authorization required. Only Admin.

PARAMETERS
No parameters

EXAMPLE REQUEST

POST http://funacademy.gergonzalez.com/users/activate/2

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 2,
    "email": "ger@gergonzalez.com",
    "type": "Website User",
    "activated": 1,
    "createdAt": "Sat, Jul 29, 2017 8:04 PM",
    "info": {
        "name": "Ringo Starr",
        "mobilePhone": "616060185"
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No query results for model [App\\User] 10"
    }
}
```

POST /providers/{user_id}/discount
Set provider discount. Authorization required. Only Admin.

PARAMETERS
discount: Integer. Between 0-100 **(***required)*

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/providers/3/discount

EXAMPLE SUCCESS RESPONSE
```
HTTP 1.1 200 OK
{
    "id": 3,
    "email": "ger1@gergonzalez.com",
    "type": "Provider",
    "activated": 1,
    "createdAt": "Sat, Jul 29, 2017 8:06 PM",
    "info": {
        "companyname": "Sgt. Pepper's Lonely Hearts Club Band",
        "phone": "616060184",
        "IBAN": "123456789019284",
        "companyDescription": "",
        "discount": "25"
    }
}
```

EXAMPLE ERROR RESPONSE
```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "The Customer must be a provider"
    }
}
```

POST /retailers/{retailer_user_id}/add-provider/{provider_user_id}

Attach provider to retailer. Authorization required. Admin or retailer with retailer_user_id.

PARAMETERS
No parameters

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/retailers/4/add-provider/3

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 4,
    "email": "ger3@gergonzalez.com",
    "type": "Retailer",
    "activated": 1,
    "createdAt": "Sat, Jul 29, 2017 8:09 PM",
    "info": {
        "name": "Paul McCartney and Wings",
        "retailerResponsibleName": "The Beatles",
        "phone": "123456789",
        "mobilePhone": "616060184",
        "address": "Strawberry Fields Square",
        "IBAN": "123456789019285",
        "providers": {
            "data": [
                {
                    "id": 3,
                    "email": "ger1@gergonzalez.com",
                    "activated": 1,
                    "createdAt": "Jul 29, 2017",
                    "type": "Provider",
                    "accepted": 0,
                    "data": {
                        "companyname": "Sgt. Pepper's Lonely
Hearts Club Band",
                        "phone": "616060184",
                        "IBAN": "123456789019284",
                        "companyDescription": "",
                        "discount": 25
                    }
                },
                {
                    "id": 5,
                    "email": "ger2@gergonzalez.com",
                    "activated": 1,
                    "createdAt": "Jul 29, 2017",
                    "type": "Provider",
                    "accepted": 0,
                    "data": {
                        "companyname": "George Harrison",
                        "phone": "616060184",
                        "IBAN": "123456789019284",
                        "companyDescription": "",
                        "discount": 10
                    }
                }
            ]
        }
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No users for those id"
    }
}
```

POST /providers/{provider_user_id}/accept-retailer/{retailer_user_id}

Provider confirms retailer. Authorization required. Admin or provider with provider_user_id.

PARAMETERS
No parameters

EXAMPLE REQUEST
POST http://funacademy.gergonzalez.com/providers/3/accept-retailer/4

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 4,
    "email": "ger3@gergonzalez.com",
    "type": "Retailer",
    "activated": 1,
    "createdAt": "Sat, Jul 29, 2017 8:09 PM",
    "info": {
        "name": "Paul McCartney and Wings",
        "retailerResponsibleName": "The Beatles",
        "phone": "123456789",
        "mobilePhone": "616060184",
        "address": "Strawberry Fields Square",
        "IBAN": "123456789019285",
        "providers": {
            "data": [
                {
                    "id": 3,
                    "email": "ger1@gergonzalez.com",
                    "activated": 1,
                    "createdAt": "Jul 29, 2017",
                    "type": "Provider",
                    "accepted": 1,
                    "data": {
                        "companyname": "Sgt. Pepper's Lonely
Hearts Club Band",
                        "phone": "616060184",
                        "IBAN": "123456789019284",
                        "companyDescription": "",
                        "discount": 25
                    }
                },
                {
                    "id": 5,
                    "email": "ger2@gergonzalez.com",
                    "activated": 1,
                    "createdAt": "Jul 29, 2017",
                    "type": "Provider",
                    "accepted": 0,
                    "data": {
                        "companyname": "George Harrison",
                        "phone": "616060184",
                        "IBAN": "123456789019284",
                        "companyDescription": "",
                        "discount": 10
                    }
                }
            ]
        }
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No users for those id"
    }
}
```

GET /products

Return all products. Authorization required. Only Admin.

PARAMETERS
No parameters

EXAMPLE REQUEST
GET http://funacademy.gergonzalez.com/products

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "data": [
        {
            "id": 1,
            "name": "Cherries",
            "price": "0.77"
        }
    ]
}
```

GET /orders

Return all orders. Authorization required. Only Admin.

PARAMETERS
No parameters

EXAMPLE REQUEST
GET http://funacademy.gergonzalez.com/orders

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "data": [
        {
            "id": 1,
            "quantity": 3,
            "amount": 2.31,
            "status": "pending",
            "createdAt": "Mon, Jul 31, 2017 8:49 AM",
            "user": {
                "id": 2,
                "email": "ger@gergonzalez.com",
                "type": "Website User",
                "activated": 1,
                "createdAt": "Sat, Jul 29, 2017 8:04 PM",
                "info": {
                    "name": "Ringo Starr",
                    "mobilePhone": "616060185"
                }
            },
            "product": {
                "id": 1,
                "name": "Cherries",
                "price": "0.77"
            }
        }
    ]
}
```

POST /orders

Store a order. Authorization required. Registered User.

PARAMETERS

quantity *required*

product *required*

EXAMPLE REQUEST

POST http://funacademy.gergonzalez.com/orders

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 1,
    "quantity": "3",
    "amount": 2.31,
    "status": "pending",
    "createdAt": "Mon, Jul 31, 2017 8:49 AM",
    "user": {
        "id": 2,
        "email": "ger@gergonzalez.com",
        "type": "Website User",
        "activated": 1,
        "createdAt": "Sat, Jul 29, 2017 8:04 PM",
        "info": {
            "name": "Ringo Starr",
            "mobilePhone": "616060185"
        }
    },
    "product": {
        "id": 1,
        "name": "Cherries",
        "price": "0.77"
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": "As a website user you can't order more than 10
units. Today you already ordered 3. Order not processed"
}
```

PATCH /orders/{order_id}

Return all orders. Authorization required. Only Admin.

PARAMETERS
status: text (*required)*
observation: text (*required if status == cancelled)*

EXAMPLE REQUEST
PATCH http://funacademy.gergonzalez.com/orders/1

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "id": 1,
    "quantity": 3,
    "amount": 2.31,
    "status": "cancelled",
    "createdAt": "Mon, Jul 31, 2017 8:49 AM",
    "user": {
        "id": 2,
        "email": "ger@gergonzalez.com",
        "type": "Website User",
        "activated": 1,
        "createdAt": "Sat, Jul 29, 2017 8:04 PM",
        "info": {
            "name": "Ringo Starr",
            "mobilePhone": "616060185"
        }
    },
    "product": {
        "id": 1,
        "name": "Cherries",
        "price": "0.77"
    }
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 422 UNPROCESSABLE ENTITY
{
    "error": {
        "message": "The given data failed to pass validation.",
        "data": {
            "status": [
                "The selected status is invalid."
            ]
        }
    }
```

DELETE /orders/{order_id}

Return all orders. Authorization required. Only Admin.

PARAMETERS
No parameters

EXAMPLE REQUEST
GET http://funacademy.gergonzalez.com/orders/1

EXAMPLE SUCCESS RESPONSE

```
HTTP 1.1 200 OK
{
    "data": [
        "Order successfully deleted"
    ]
}
```

EXAMPLE ERROR RESPONSE

```
HTTP 1.1 400 BAD REQUEST
{
    "error": {
        "message": "No order with id 7"
    }
}
```

FINAL CONSIDERATIONS

During the ~25 hours I dedicated to implement, document and test the project, I tried to use the most standard, readable and efficient solution possible. Also, I tried to use as much Lumen features as I could.

About the extra features, due to the time constraints, I couldn't implement them, but for implement:

- There are some Swagger packages on Lumen like https://github.com/DarkaOnLine/SwaggerLume for easily use Swagger.
- The same for Graphql https://github.com/Folkloreatelier/laravel-graphql
- About the file upload, it is needed to add the official Laravel file management package and then you can store files local or S3
- The password recovery needs you to add a new table, something like password_recovery, and then implement the logic to handle it.

I hope this test gives you a glimpse about what my skills and strengths are.

Please, don’t hesitate to contact me if you need any further information.

Thanks so much for giving me the opportunity,
Germán González Rodríguez
July 31th, 2017