

Kuizawa_api

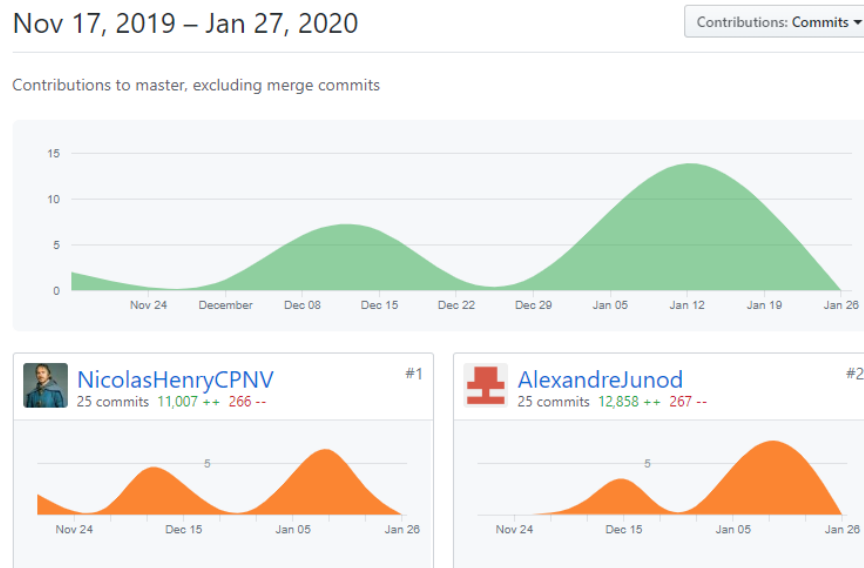
Introduction

Ce projet a pour but de réaliser une API pour les applications mobiles et une page d'administration.

Afin de réaliser ce projet, nous avons choisis d'utiliser le Framework Laravel, car il permet de :

- Gérer une base de donnée
- Fournir des routes API
- Gérer les données retournées de façon simple
- Faciliter la création de models/controller/provider
- Gérer l'authentification par token

Nous avons également réparti la charge de travail de façon égale.



Libraires choisies

Swagger : Cette librairie nous permet de documenter une API de façon simple et directement dans le code grâce à des commentaires. Celle-ci nous permet de gagner du temps afin de ne pas avoir à entretenir une autre documentation en parallèle.

Explication du code

Notre back-end possède des migrations, afin de créer la base de donnée.

Pour gérer les relations entre les model nous utilisons les classes respectives qui font référence aux migrations.

Afin de renvoyer des données aux utilisateurs nous utilisons des routes dans laquelle une authentification par token est faite par le middleware.

Les données renvoyées à l'utilisateur sont gérées par le controller, celui-ci va appeler une ressource créer au préalable qui renvoie des données sous forme de collection.

Les tags et la doc de swagger est générée par les commentaires au niveau des controlleurs.

```

class User extends Authenticatable
{
    use Notifiable, SoftDeletes;

    public $timestamps = false;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'email', 'password', 'pseudo', 'firstname', 'lastname', 'admin',
        'creator', 'classroom_id', 'api_token',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    public function quizzes() {
        return $this->hasMany('App\Quizz');
    }

    public function classroom() {
        return $this->belongsTo('App\Classroom');
    }
}

```

1. Classe User (model)

```
Route::apiResource('users', 'Api\UserController')->except('store');
```

2. Route User

```
public function index()
{
    return new UserCollection(User::orderBy('lastname')->get());
}
```

3. Controller User (index)

```
public function toArray($request)
{
    return [
        'id' => $this->id,
        'email' => $this->email,
        'pseudo' => $this->pseudo,
        'firstname' => $this->firstname,
        'lastname' => $this->lastname,
    ];
}
```

4. Ressource de User