Restful API with Laravel 9

Validation

Objective

1. Why need validation?
2. Modifying EmployeesController class
3. Modifying UsersController class

Why need validation?

- A programmer should consider that any inputs a user makes may be incorrect and should plan arrangements for such unexpected actions.
- Using validation helps a programmer to ensure that any data input is possible and sensible.
- Validation applies rules to inputted data. If the data does not follow the rules, it is rejected, reducing the risk that incorrectly input data may crash a program.
- Url: https://laravel.com/docs/9.x/validation

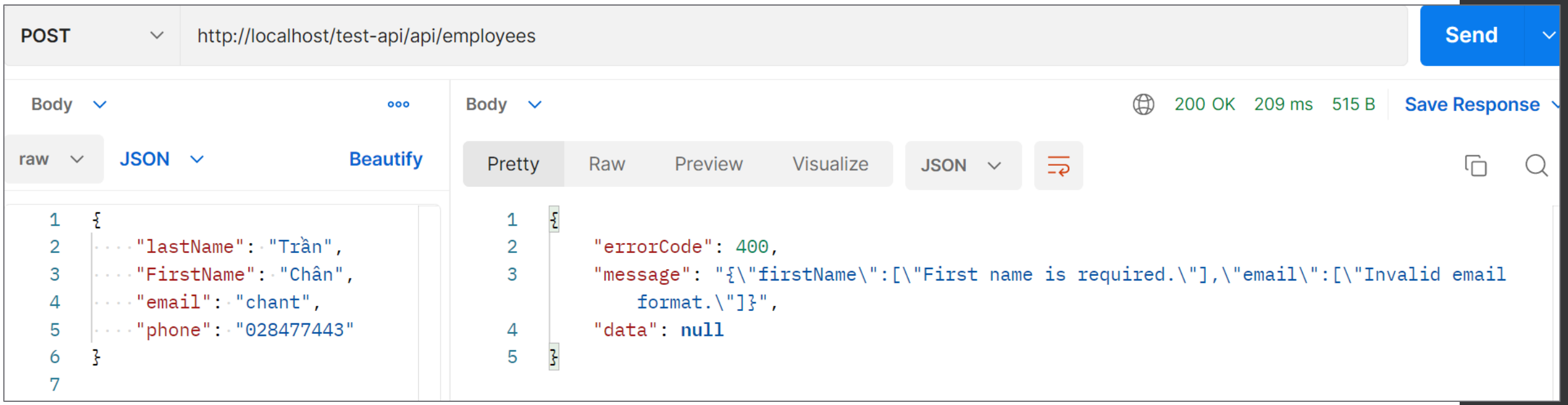
Modifying EmployeesController class

```php
use Illuminate\Support\Facades\Validator;

class EmployeesController extends Controller
{
    private $rules = [
        'lastName' => 'required|min:2',
        'firstName' => 'required',
        'email' => 'email'
    ];
    private $messages = [
        'lastName.required' => 'Last name is required.',
        'lastName.min' => 'Last name must at least 2 characters.',
        'firstName.required' => 'First name is required.',
        'email.email' => 'Invalid email format.'
    ];
```

```php
public function create(Request $request) {
    $validator = Validator::make($request->all(), $this->rules, $this->messages);
    if ($validator->fails()) {
        return BaseResponse::error(400, $validator->messages()->toJson());
    } else {
        try {
            $employee = new Employee();
            $employee->LastName = $request->input('lastName');
            $employee->FirstName = $request->firstName;
            $employee->Email = $request->email;
            $employee->Phone = $request->phone;
            $employee->save();
            return BaseResponse::withData($employee);
        } catch(Throwable $e) {
            return BaseResponse::error(500, $e->getMessage());
        }
    }
}
```

```php
public function update($id, Request $request) {
    $validator = Validator::make($request->all(), $this->rules, $this->messages);
    if ($validator->fails()) {
        return BaseResponse::error(400, $validator->messages()->toJson());
    } else {
        $employee = Employee::find($id);
        if ($employee) {
            try {
                $employee->LastName = $request->lastName;
                $employee->FirstName = $request->firstName;
                if ($request->has('email')) $employee->Email = $request->email;
                if ($request->has('phone')) $employee->Phone = $request->phone;

                $employee->save();
                return BaseResponse::withData($employee);
            } catch(Throwable $e) {
                return BaseResponse::error(500, $e->getMessage());
            }
        } else {
            return BaseResponse::error(404, 'Data not found!');
        }
    }
}
```

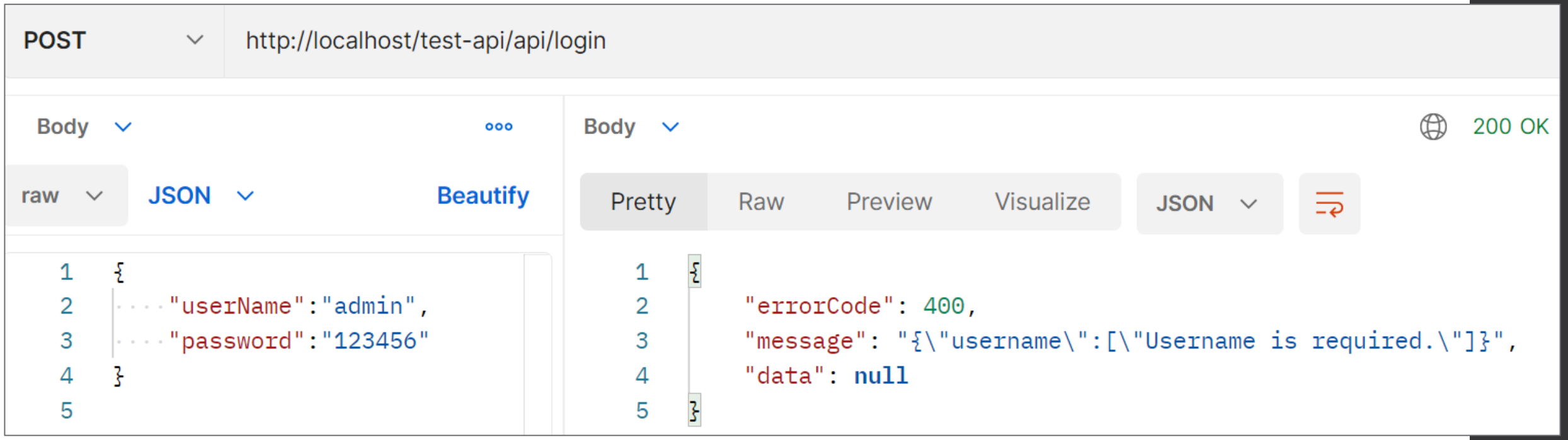
Testing


POST
http://localhost/test-api/api/employees
Send
Body
raw
JSON
Beautify
{
"lastName": "Trần",
"FirstName": "Chân",
"email": "chant",
"phone": "028477443"
}
Body
200 OK
209 ms
515 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"errorCode": 400,
"message": "{\"firstName\":[\"First name is required.\"],\"email\":[\"Invalid email format.\"]}",
"data": null
}

Modifying UsersController class

```
    public function login(Request $req) {
        $rules = [ 'username' => 'required', 'password' => 'required' ];
        $messages = [
            'username.required' => 'Username is required.',
            'password.required' => 'Password is requied.',
        ];

        $validator = Validator::make($req->all(), $rules, $messages);
        if ($validator->fails()) {
            return BaseResponse::error(400, $validator->messages()->toJson());
        } else {
            $credentials = [ ···
            ];
            if (auth()->attempt($credentials)) { ···
            } else { ···
            }
        }
    }
```

Testing


POST
http://localhost/test-api/api/login
Body
raw
JSON
Beautify
1 {
2 "userName":"admin",
3 "password":"123456"
4 }
5
Body
200 OK
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "errorCode": 400,
3 "message": "{\"username\":[\"Username is required.\"]}",
4 "data": null
5 }