Unit

API Validation

API Validation:

when we receive any kind of data from our API its required to validate form of data and type of data to ensure correctness of received data and ensure integrity of our database.

Create new Project.

Open Visual Studio Code.

Open Console with ctrl + j shortcut

Go to Laravel projects folder

Create new project with: **laravel new APIValidation** Command

Select MySql as Database and migrate it

After completed of project creation.

Open mysql server with localhost/phpMyAdmin and select your database

Create following table

CREATE TABLE students (id int AUTO_INCREMENT PRIMARY key, fname varchar(20), lname varchar(20), city varchar(20), email varchar(64), phone varchar(15), gender varchar(10), age tinyint, created_at timestamp DEFAULT CURRENT_TIMESTAMP, updated_at timestamp DEFAULT CURRENT_TIMESTAMP on UPDATE CURRENT_TIMESTAMP)

Open newly created project in code editor.

Open console in vs code and run following command for install API module in Laravel

php artisan install:api

create required model and controller

php artisan make:model student

php artisan make:controller studentController –api

open api.php file from routes

api.php code

```php
<?php

use App\Http\Controllers\studentController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

Route::get('/user', function (Request $request) {
    return $request->user();
})->middleware('auth:sanctum');

Route::resource("/students", studentController::class);
```

Run project with following command.

php artisan serve

Controller Code

Create New Student API

Import validator and student model in Controller

```
use Illuminate\Support\Facades\Validator;
use App\Models\student;
```

Code for Store Function

```
/**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        //
        $validationRules = [
            "fname" => "required",
            "lname" => "required",
            "city" => "required",
            "email" => "required|email|unique:students,email",
            "phone" => "required|numeric|digits:10",
            "gender" => "required|in:male,female",
            "age" => "required|numeric|max:100|min:0",
        ];

        $validationResult = validator::make($request->all(),
$validationRules);

        if(!$validationResult->fails()){
            $student = new student();

            $student->fname = $request->fname;
```

```
            $student->lname = $request->lname;
            $student->city = $request->city;
            $student->email = $request->email;
            $student->phone = $request->phone;
            $student->gender = $request->gender;
            $student->age = $request->age;

            $student->save();

            return response([
                "message" => "New Student Created"
            ], 200);

        }else{
            return response([
                "message" => "Validation Failed",
                "errors" => $validationResult->errors()
            ], 202);
        }
    }
```

For Testing API

Open Postman

Create new POST request

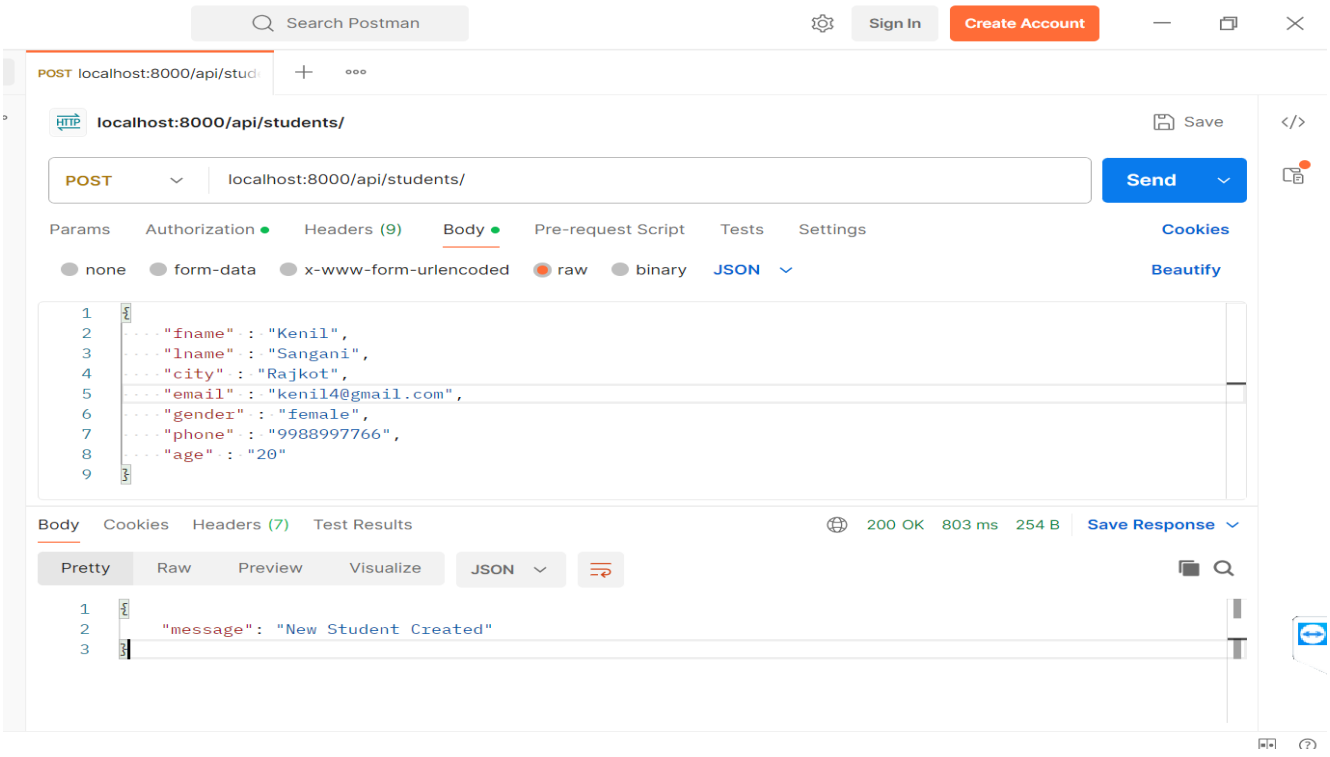
Enter Following URL : localhost:8000/api/students/

Go to body -> raw -> select JSON format and send data in JSON format like following

```
{
        "fname": "Kenil",
        "lname": "Sangani",
        "city": "Rajkot",
        "email": "kenil4@gmail.com",
        "gender": "female",
        "phone": "9988997766",
```

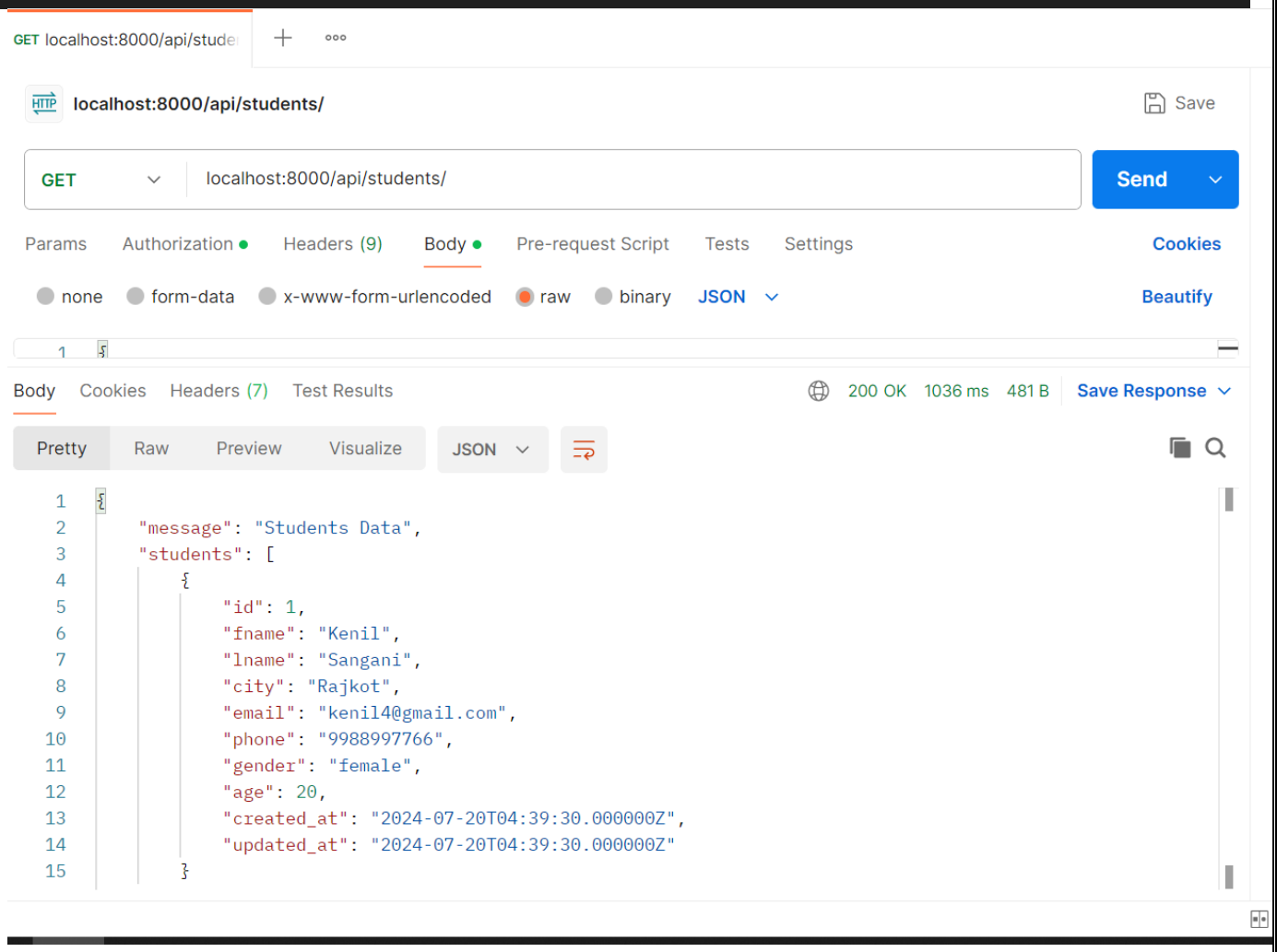
```
    "age": "20"
}
```

Click on send Button.



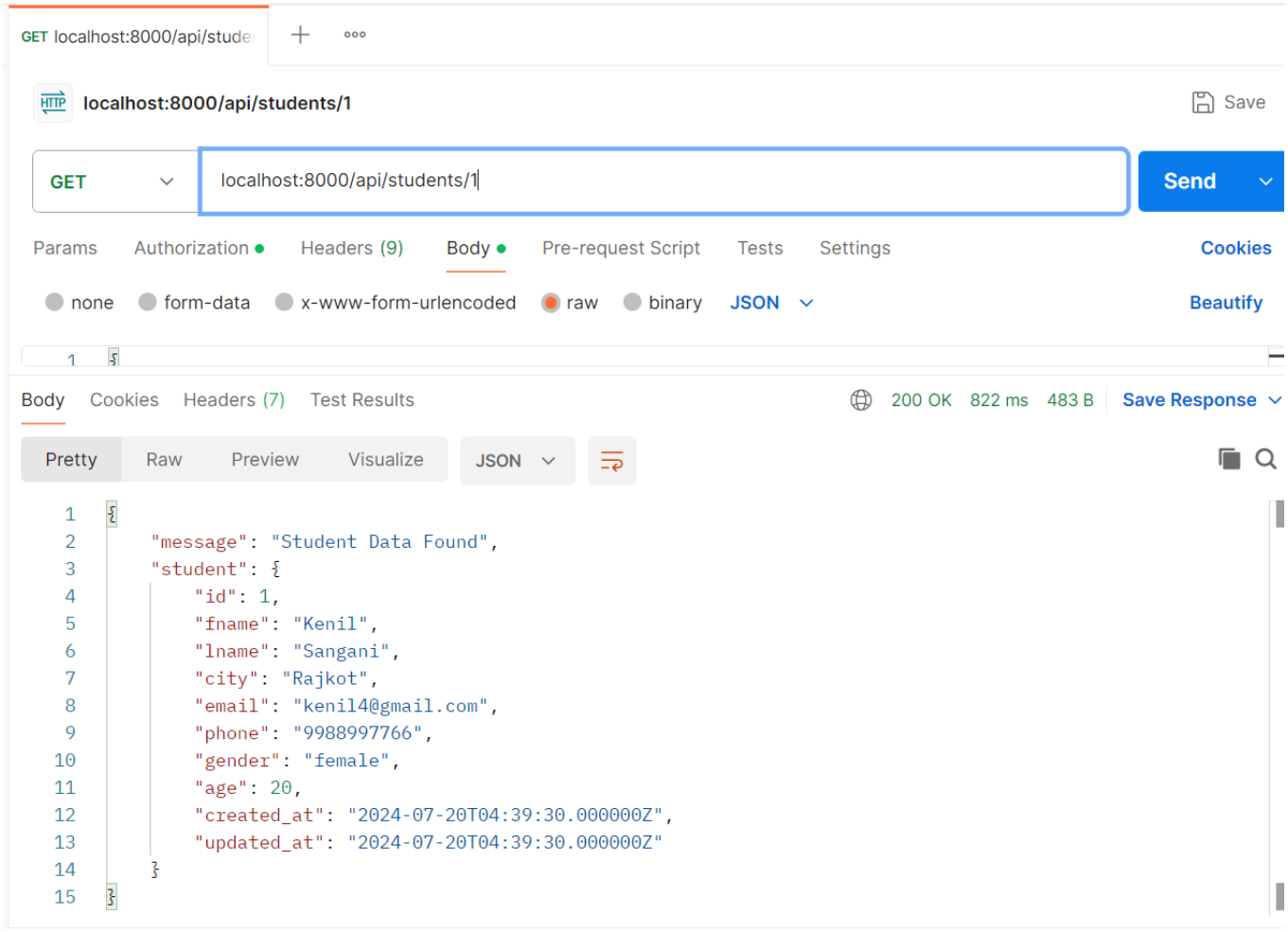
Get All Students API with index function.

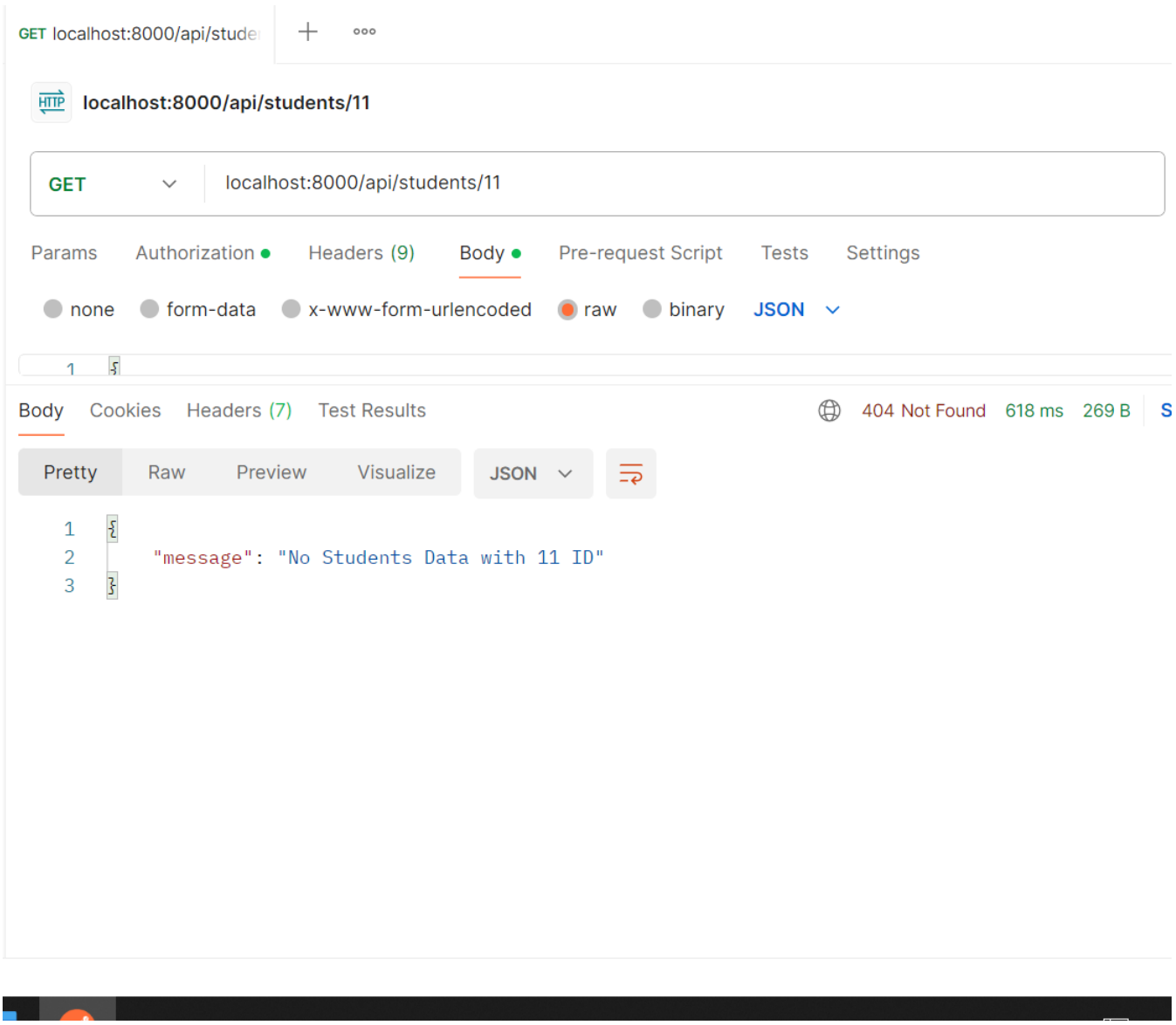
```php
/**
     * Display a listing of the resource.
     */
    public function index()
    {
        //
        $studentsData = student::all();
        return response([
            "message" => "Students Data",
            "students" => $studentsData
        ], 200);
    }
```



Find data of Specific ID

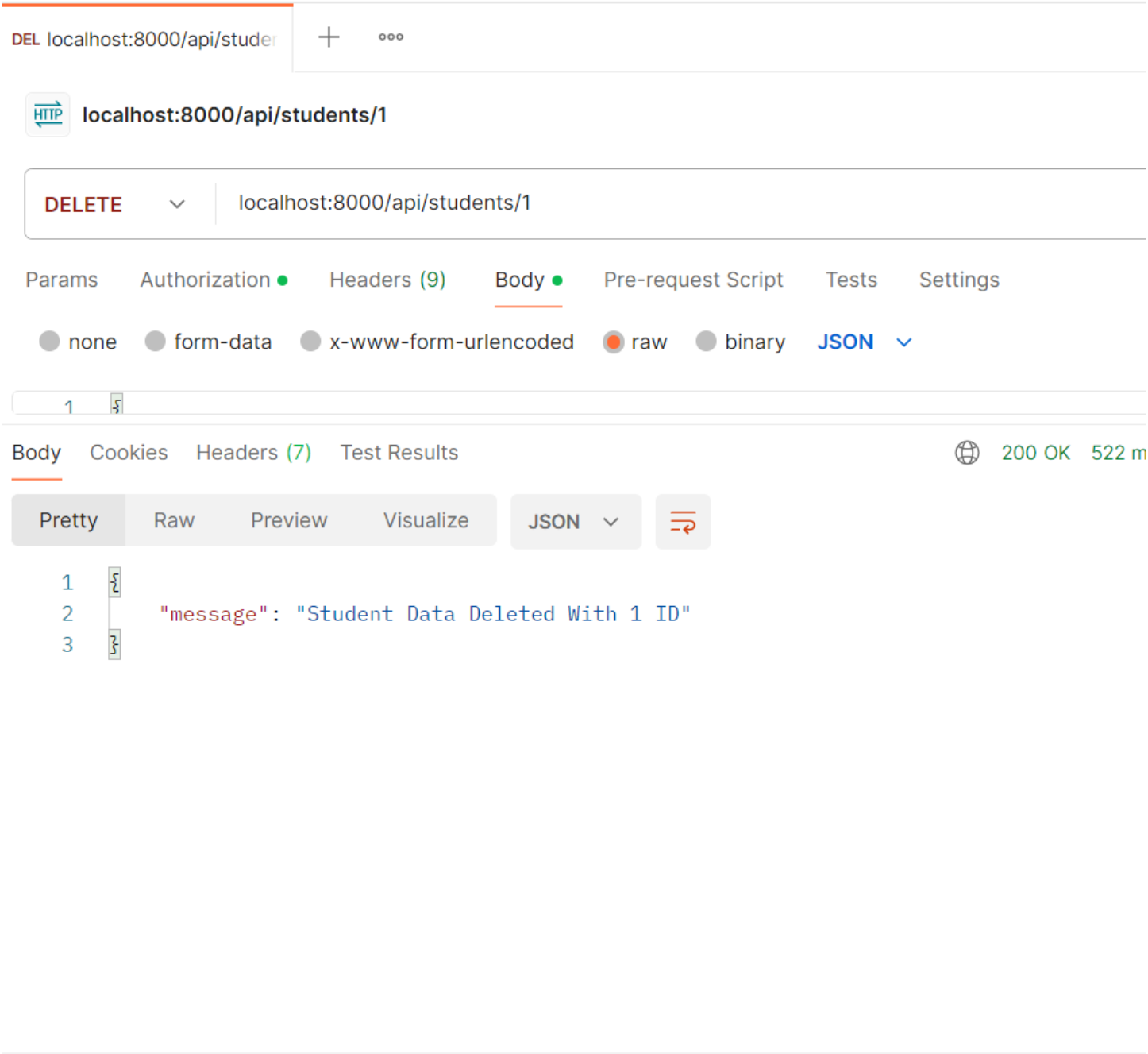
```
/**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
        $studentData = student::find($id);

        if(!$studentData == null){
            return response([
                "message" => "Student Data Found",
                "student" => $studentData
            ], 200);
        }else{
            return response([
                "message" => "No Students Data with $id ID",
            ], 404);
        }
    }
```

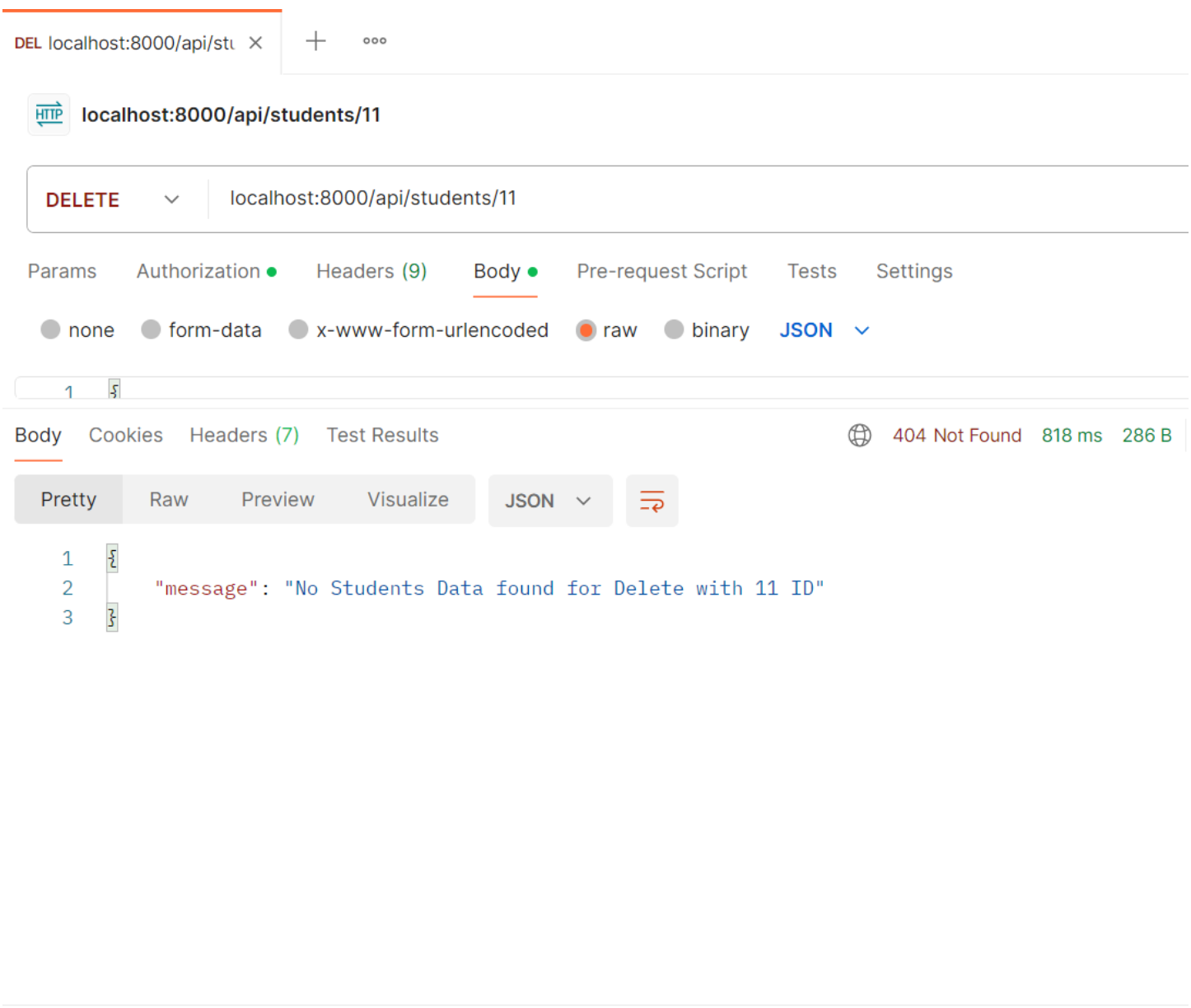

GET localhost:8000/api/stude
localhost:8000/api/students/1
Save
GET
localhost:8000/api/students/1
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
JSON
Beautify
Body
Cookies
Headers (7)
Test Results
200 OK
822 ms
483 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Student Data Found",
"student": {
"id": 1,
"fname": "Kenil",
"lname": "Sangani",
"city": "Rajkot",
"email": "kenil4@gmail.com",
"phone": "9988997766",
"gender": "female",
"age": 20,
"created_at": "2024-07-20T04:39:30.000000Z",
"updated_at": "2024-07-20T04:39:30.000000Z"
}
}


GET localhost:8000/api/stude
localhost:8000/api/students/11
GET
localhost:8000/api/students/11
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
JSON
Body
Cookies
Headers (7)
Test Results
404 Not Found
618 ms
269 B
Pretty
Raw
Preview
Visualize
JSON
{
"message": "No Students Data with 11 ID"
}

Delete API

```
/**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        //
        $studentData = student::find($id);

        if(!$studentData == null){
            $studentData->delete();
            return response([
                "message" => "Student Data Deleted With $id ID",
            ], 200);
        }else{
            return response([
                "message" => "No Students Data found for Delete
with $id ID",
            ], 404);
        }
    }
```

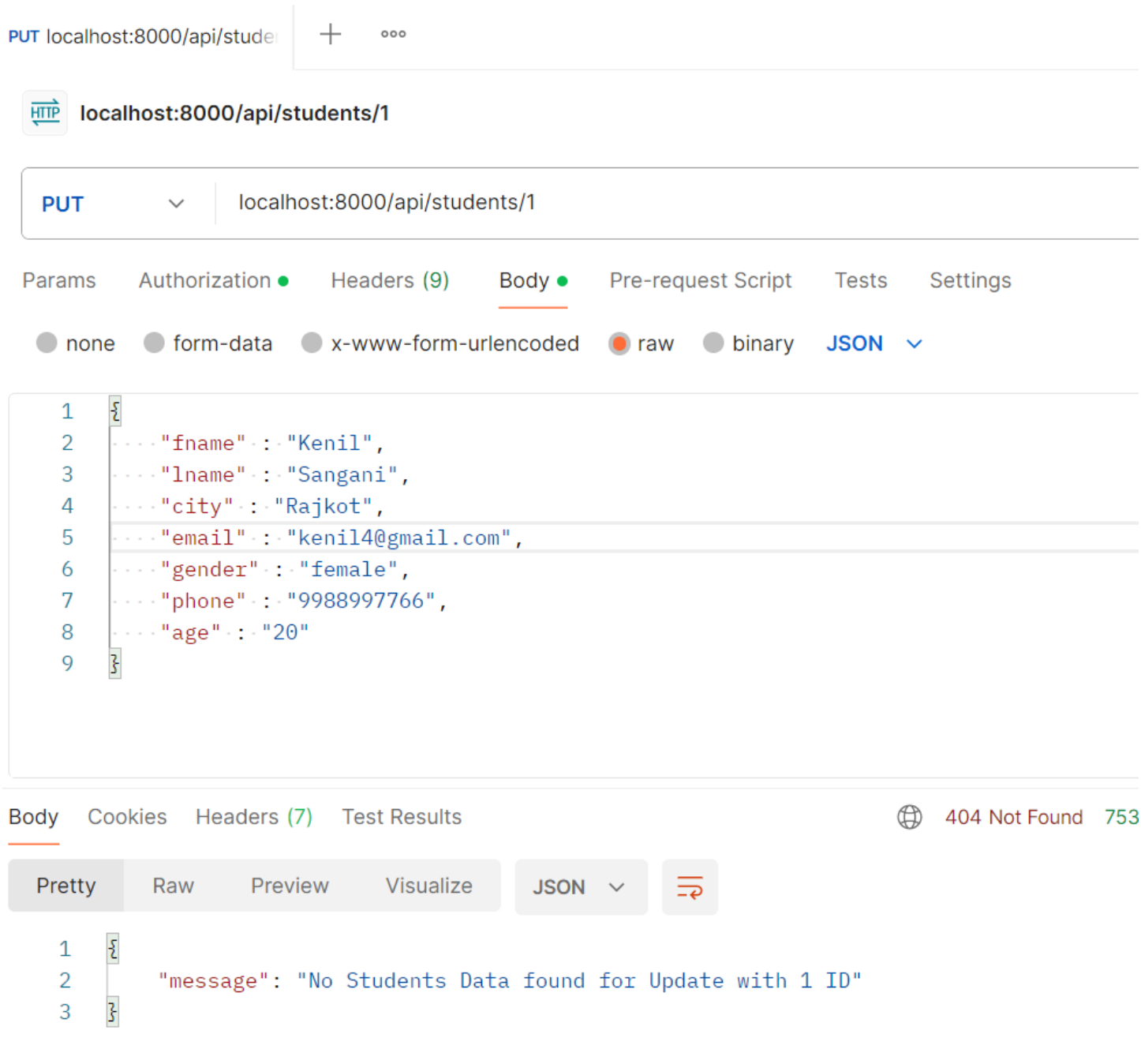

DEL localhost:8000/api/stude
localhost:8000/api/students/1
DELETE
localhost:8000/api/students/1
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
JSON
Body
Cookies
Headers (7)
Test Results
200 OK
522 m
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Student Data Deleted With 1 ID"
}

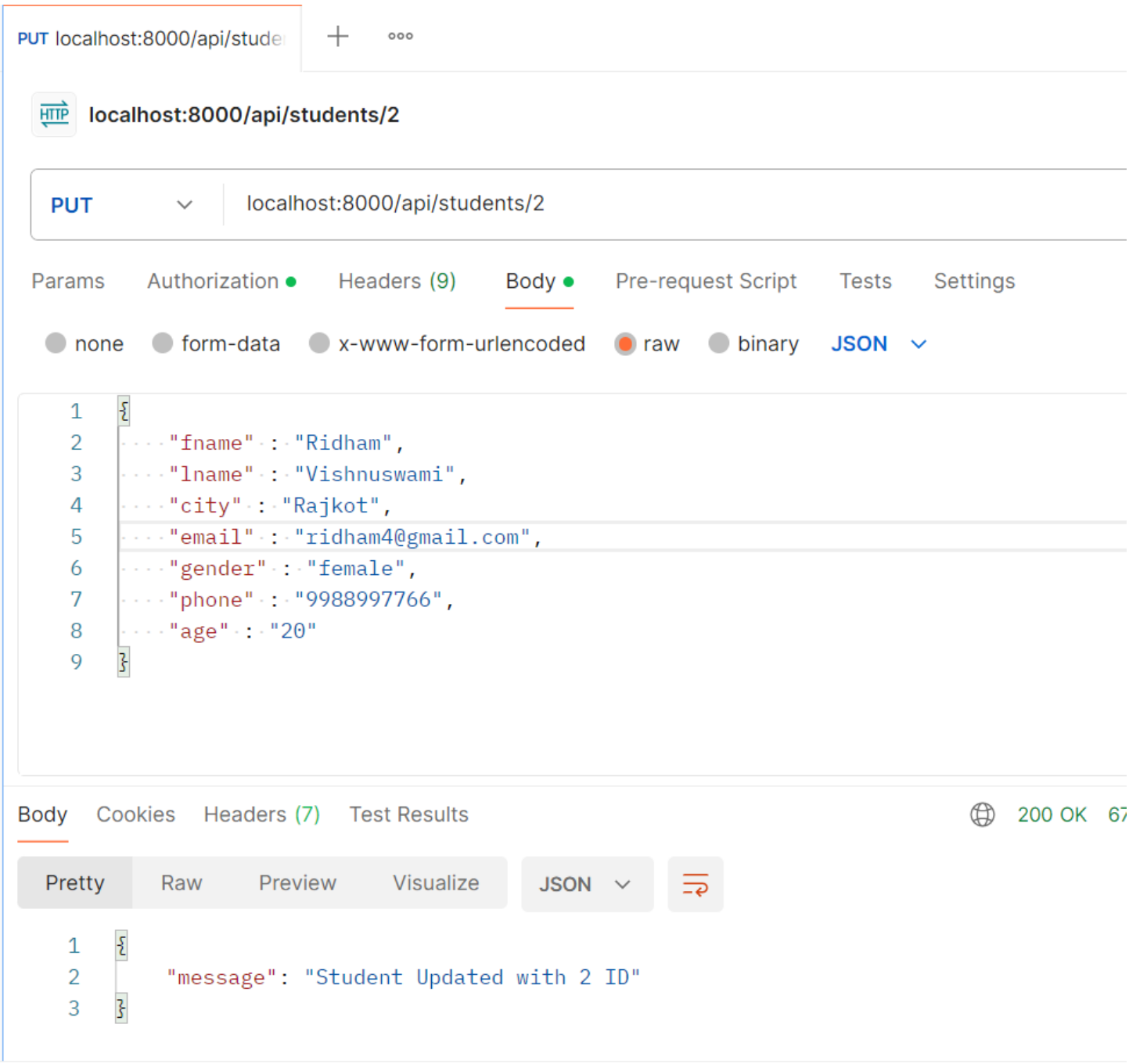

DEL localhost:8000/api/stu
localhost:8000/api/students/11
DELETE
localhost:8000/api/students/11
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
JSON
1 {
Body
Cookies
Headers (7)
Test Results
404 Not Found
818 ms
286 B
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "message": "No Students Data found for Delete with 11 ID"
3 }

Update Student API

```
 /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        //
        $studentData = student::find($id);
        if(!$studentData == null){
            $validationRules = [
                "fname" => "required",
                "lname" => "required",
                "city" => "required",
                "email" =>
"required|email|unique:students,email",
                "phone" => "required|numeric|digits:10",
                "gender" => "required|in:male,female",
                "age" => "required|numeric|max:100|min:0",
            ];

            $validationResult = validator::make($request->all(),
$validationRules);

            if(!$validationResult->fails()){


                $studentData->fname = $request->fname;
                $studentData->lname = $request->lname;
                $studentData->city = $request->city;
                $studentData->email = $request->email;
                $studentData->phone = $request->phone;
```

```
                $studentData->gender = $request->gender;
                $studentData->age = $request->age;

                $studentData->save();

                return response([
                    "message" => "Student Updated with $id ID"
                ], 200);

            }else{
                return response([
                    "message" => "Validation Failed",
                    "errors" => $validationResult->errors()
                ], 202);
            }
        }else{
            return response([
                "message" => "No Students Data found for Update
with $id ID",
            ], 404);
        }
    }
```


PUT localhost:8000/api/stude
localhost:8000/api/students/1
PUT
localhost:8000/api/students/1
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
JSON
{
"fname" : "Kenil",
"lname" : "Sangani",
"city" : "Rajkot",
"email" : "kenil4@gmail.com",
"gender" : "female",
"phone" : "9988997766",
"age" : "20"
}
Body
Cookies
Headers (7)
Test Results
404 Not Found
753
Pretty
Raw
Preview
Visualize
JSON
{
"message": "No Students Data found for Update with 1 ID"
}


PUT localhost:8000/api/stude
localhost:8000/api/students/2
PUT
localhost:8000/api/students/2
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
JSON
{
"fname" : "Ridham",
"lname" : "Vishnuswami",
"city" : "Rajkot",
"email" : "ridham4@gmail.com",
"gender" : "female",
"phone" : "9988997766",
"age" : "20"
}
Body
Cookies
Headers (7)
Test Results
200 OK
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Student Updated with 2 ID"
}

GET localhost:8000/api/st

localhost:8000/api/students/2

GET localhost:8000/api/students/2

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (7) Test Results 200 OK 779 ms 489 B

Pretty Raw Preview Visualize JSON


```
{
    "message": "Student Data Found",
    "student": {
        "id": 2,
        "fname": "Ridham",
        "lname": "Vishnuswami",
        "city": "Rajkot",
        "email": "ridham4@gmail.com",
        "phone": "9988997766",
        "gender": "female",
        "age": 20,
        "created_at": "2024-07-20T04:54:54.000000Z",
        "updated_at": "2024-07-20T05:03:07.000000Z"
    }
}
```


studentController.php File

```php
<?php
namespace App\Http\Controllers;
use App\Models\student;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
class studentController extends Controller
{
    /** Display a listing of the resource.*/
    public function index()
    {
        $studentsData = student::all();
        return response([
            "message" => "Students Data",
            "students" => $studentsData
        ], 200);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        //
        $validationRules = [
            "fname" => "required",
            "lname" => "required",
            "city" => "required",
            "email" => "required|email|unique:students,email",
            "phone" => "required|numeric|digits:10",
            "gender" => "required|in:male,female",
            "age" => "required|numeric|max:100|min:0",
```

```
        ];

        $validationResult = validator::make($request->all(), $validationRules);

        if(!$validationResult->fails()){
            $student = new student();

            $student->fname = $request->fname;
            $student->lname = $request->lname;
            $student->city = $request->city;
            $student->email = $request->email;
            $student->phone = $request->phone;
            $student->gender = $request->gender;
            $student->age = $request->age;

            $student->save();

            return response([
                "message" => "New Student Created"
            ], 200);

        }else{
            return response([
                "message" => "Validation Failed",
                "errors" => $validationResult->errors()
            ], 202);
        }
    }
```

```php
/**
 * Display the specified resource.
 */
public function show(string $id)
{
    //
    $studentData = student::find($id);

    if(!$studentData == null){
        return response([
            "message" => "Student Data Found",
            "student" => $studentData
        ], 200);
    }else{
        return response([
            "message" => "No Students Data with $id ID",
        ], 404);
    }
}
```

```php
/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    //
    $studentData = student::find($id);
    if(!$studentData == null){
        $validationRules = [
            "fname" => "required",
            "lname" => "required",
            "city" => "required",
            "email" => "required|email|unique:students,email",
            "phone" => "required|numeric|digits:10",
            "gender" => "required|in:male,female",
            "age" => "required|numeric|max:100|min:0",
        ];

        $validationResult = validator::make($request->all(), $validationRules);

        if(!$validationResult->fails()){


            $studentData->fname = $request->fname;
            $studentData->lname = $request->lname;
            $studentData->city = $request->city;
            $studentData->email = $request->email;
            $studentData->phone = $request->phone;
            $studentData->gender = $request->gender;
            $studentData->age = $request->age;

            $studentData->save();
```

```
            return response([
                "message" => "Student Updated with $id ID"
            ], 200);

        }else{
            return response([
                "message" => "Validation Failed",
                "errors" => $validationResult->errors()
            ], 202);
        }
    }else{
        return response([
            "message" => "No Students Data found for Update with $id ID",
        ], 404);
    }
}
```

```
    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        //
        $studentData = student::find($id);

        if(!$studentData == null){
            $studentData->delete();
            return response([
                "message" => "Student Data Deleted With $id ID",
            ], 200);
        }else{
            return response([
                "message" => "No Students Data found for Delete with $id ID",
            ], 404);
        }
    }
}
```