

Build Simple REST API with PHP and MySQL

📅 [October 28, 2019](#) 👤 by [TechArise Team](#)

REST stands for Representational State Transfer. RESTful Web services are one way of providing interoperability between computer systems on the Internet. Rest API help to communication between the client app and the server application.REST is an architecture style for designing networked applications.A REST API defines bunch of functions which developers can perform requests and receive responses via HTTP protocol such as GET and POST.

In this tutorial, you will learn to how create CRUD (create, read, update, delete) operation REST API with PHP and MySQL. This is a very simple example, you can just copy paste and change according to your requirement.

[Also Read : Simple CRUD implementation with CodeIgniter using Mysql, Ajax and Bootstrap Model](#)

Before started to implement the REST API with PHP and MySQL, look files structure:

```
build-simple-rest-api-with-php-mysql
├── class
│   ├── DBConnection.php
│   └── Student.php.
├── student
│   ├── create.php
│   ├── read.php
│   ├── update.php
│   ├── delete.php
│   └── .htaccess
```

Understanding REST API

REST provides a block of HTTP methods which are used to alter the data. The following are common HTTP methods:

- **GET** — is used for reading and retrieving data.
- **POST** — is used for inserting data.
- **PUT/PATCH** — is used for updating data.
- **DELETE** — is used for deleting data.

Step 1: Create MySQL Database and Table

For this tutorial, you need a MySQL database with the following table:

```
1 CREATE TABLE `student` (  
2   `id` int(11) NOT NULL,  
3   `roll_no` varchar(10) DEFAULT 'NULL',  
4   `first_name` varchar(255) NOT NULL,  
5   `last_name` varchar(255) NOT NULL,  
6   `class` varchar(255) NOT NULL,  
7   `age` int(11) NOT NULL,  
8   `address` varchar(255) NOT NULL,  
9   `status` int(1) NOT NULL DEFAULT 0  
10  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
11  
12 INSERT INTO `student` (`id`, `roll_no`, `first_name`, `last_name`, `class`, `age`, `address`, `status`) VALUES  
13 (1, 'R001', 'Tiger', 'Wood', 'LKG', 3, 'USA', 1);  
14  
15 ALTER TABLE `student`  
16   ADD PRIMARY KEY (`id`);  
17  
18 ALTER TABLE `student`  
19   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

Step 2: Connect to Database file named DBConnection.php

The code below shows the database credentials

```
1 <?php  
2 /**  
3  * @package PHP Rest API(DBConnection)  
4  *  
5  * @author TechArise Team  
6  *  
7  * @email info@techarise.com  
8  *  
9  */  
10  
11 // Database Connection  
12 class DBConnection {  
13     private $_dbHostname = "localhost";  
14     private $_dbName = "test_DB";  
15     private $_dbUsername = "root";  
16     private $_dbPassword = "";  
17     private $_con;  
18  
19     public function __construct() {  
20         try {  
21             $this->_con = new PDO("mysql:host=$this->_dbHostname;dbname=$this->_dbName", $this->_dbUsername, $this->_dbPasswo  
22             $this->_con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
23         } catch(PDOException $e) {  
24             echo "Connection failed: " . $e->getMessage();  
25         }  
26  
27     }  
28     // return Connection  
29     public function returnConnection() {  
30         return $this->_con;  
31     }  
32 }  
33 ?>
```

Step 3: Create class

Create a class file named Student.php inside class/ folder.

- The Student class handles the CRUD process
- __construct() — Loads the required DBConnection.
- createStudent() — Add Student Record.
- updateStudent() — Update Student Record.
- getAllStudent() — get Student all Records.
- getStudent() — get Student single Record.
- deleteStudent() — delete Student Record.

```
1 <?php  
2 /**  
3  * @package Student class  
4  *  
5  * @author TechArise Team  
6  *  
7  * @email info@techarise.com  
8  *  
9  */  
10  
11 include("DBConnection.php");  
12 class Student  
13 {  
14     protected $db;  
15     private $_studentID;  
16     private $_firstName;  
17     private $_lastName;  
18     private $_rollNo;  
19     private $_className;
```

```
20     private $_age;
21
22     public function setStudentID($studentID) {
23         $this->_studentID = $studentID;
24     }
25     public function setFirstName($firstName) {
26         $this->_firstName = $firstName;
27     }
28     public function setLastName($lastName) {
29         $this->_lastName = $lastName;
30     }
31     public function setRollNo($rollNo) {
32         $this->_rollNo = $rollNo;
33     }
34     public function setClassName($className) {
35         $this->_className = $className;
36     }
37     public function setAge($age) {
38         $this->_age = $age;
39     }
40     public function setAddress($address) {
41         $this->_address = $address;
42     }
43
44     public function __construct() {
45         $this->db = new DBConnection();
46         $this->db = $this->db->returnConnection();
47     }
48
49     // create Student
50     public function createStudent() {
51         try {
52             $sql = 'INSERT INTO student (first_name, last_name, roll_no, class, age, address, status) VALUES (:first_name,
53                 $data = [
54                     'first_name' => $this->_firstName,
55                     'last_name' => $this->_lastName,
56                     'roll_no' => $this->_rollNo,
57                     'class' => $this->_className,
58                     'age' => $this->_age,
59                     'address' => $this->_address,
60                     'status' => 1,
61                 ];
62             $stmt = $this->db->prepare($sql);
63             $stmt->execute($data);
64             $status = $stmt->rowCount();
65             return $status;
66         } catch (Exception $e) {
67             die("Oh noes! There's an error in the query!");
68         }
69     }
70
71 }
72
73 // update Student
74 public function updateStudent() {
75     try {
76         $sql = "UPDATE student SET first_name=:first_name, last_name=:last_name, roll_no=:roll_no, class=:class, age=:age,
77         $data = [
78             'first_name' => $this->_firstName,
79             'last_name' => $this->_lastName,
80             'roll_no' => $this->_rollNo,
81             'class' => $this->_className,
82             'age' => $this->_age,
83             'address' => $this->_address,
84             'status' => 1,
85             'student_id' => $this->_studentID
86         ];
87         $stmt = $this->db->prepare($sql);
88         $stmt->execute($data);
89         $status = $stmt->rowCount();
90         return $status;
91     } catch (Exception $e) {
92         die("Oh noes! There's an error in the query!");
93     }
94 }
95
96 // getAll Student
97 public function getAllStudent() {
98     try {
99         $sql = "SELECT * FROM student";
100         $stmt = $this->db->prepare($sql);
101
102         $stmt->execute();
103         $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
104         return $result;
105     } catch (Exception $e) {
106         die("Oh noes! There's an error in the query!");
107     }
108 }
109
110 // get Student
111 public function getStudent() {
112     try {
113         $sql = "SELECT * FROM student WHERE id=:student_id";
114         $stmt = $this->db->prepare($sql);
115         $data = [
116             'student_id' => $this->_studentID
117         ];
118         $stmt->execute($data);
119         $result = $stmt->fetch(PDO::FETCH_ASSOC);
```

```
120         return $result;
121     } catch (Exception $e) {
122         die("Oh noes! There's an error in the query!");
123     }
124 }
125
126 // delete Student
127 public function deleteStudent() {
128     try {
129         $sql = "DELETE FROM student WHERE id=:student_id";
130         $stmt = $this->db->prepare($sql);
131         $data = [
132             'student_id' => $this->_studentID
133         ];
134         $stmt->execute($data);
135         $status = $stmt->rowCount();
136         return $status;
137     } catch (Exception $e) {
138         die("Oh noes! There's an error in the query!");
139     }
140 }
141
142 }
143 }
144 ?>
```

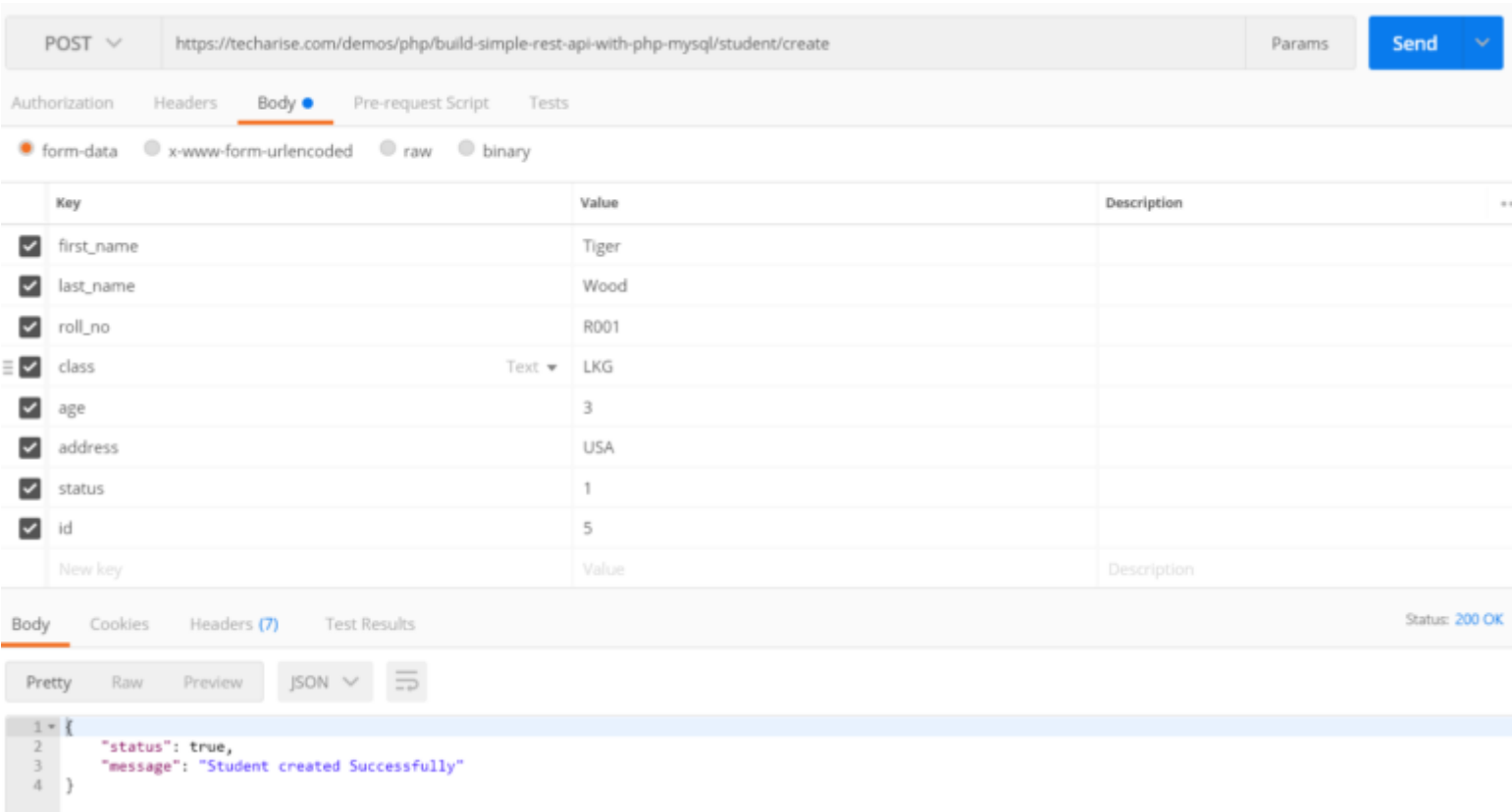
Step 4: Add Student Record — POST Method

Create PHP file named `student/create.php` to insert student records to MySQL database. We will check for POST HTTP request and call method `createStudent()` to insert student data to MySQL database table and return JSON data

```
1 <?php
2 $requestMethod = $_SERVER["REQUEST_METHOD"];
3 include('../class/Student.php');
4 $student = new Student();
5 switch($requestMethod) {
6     case 'POST':
7         $first_name = $_POST['first_name'];
8         $last_name = $_POST['last_name'];
9         $roll_no = $_POST['roll_no'];
10        $class = $_POST['class'];
11        $age = $_POST['age'];
12        $address = $_POST['address'];
13        $status = $_POST['status'];
14
15        $student->setFirstName($first_name);
16        $student->setLastName($last_name);
17        $student->setRollNo($roll_no);
18        $student->setClassName($class);
19        $student->setAge($age);
20        $student->setAddress($address);
21        $studentInfo = $student->createStudent();
22
23        if(!empty($studentInfo)) {
24            $js_encode = json_encode(array('status'=>TRUE, 'message'=>'Student created Successfully'), true);
25        } else {
26            $js_encode = json_encode(array('status'=>FALSE, 'message'=>'Student creation failed.'), true);
27        }
28        header('Content-Type: application/json');
29        echo $js_encode;
30        break;
31    default:
32        header("HTTP/1.0 405 Method Not Allowed");
33        break;
34 }
35 ?>
```

In this method `createStudent()` from class `Student.php` , we will insert record into student table.

```
1 <?php
2 // create Student
3 public function createStudent() {
4     try {
5         $sql = 'INSERT INTO student (first_name, last_name, roll_no, class, age, address, status) VALUES (:first_name, :last
6         $data = [
7             'first_name' => $this->_firstName,
8             'last_name' => $this->_lastName,
9             'roll_no' => $this->_rollNo,
10            'class' => $this->_className,
11            'age' => $this->_age,
12            'address' => $this->_address,
13            'status' => 1,
14        ];
15        $stmt = $this->db->prepare($sql);
16        $stmt->execute($data);
17        $status = $stmt->rowCount();
18        return $status;
19    } catch (Exception $e) {
20        die("Oh noes! There's an error in the query!");
21    }
22 }
23
24 }
25 ?>
```



Step 5: Read Student Record from the Database — GET Method

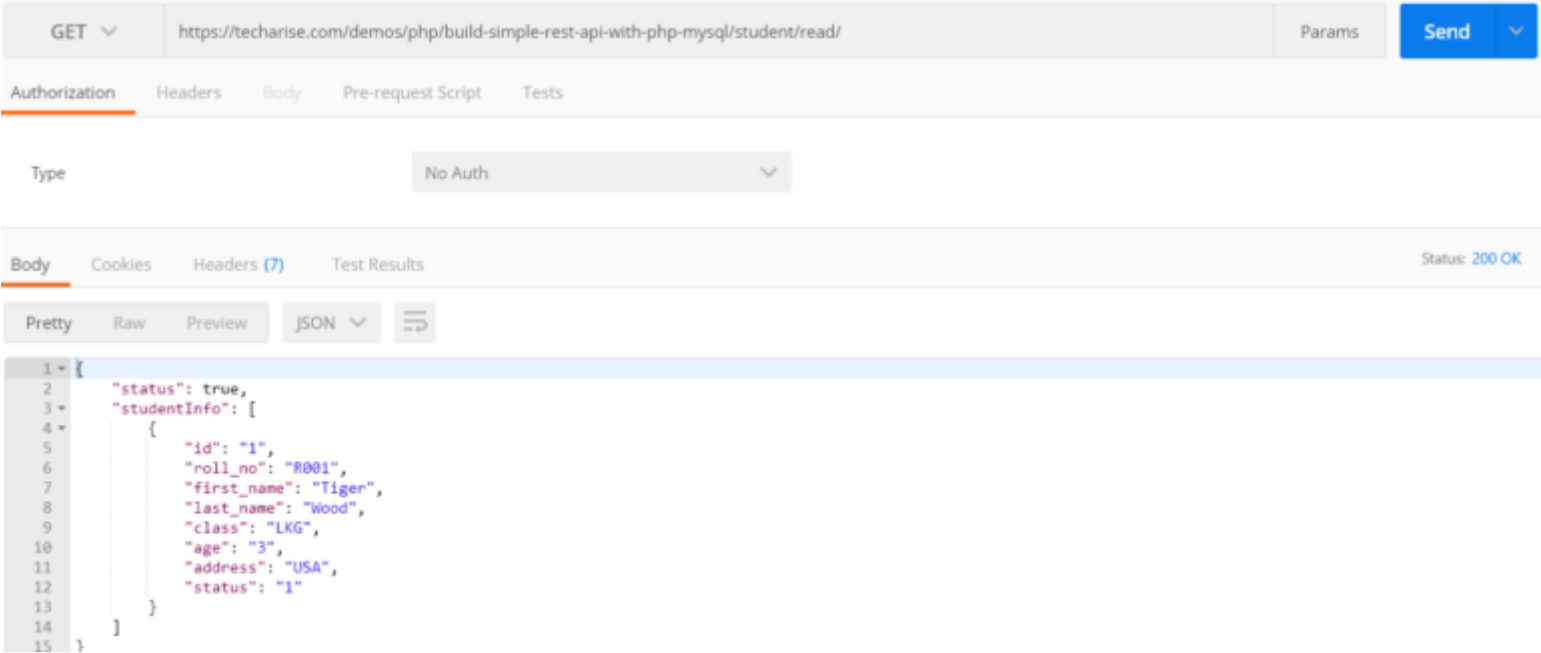
Create PHP file named `student/read.php` to get student records to MySQL database. We will check for GET HTTP request and call method `getAllStudent()` or `getStudent()` to get student data to MySQL database table and return JSON response

```
1 <?php
2 $requestMethod = $_SERVER["REQUEST_METHOD"];
3 include('../class/Student.php');
4 $student = new Student();
5 switch($requestMethod) {
6     case 'GET':
7         $studentID = '';
8         if($_GET['id']) {
9             $studentID = $_GET['id'];
10            $student->setStudentID($studentID);
11            $studentInfo = $student->getStudent();
12        } else {
13            $studentInfo = $student->getAllStudent();
14        }
15        if(!empty($studentInfo)) {
16            $js_encode = json_encode(array('status'=>TRUE, 'studentInfo'=>$studentInfo), true);
17        } else {
18            $js_encode = json_encode(array('status'=>FALSE, 'message'=>'There is no record yet.'), true);
19        }
20        header('Content-Type: application/json');
21        echo $js_encode;
22        break;
23    default:
24        header("HTTP/1.0 405 Method Not Allowed");
25        break;
26 }
27 ?>
```

In this method `getAllStudent()` or `getStudent()` from class `Student.php` , we will get record(s) into student table.

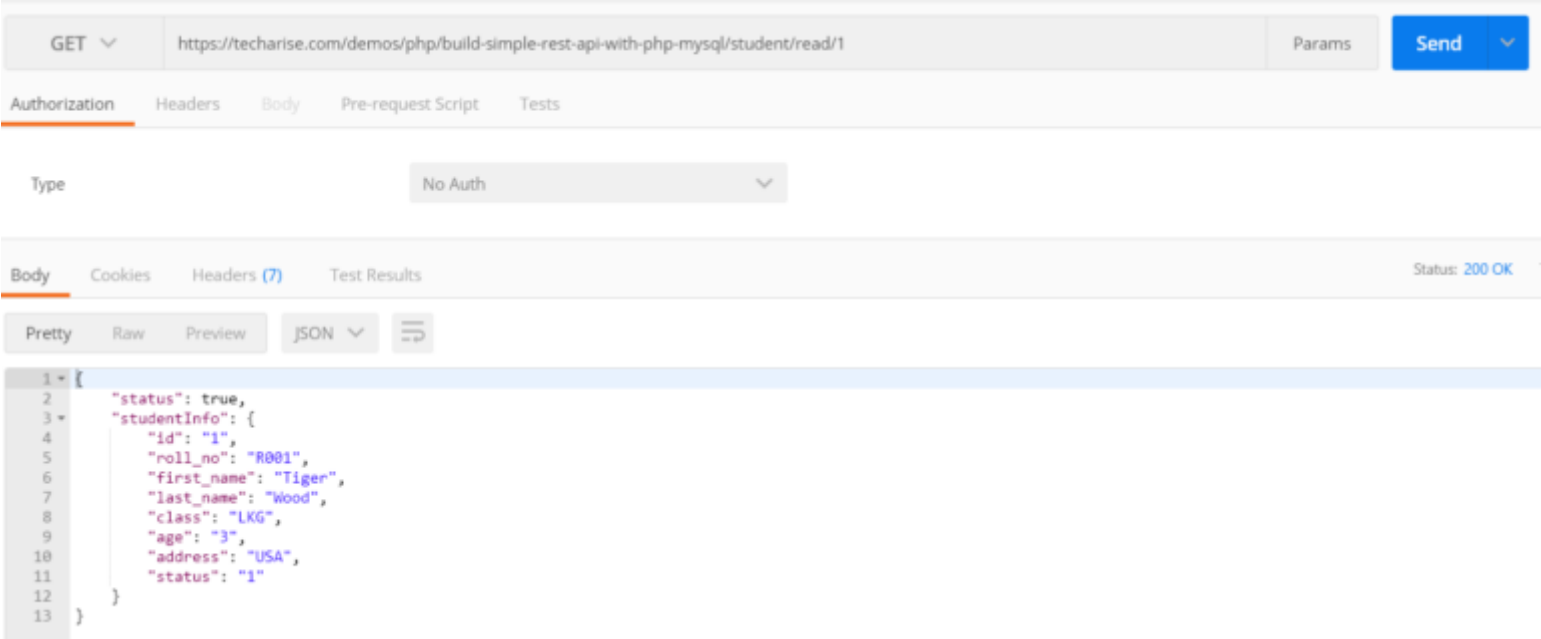
Get All Student Records

```
1 <?php
2 // getAll Student
3 public function getAllStudent() {
4     try {
5         $sql = "SELECT * FROM student";
6         $stmt = $this->db->prepare($sql);
7
8         $stmt->execute();
9         $result = $stmt->fetchAll(\PDO::FETCH_ASSOC);
10        return $result;
11    } catch (Exception $e) {
12        die("Oh noes! There's an error in the query!");
13    }
14 }
15 ?>
```



Get Student single Record

```
1 <?php
2 // get Student
3 public function getStudent() {
4     try {
5         $sql = "SELECT * FROM student WHERE id=:student_id";
6         $stmt = $this->db->prepare($sql);
7         $data = [
8             'student_id' => $this->_studentID
9         ];
10        $stmt->execute($data);
11        $result = $stmt->fetch(\PDO::FETCH_ASSOC);
12        return $result;
13    } catch (Exception $e) {
14        die("Oh noes! There's an error in the query!");
15    }
16 }
17 ?>
```



Step 6: Update Student Record — POST Method

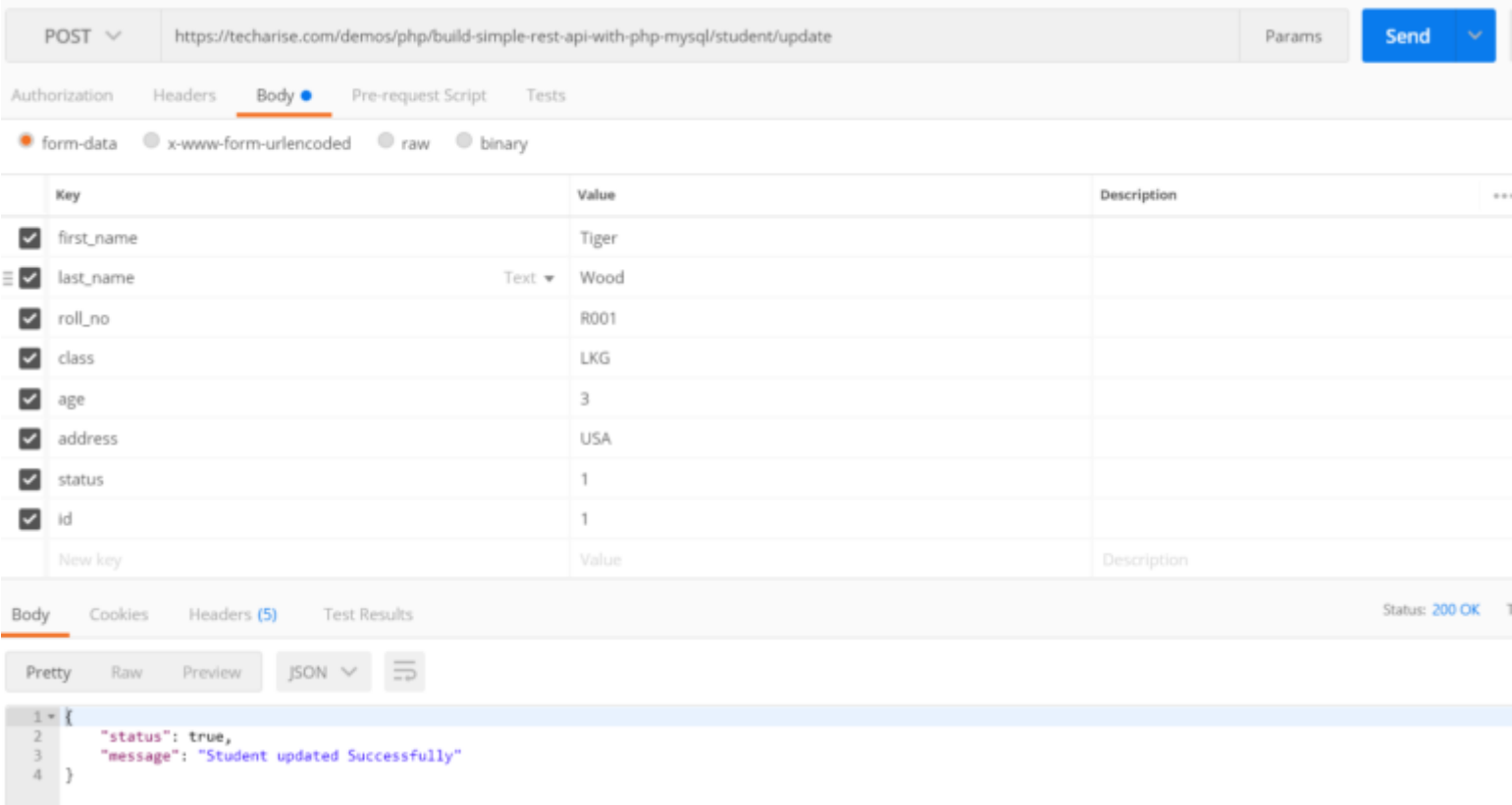
Create PHP file named `student/update.php` to update student records to MySQL database. We will check for POST HTTP request and call method `updateStudent()` to update student data to MySQL database table and return JSON response




```
1 <?php
2 $requestMethod = $_SERVER["REQUEST_METHOD"];
3 include('../class/Student.php');
4 $student = new Student();
5 switch($requestMethod) {
6     case 'POST':
7         $studentID = $_POST['id'];
8         $first_name = $_POST['first_name'];
9         $last_name = $_POST['last_name'];
10        $roll_no = $_POST['roll_no'];
11        $class = $_POST['class'];
12        $age = $_POST['age'];
13        $address = $_POST['address'];
14        $status = $_POST['status'];
15
16        $student->setStudentID($studentID);
17        $student->setFirstName($first_name);
18        $student->setLastName($last_name);
19        $student->setRollNo($roll_no);
20        $student->setClassName($class);
21        $student->setAge($age);
22        $student->setAddress($address);
23        $studentInfo = $student->updateStudent();
24        if(!empty($studentInfo)) {
25            $js_encode = json_encode(array('status'=>TRUE, 'message'=>'Student updated Successfully'), true);
26        } else {
27            $js_encode = json_encode(array('status'=>FALSE, 'message'=>'Student updation failed.'), true);
28        }
29        header('Content-Type: application/json');
30        echo $js_encode;
31    default:
32        header("HTTP/1.0 405 Method Not Allowed");
33        break;
34 }
35 ?>
```

In this method `updateStudent()` from class `Student.php` , we will update record into student table.

```
1 <?php
2 // update Student
3 public function updateStudent() {
4     try {
5         $sql = "UPDATE student SET first_name=:first_name, last_name=:last_name, roll_no=:roll_no, class=:class, age=:age, ad
6         $data = [
7             'first_name' => $this->_firstName,
8             'last_name' => $this->_lastName,
9             'roll_no' => $this->_rollNo,
10            'class' => $this->_className,
11            'age' => $this->_age,
12            'address' => $this->_address,
13            'status' => 1,
14            'student_id' => $this->_studentID
15        ];
16        $stmt = $this->db->prepare($sql);
17        $stmt->execute($data);
18        $status = $stmt->rowCount();
19        return $status;
20    } catch (Exception $e) {
21        die("Oh noes! There's an error in the query!");
22    }
23 }
24 ?>
```



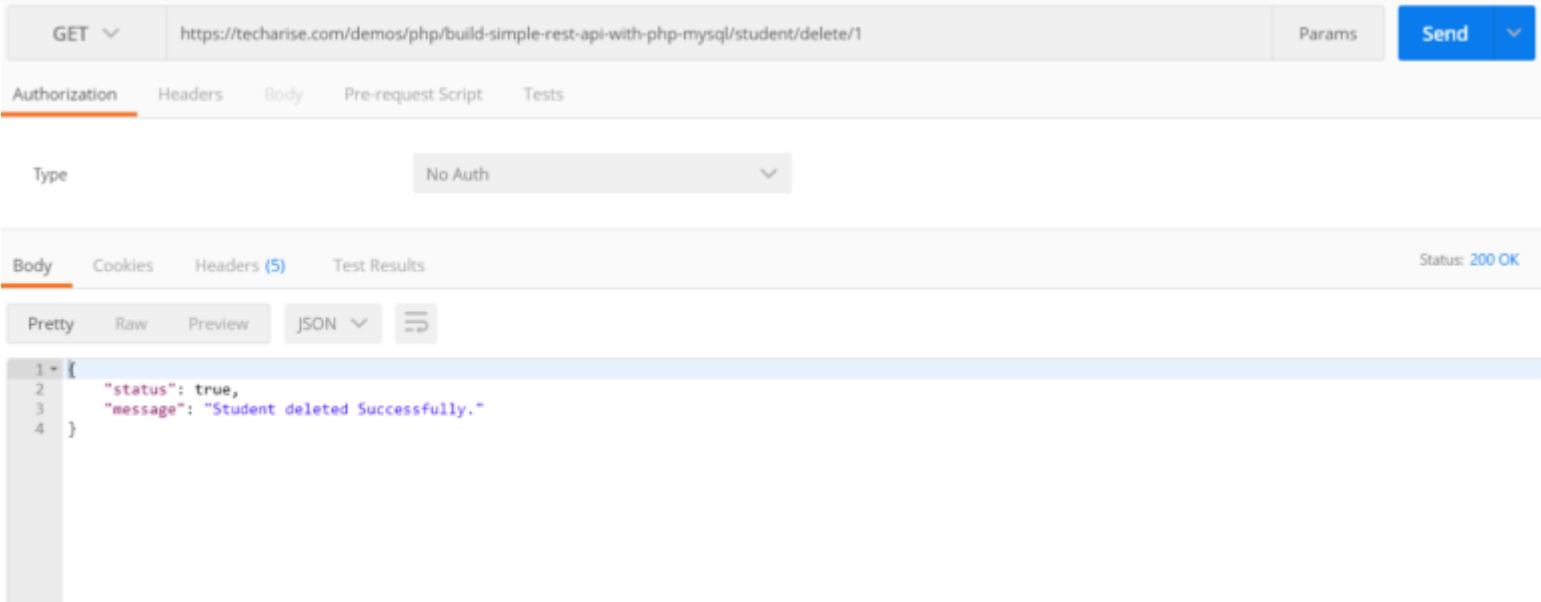
Step 7: Delete Student Record from the Database — GET Method

Create PHP file named `student/delete.php` to delete student record to MySQL database. We will check for GET HTTP request and call method `deleteStudent()` to delete student data to MySQL database table and return JSON response

```
1 <?php
2 $requestMethod = $_SERVER["REQUEST_METHOD"];
3 include('../class/Student.php');
4 $student = new Student();
5 switch($requestMethod) {
6     case 'GET':
7         $empId = '';
8         if($_GET['id']) {
9             $studentID = $_GET['id'];
10            $student->setStudentID($studentID);
11        }
12        $studentInfo = $student->deleteStudent();
13        if(!empty($studentInfo)) {
14            $js_encode = json_encode(array('status'=>TRUE, 'message'=>'Student deleted Successfully.'), true);
15        } else {
16            $js_encode = json_encode(array('status'=>FALSE, 'message'=>'Student delete failed.'), true);
17        }
18        header('Content-Type: application/json');
19        echo $js_encode;
20        break;
21    default:
22        header("HTTP/1.0 405 Method Not Allowed");
23        break;
24 }
25 ?>
```

In this method `deleteStudent()` from class `Student.php` , we will delete record into student table.

```
1 <?php
2 // delete Student
3 public function deleteStudent() {
4     try {
5         $sql = "DELETE FROM student WHERE id=:student_id";
6         $stmt = $this->db->prepare($sql);
7         $data = [
8             'student_id' => $this->_studentID
9         ];
10        $stmt->execute($data);
11        $status = $stmt->rowCount();
12        return $status;
13    } catch (Exception $e) {
14        die("Oh noes! There's an error in the query!");
15    }
16 }
17 ?>
```



Step 8: Create .htaccess Rewrite Rule with PHP for Clean URLs

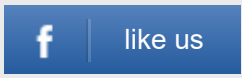
Create student/.htaccess file to write some rule to access rest api with pretty URLs. We will add following rules.

```
1 RewriteEngine On
2 # Turn on the rewriting engine
3 RewriteRule ^read/([0-9a-zA-Z_-]*)$ read.php?id=$1 [NC,L]
4 RewriteRule ^delete/([0-9]*)$ delete.php?id=$1 [NC,L]
5 RewriteRule ^create create.php [NC,L]
6 RewriteRule ^update update.php [NC,L]
```



🔒 This content is locked 🔒

Please support us, use one of the buttons below to unlock the content.



Posted in: [API](#), [PHP](#), [REST API](#), [Tutorials](#) •

Prev

[Build Captcha in CodeIgniter using Captcha Helper](#)

Next

[Integrate the Google reCAPTCHA in PHP Contact Form](#)

Search ...



Latest Articles

[Build a CRUD Operations with Node JS, Express & MongoDB](#)

March 1, 2020

[Build a Newsletter Email Subscription with PHP and MySQL](#)

February 29, 2020

[PayKun Payment Gateway Integration using PHP](#)

February 25, 2020

[Codelobster IDE – Free PHP, HTML, CSS, JavaScript editor](#)

February 19, 2020

[Dynamic TinyMCE WYSIWYG Editor with PHP, MySQL and AJAX](#)

February 8, 2020

[Home](#) | [Contact Us](#) | [About Us](#)

Copyright © 2011 - 2020 [TECHARISE.COM](#) All rights reserved.

Support by [INFOTECHNESIA](#)