Shashank Srivastava

75 Followers About Follow

Create a REST API Using PHP & MySQL

Learn how to create a RESTful web-service with PHP & MySQL in less than five minutes

Shashank Srivastava Jan 10, 2020 · 5 min read

I am writing this tutorial to show you how to create a **RESTful** web-service with **PHP 7 & MySQL 8** in a very short time. You can think of this post as a primer. I'll try to explain the core topics as much as possible while avoiding generic things.

Please note that `mysql` extension for PHP has been completely discontinued in PHP 7. That is why you can only use `mysqli` extension. My entire API is based on `mysqli` extension only.

This tutorial assumes that you know: -

- What an API & REST means.
- What is a REST client.
- What is a RESTful service.
- How to set up a web-server. I am using `Apache/2.4.34`.
- How to install the necessary Apache modules.

Application description

It is a very simple **To-Do** app that is completely based on REST architecture & doesn't have any GUI. You create or retrieve your To-Do's using the REST API only.

I have kept this API extremely simple on purpose. My sole aim here is to get you (& me) started with writing an API & interacting with it. I have skipped database security intentionally & will modify the code later. This tutorial is all about knowing how APIs are created & how you can see REST in action.

Requirements

- PHP 7
- MySQL 8
- REST Client such as Postman. `cURL` or your favorite browser can also be used.

I am using `PHP 7.1.23`, `MySQL 8.0.16` & Postman for this tutorial. You can use any REST client of your choice. I like Postman better as it makes it very easy & convenient to work with REST.

Steps to perform

1. Create a Database & table

Create a database & table in MySQL to store the data.

```
CREATE DATABASE IF NOT EXISTS `my_to_do_db`;
USE my_to_do_db
--
-- Table structure for table `my_to_do_tb`
--

CREATE TABLE IF NOT EXISTS `my_to_do_tb` (
  `task` text NOT NULL,
  `date` text NOT NULL,
  `priority` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

2. Create `config.php` script

This PHP script will store database connection-related information. I have kept database credentials in a separate file `to-do.ini` to avoid hard-coding the password. This file resides in a directory one level above the `DocumentRoot`. You can put this file anywhere. Just make sure that you refer to the correct location in `config.php` script.

```
<?php
$config = parse_ini_file('/Users/admin/Sites/to-do.ini');
$conn = mysqli_connect($config['dbhost'], $config['username'],
$config['password']);
mysqli_select_db($conn, $config['db']);
```

> *Please check my GitHub repository (linked further below) for more information about this file & its location.*

3. Create a PHP script `add-to-do.php` to add To-Do's

Now write a PHP script that will put the data in the MySQL table. The idea here is to take three values viz `task`, `date` & `priority` as payload & `POST` it to the database.

```
<?php
include_once('config.php');
if($_SERVER['REQUEST_METHOD'] == "POST"){
        // Get data from the REST client
        $task = isset($_POST['task']) ?
mysqli_real_escape_string($conn, $_POST['task']) : "";
        $date = isset($_POST['date']) ?
mysqli_real_escape_string($conn, $_POST['date']) : "";
        $priority = isset($_POST['priority']) ?
mysqli_real_escape_string($conn, $_POST['priority']) : "";
        // Insert data into database
        $sql = "INSERT INTO `my_to_do_db`.`my_to_do_tb` (`task`,
`date`, `priority`) VALUES ('$task', '$date', '$priority');";
        $post_data_query = mysqli_query($conn, $sql);
        if($post_data_query){
                $json = array("status" => 1, "Success" => "To-Do
has been added successfully!");
        }
        else{
                $json = array("status" => 0, "Error" => "Error
adding To-Do! Please try again!");
        }
}
else{
        $json = array("status" => 0, "Info" => "Request method not
accepted!");
}
@mysqli_close($conn);
// Set Content-type to JSON
header('Content-type: application/json');
echo json_encode($json);
```

This script `GET` s the data from the MySQL database using `task` as the request query parameter. In other words, this script allows us to fetch a To-Do from the list using a To-Do task.

For example, let's assume we have a To-Do whose task name is **Write Code**. Now you can retrieve its information by using **Write Code** as a query string. More on it is covered later in this post.

```
<?php
        include_once('config.php');
        $task = isset($_GET['task']) ?
mysqli_real_escape_string($conn, $_GET['task']) :  "";
        $sql = "SELECT * FROM `my_to_do_db`.`my_to_do_tb` WHERE
task='{$task}';";
        $get_data_query = mysqli_query($conn, $sql) or
die(mysqli_error($conn));
                if(mysqli_num_rows($get_data_query)!=0){
                $result = array();

                while($r = mysqli_fetch_array($get_data_query)){
                        extract($r);
                        $result[] = array("Task" => $task, "Date"
=> $date, 'Priority' => $priority);
                }
                $json = array("status" => 1, "info" => $result);
        }
        else{
                $json = array("status" => 0, "error" => "To-Do not
found!");
        }
@mysqli_close($conn);

// Set Content-type to JSON
header('Content-type: application/json');
echo json_encode($json);
```

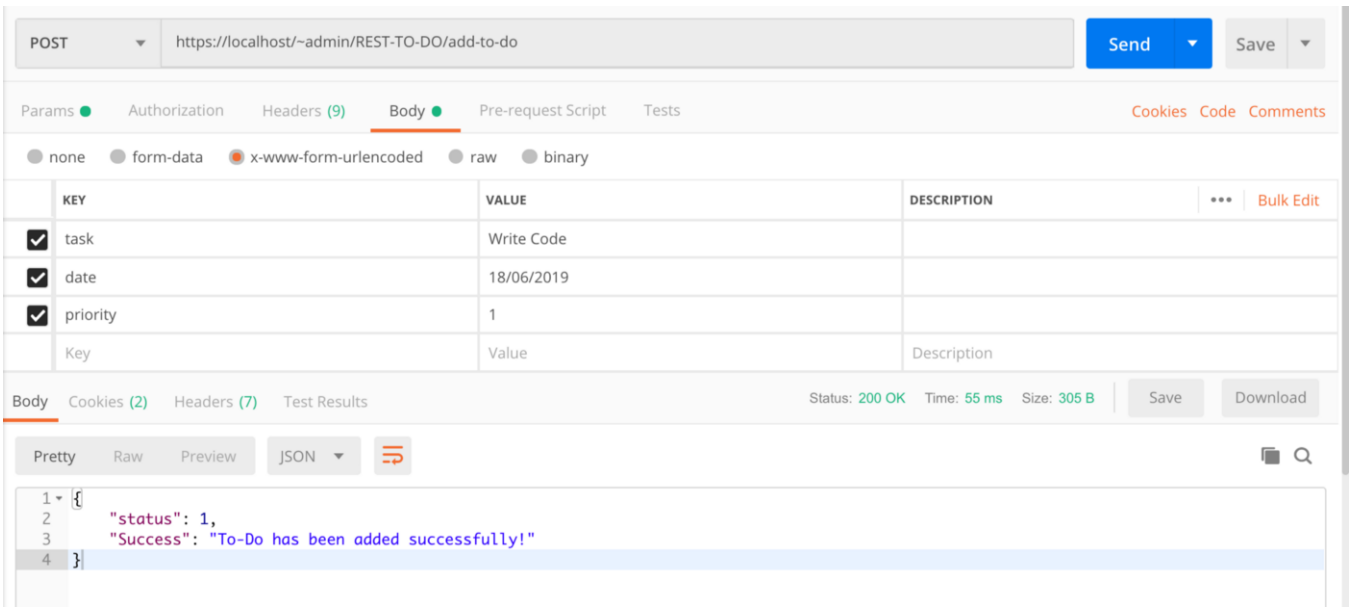
We're now done with the coding. Time to see our API in action.

Try creating a To-Do

Now that you have written the code, it is time to test the API. For this, open your favorite REST client & send a `POST` call like below. You need to make sure you choose

REST endpoint

```
{
    https://localhost/~admin/REST-TO-DO/add-to-do
}
```



If you have followed the steps correctly, you'll see that your To-Do has been added to the database successfully! You should see an output similar to the below.

```
{
    "status": 1,
    "Success": "To-Do has been added successfully!"
}
```

You can grab the entire code from my GitHub Repository. It also has the `SQL` script to create the database & its table. You can also click the link below to go to the repository.

shashank-ssriva/REST-To-Do

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or...

github.com

Fetch To-Do information

To retrieve a To-Do's information, invoke a `GET` call against the below REST endpoint. Note how I am using `?task=Write Code` as the request query parameter.

REST endpoint

```
{
    https://localhost/~admin/REST-TO-DO/info?task=Write Code
}
```

Output

```
{
    "status": 1,
    "info": [
        {
            "Task": "Write Code",
            "Date": "18/06/2019",
            "Priority": "1"
        }
    ]
}
```

You have now reached the end of this tutorial. I hope it was informative & helpful. In my next post, I will show you how you can add **Basic Authentication** to this API so that only authenticated users are able to add/fetch the To-Do's.

I'll also demonstrate how you can enable HTTPS for this API.

Thanks for reading this post!

PHP Rest Api MySQL