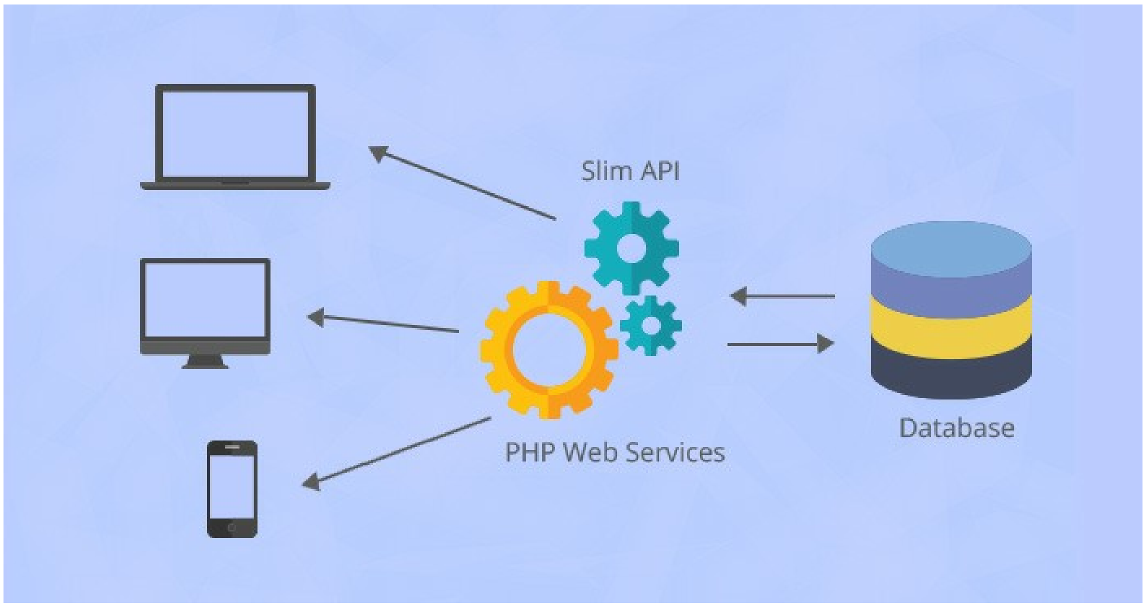
The internet is chock full of third-party and custom APIs that offer a wide range of functionalities. PHP offers several frameworks for web app development rapidly. However, time is always the enemy of web developers and the app needs to be pushed at an impossible deadline. In such times, frameworks are the best option for rapid application development.



In this tutorial, I will introduce you to Slim framework for PHP. Slim is fast becoming the most-opted framework for API development and small web apps. Although you can create

REST API in several other frameworks like CakePHP, Symfony Laravel, Codeigniter, they have a steep learning curve and are often too cumbersome to use in rapid development scenarios.

Understanding Slim Framework

Slim is super lightweight framework, ideal for rapid web app development. One of the important usages is in REST API development. Slim supports all HTTP method (GET,POST,PUT,DELETE). Slim contains very handy URL structure with routers, middlewares, bodyparser along with page templates, flash messages, encrypted cookies and lots more.

At this point, it is important to understand the structure of the REST API.

Related: Creating Twig Templates in Slim Microframework

Understanding REST API

REST is the abbreviation of Representational State Transfer. This is a bridge or medium between data resource and application interface, whether it's on mobile devices or desktops. 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	is used for updating data.
DELETE	is used for deleting data.

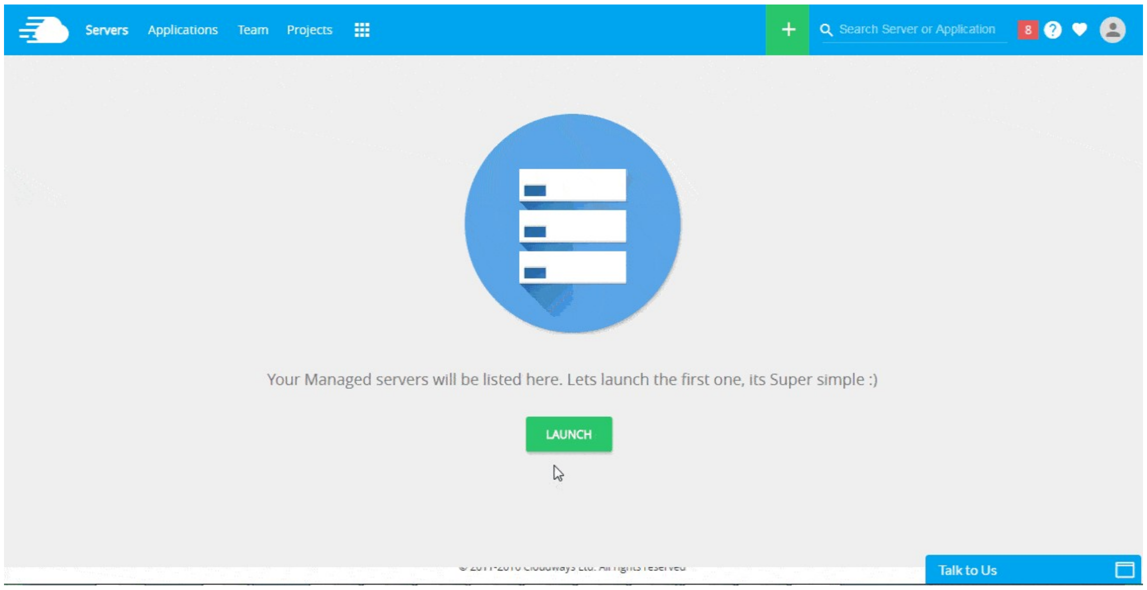
Basically, REST phenomena works on actions and resources. Whenever any action URL is invoked, it performs an individual method (or a set of methods) on that URL. I will further discuss this below with examples.

First we will need to install Slim framework for the REST API project.

I assume that you already have your Cloudways server launched with PHPstack and if you didn't launch your server signup to get it.

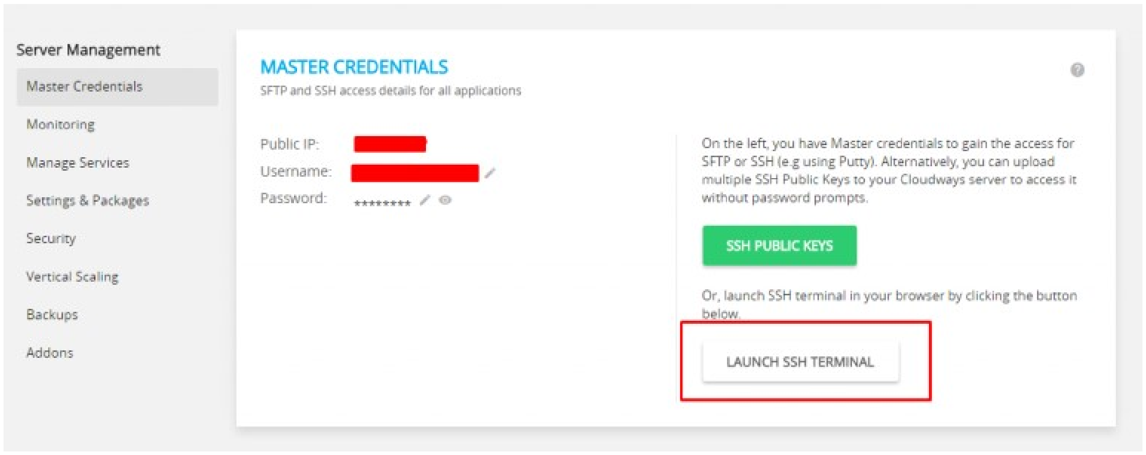
PHP Hosting: Best PHP 7 & PHP 5.6 Web Hosting

*(Note: You may use promo code: **PHP15** to get **FREE** Cloudways hosting credit of $15 on signup.)*



Your Managed servers will be listed here. Lets launch the first one, its Super simple :)

After creating the server launch SSH terminal.



MASTER CREDENTIALS

SFTP and SSH access details for all applications

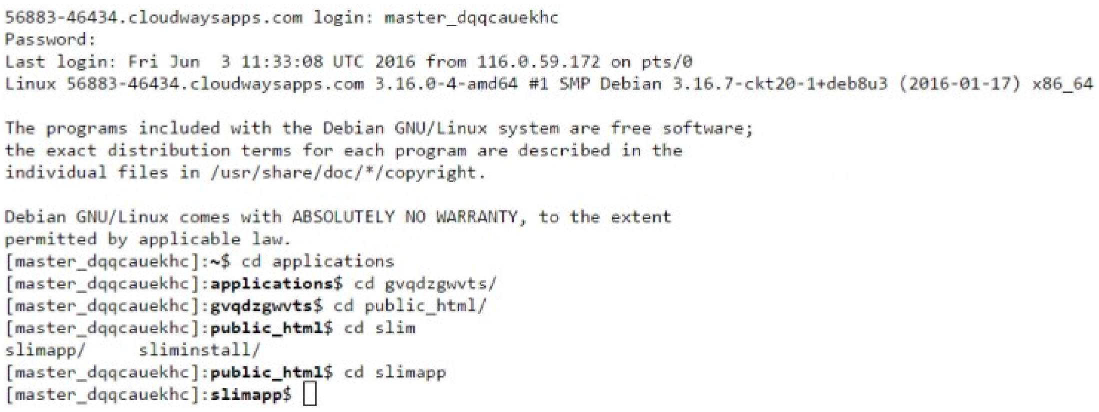
Public IP: [redacted]
Username: [redacted]
Password: ********

On the left, you have Master credentials to gain the access for SFTP or SSH (e.g using Putty). Alternatively, you can upload multiple SSH Public Keys to your Cloudways server to access it without password prompts.

Or, launch SSH terminal in your browser by clicking the button below.

Step 1: Install Slim Framework From Composer

Open SSH terminal from the Cloudways panel and and sign in with your username and password. Now go to the folder where you want to install SLIM with cd command

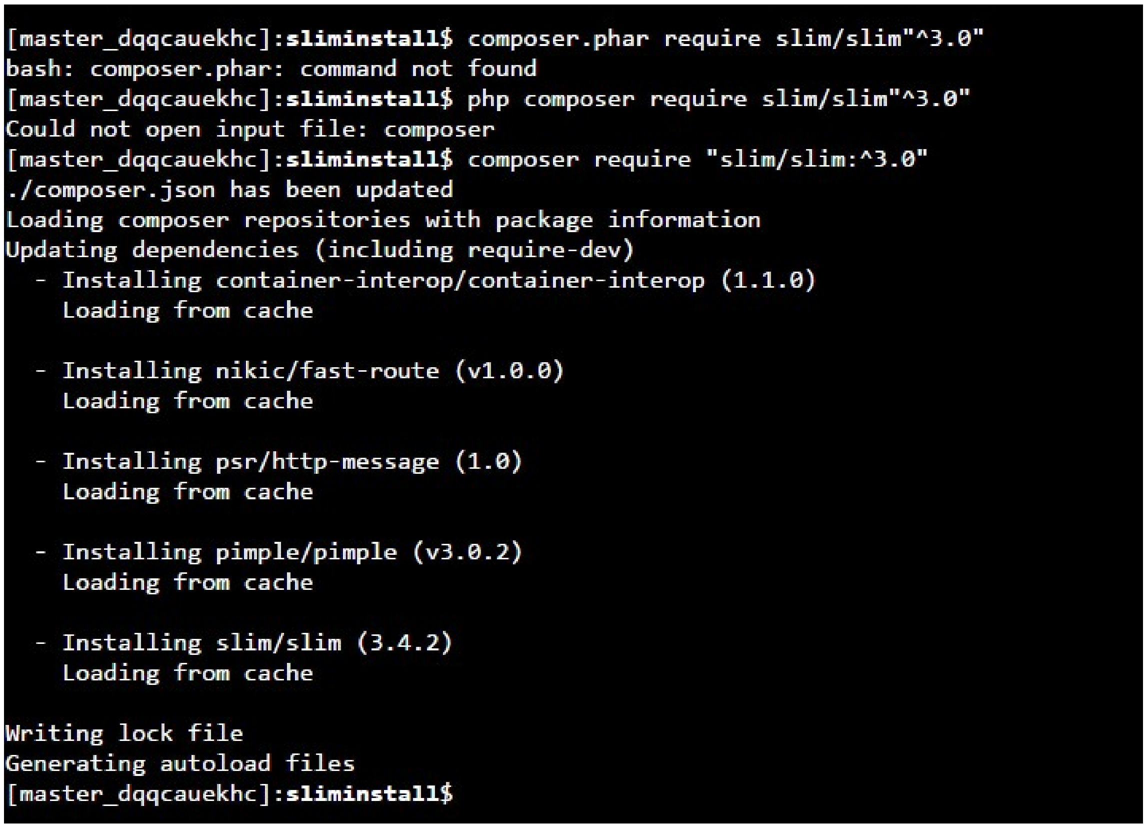


```
56883-46434.cloudwaysapps.com login: master_dqqcauekhc
Password:
Last login: Fri Jun  3 11:33:08 UTC 2016 from 116.0.59.172 on pts/0
Linux 56883-46434.cloudwaysapps.com 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt20-1+deb8u3 (2016-01-17) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[master_dqqcauekhc]:~$ cd applications
[master_dqqcauekhc]:applications$ cd gvqdzgwvts/
[master_dqqcauekhc]:gvqdzgwvts$ cd public_html/
[master_dqqcauekhc]:public_html$ cd slim
slimapp/     sliminstall/
[master_dqqcauekhc]:public_html$ cd slimapp
[master_dqqcauekhc]:slimapp$
```

Input the following command in the terminal to install Slim via composer.

```
composer require slim/slim"^3.0"
composer require slim/slim"^3.0"
```

```
composer require slim/slim"^3.0"
```



```
[master_dqqcauekhc]:sliminstall$ composer.phar require slim/slim"^3.0"
bash: composer.phar: command not found
[master_dqqcauekhc]:sliminstall$ php composer require slim/slim"^3.0"
Could not open input file: composer
[master_dqqcauekhc]:sliminstall$ composer require "slim/slim:^3.0"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
  - Installing container-interop/container-interop (1.1.0)
    Loading from cache

  - Installing nikic/fast-route (v1.0.0)
    Loading from cache

  - Installing psr/http-message (1.0)
    Loading from cache

  - Installing pimple/pimple (v3.0.2)
    Loading from cache

  - Installing slim/slim (3.4.2)
    Loading from cache

Writing lock file
Generating autoload files
[master_dqqcauekhc]:sliminstall$
```

After installing Slim, the following piece of code will require it in the index.php file to require autoload file and instantiate Slim.

```
<?php require 'vendor/autoload.php'; $app = new Slim\App();
```

```
<?php

require 'vendor/autoload.php';
$app = new Slim\App();
```

```
<?php require 'vendor/autoload.php'; $app = new Slim\App();
```

Composer comes pre-installed on Cloudways servers. If you are working on the localhost, you need to install it. If you haven't installed it yet, just go to the following link and follow the instructions.

Step 2: Making a .htaccess File for Clean URL Structure

To make your life easier, you should create a .htaccess file that defines clean URL structure. At the root directory, make a .htaccess file and add the below code in it. This will provide a clean URL structure for the PHP file. (this just means that you don't want to include PHP filename in the URL calls).

```
RewriteEngine On RewriteCond %{Request_Filename} !-F RewriteCond %{Request_Filename} !-
d RewriteRule ^ index.php [QSA,L]
```

```
RewriteEngine On

RewriteCond %{Request_Filename} !-F

RewriteCond %{Request_Filename} !-d

RewriteRule ^ index.php [QSA,L]
```

```
RewriteEngine On RewriteCond %{Request_Filename} !-F RewriteCond %{Request_Filename} !-
d RewriteRule ^ index.php [QSA,L]
```

If your index file is located in different folder (for instance, the “public” folder), then you can insert the full path of the index file in the last line:

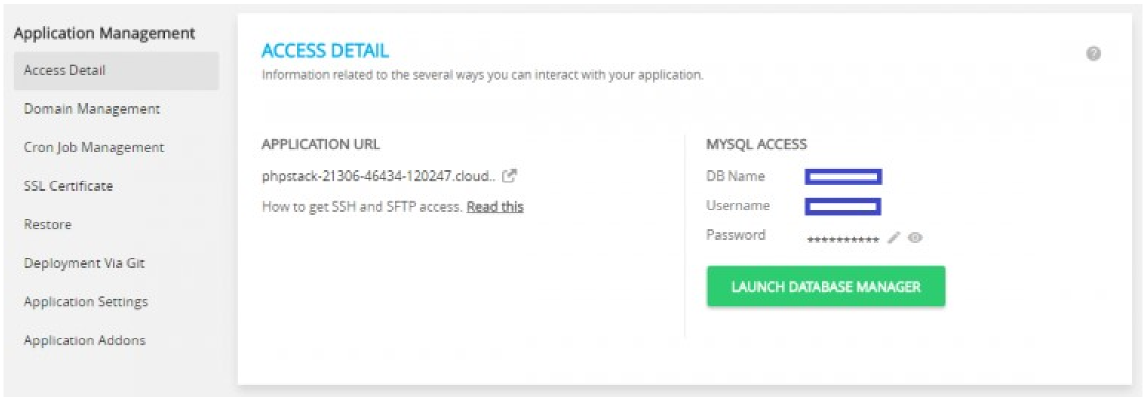
```
RewriteRule ^ public/index.php [QSA,L]
```

```
RewriteRule ^ public/index.php [QSA,L]
```

```
RewriteRule ^ public/index.php [QSA,L]
```

Step 3: Create a Database in MySQL

With each PHP Stack on Cloudways, you get an empty database.



ACCESS DETAIL

Information related to the several ways you can interact with your application.

APPLICATION URL

phpstack-21306-46434-120247.cloud..

How to get SSH and SFTP access. Read this

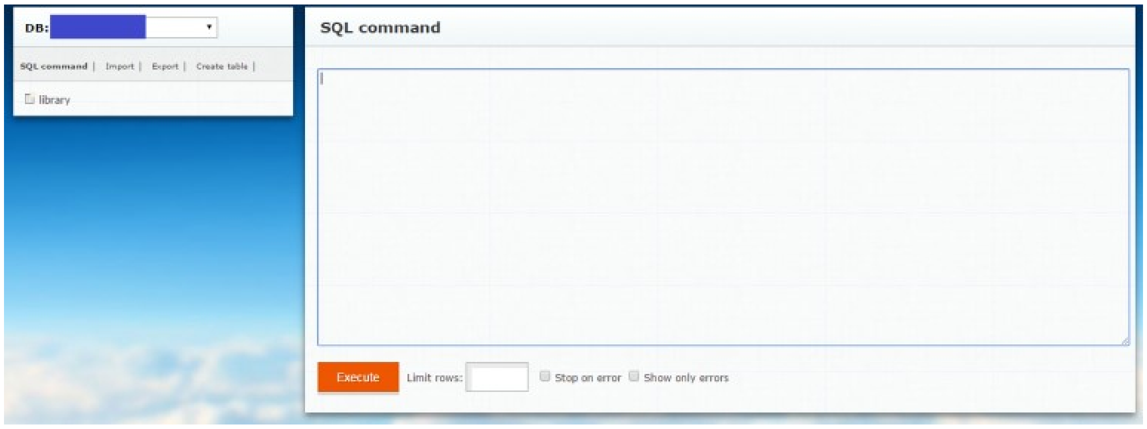
MYSQL ACCESS

DB Name [redacted]

Username [redacted]

Password ***********

Click on **Launch Database Manager.** To create the requisite tables, run the following query in SQL Command box:



```
CREATE TABLE IF NOT EXISTS `library` (  `book_id` int(11) NOT NULL,  `book_name` varcha
r(100) NOT NULL,  `book_isbn` varchar(100) NOT NULL,  `book_category` varchar(100) NOT
NULL ) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8; INSERT INTO `library` (`boo
k_id`, `book_name`, `book_isbn`, `book_category`) VALUES (1, 'PHP', 'bk001', 'Server Si
de'), (3, 'javascript', 'bk002', 'Client Side'), (4, 'Python', 'bk003', 'Data Analysis'
);
```

Now it is time for the first API call. Let's make it systematically.

You might also like: Using Eloquent ORM With Slim

Step 4: Retrieving All Books

Enter the following code in the index.php file to get all the books from the database. A GET call is used for retrieval.

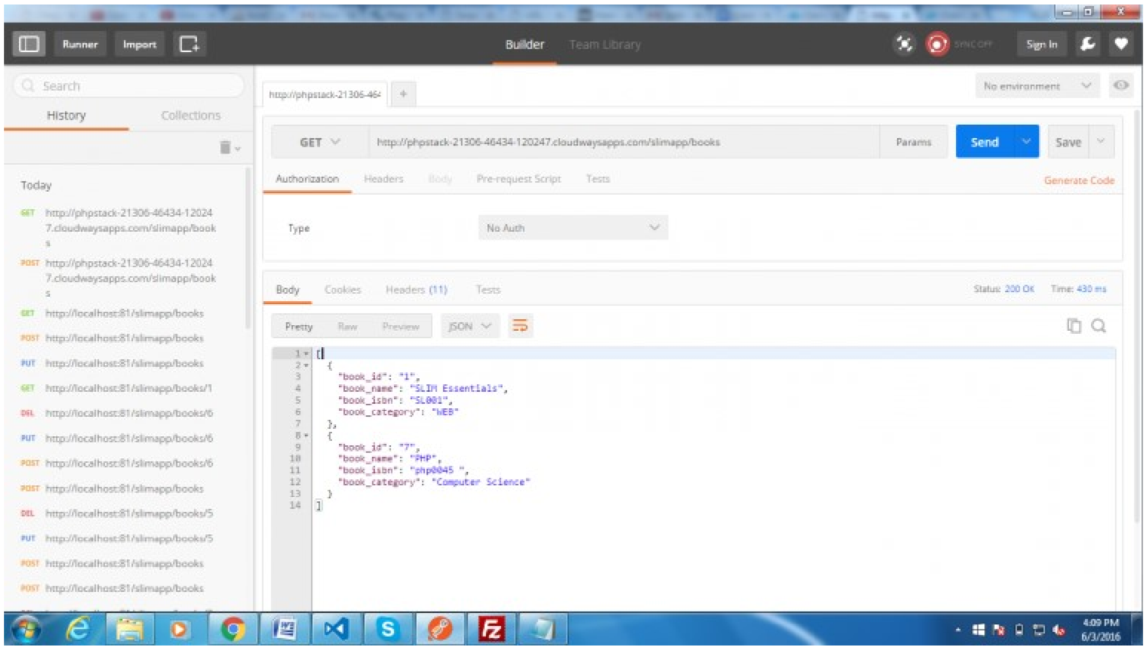
```
$app->get('/books', function() {  require_once('db.php');  $query = "select * from libr
ary order by book_id";  $result = $connection->query($query);  // var_dump($result);  w
hile ($row = $result->fetch_assoc()){ $data[] = $row;  }  echo json_encode($data); });
```

```
$app->get('/books', function() {

 require_once('db.php');

 $query = "select * from library order by book_id";

 $result = $connection->query($query);

 // var_dump($result);

 while ($row = $result->fetch_assoc()){

$data[] = $row;

 }

 echo json_encode($data);

});
```

```
$app->get('/books', function() { require_once('db.php'); $query = "select * from librar
y order by book_id"; $result = $connection->query($query); // var_dump($result); while
($row = $result->fetch_assoc()){ $data[] = $row; } echo json_encode($data); });
```

To streamline working with the API calls, I recommend using Postman (available from the Chrome App Store). This plugin greatly helps in API management and usage.

In postman, make a GET call with API URL.



GET http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books

Type: No Auth

Status: 200 OK Time: 430 ms

History:

```
GET  http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books
POST http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books
GET  http://localhost:81/slimapp/books
POST http://localhost:81/slimapp/books
PUT  http://localhost:81/slimapp/books
GET  http://localhost:81/slimapp/books/1
DEL  http://localhost:81/slimapp/books/6
PUT  http://localhost:81/slimapp/books/6
POST http://localhost:81/slimapp/books/6
POST http://localhost:81/slimapp/books
DEL  http://localhost:81/slimapp/books/5
PUT  http://localhost:81/slimapp/books/5
POST http://localhost:81/slimapp/books
POST http://localhost:81/slimapp/books
```

```
[
  {
    "book_id": "1",
    "book_name": "SLIM Essentials",
    "book_isbn": "SL001",
    "book_category": "WEB"
  },
  {
    "book_id": "7",
    "book_name": "PHP",
    "book_isbn": "php0045 ",
    "book_category": "Computer Science"
  }
]
```

Step 5: Creating a Book's Record

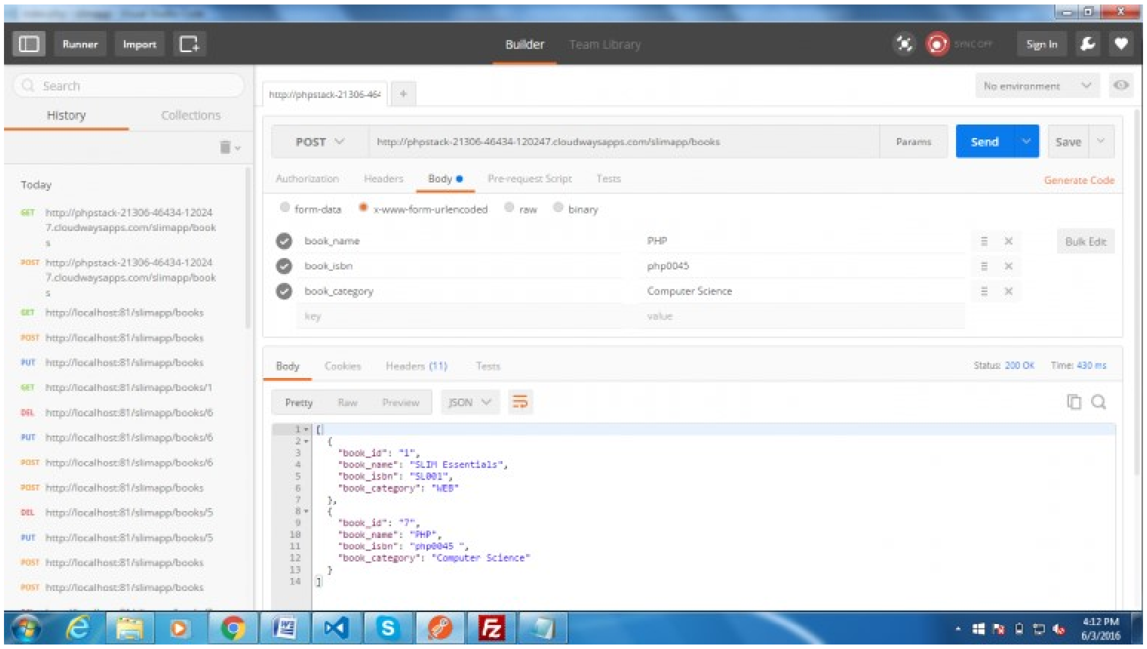
Make a new API call in the index.php through the following code:

```
$app->post('/books', function($request){  require_once('db.php');  $query = "INSERT INT
O library (book_name,book_isbn,book_category) VALUES (?,?,?)";  $stmt = $connection->pr
epare($query);  $stmt->bind_param("sss",$book_name,$book_isbn,$book_category);  $book_n
ame = $request->getParsedBody()['book_name'];  $book_isbn = $request->getParsedBody()['
book_isbn'];  $book_category = $request->getParsedBody()['book_category'];   $stmt->exe
cute(); });
```

```
$app->post('/books', function($request){

 require_once('db.php');

 $query = "INSERT INTO library (book_name,book_isbn,book_category) VALUES
(?,?,?)";

 $stmt = $connection->prepare($query);

 $stmt->bind_param("sss",$book_name,$book_isbn,$book_category);

 $book_name = $request->getParsedBody()['book_name'];

 $book_isbn = $request->getParsedBody()['book_isbn'];

 $book_category = $request->getParsedBody()['book_category'];

  $stmt->execute();
 });
```

```
$app->post('/books', function($request){ require_once('db.php'); $query = "INSERT INTO library (book_name,book_isbn,book_category) VALUES (?,?,?)"; $stmt = $connection->prepare($query); $stmt->bind_param("sss",$book_name,$book_isbn,$book_category); $book_name = $request->getParsedBody()['book_name']; $book_isbn = $request->getParsedBody()['book_isbn']; $book_category = $request->getParsedBody()['book_category']; $stmt->execute(); });
```

Open Postman and click **Body.** Select **x.www-form-urlencoded.** Now add records to insert via POST call.



```
Today
GET  http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books
POST http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books
GET  http://localhost:81/slimapp/books
POST http://localhost:81/slimapp/books
PUT  http://localhost:81/slimapp/books
GET  http://localhost:81/slimapp/books/1
DEL  http://localhost:81/slimapp/books/6
PUT  http://localhost:81/slimapp/books/6
POST http://localhost:81/slimapp/books/6
POST http://localhost:81/slimapp/books
DEL  http://localhost:81/slimapp/books/5
PUT  http://localhost:81/slimapp/books/5
POST http://localhost:81/slimapp/books
POST http://localhost:81/slimapp/books
```

POST http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books

key	value
book_name	PHP
book_isbn	php0045
book_category	Computer Science

Status: 200 OK Time: 430 ms

```
[
  {
    "book_id": "1",
    "book_name": "SLIM Essentials",
    "book_isbn": "SL001",
    "book_category": "WEB"
  },
  {
    "book_id": "7",
    "book_name": "PHP",
    "book_isbn": "php0045 ",
    "book_category": "Computer Science"
  }
]
```

Step 6: Updating a Book's Record

Make a new API call like below to update a record in the database.

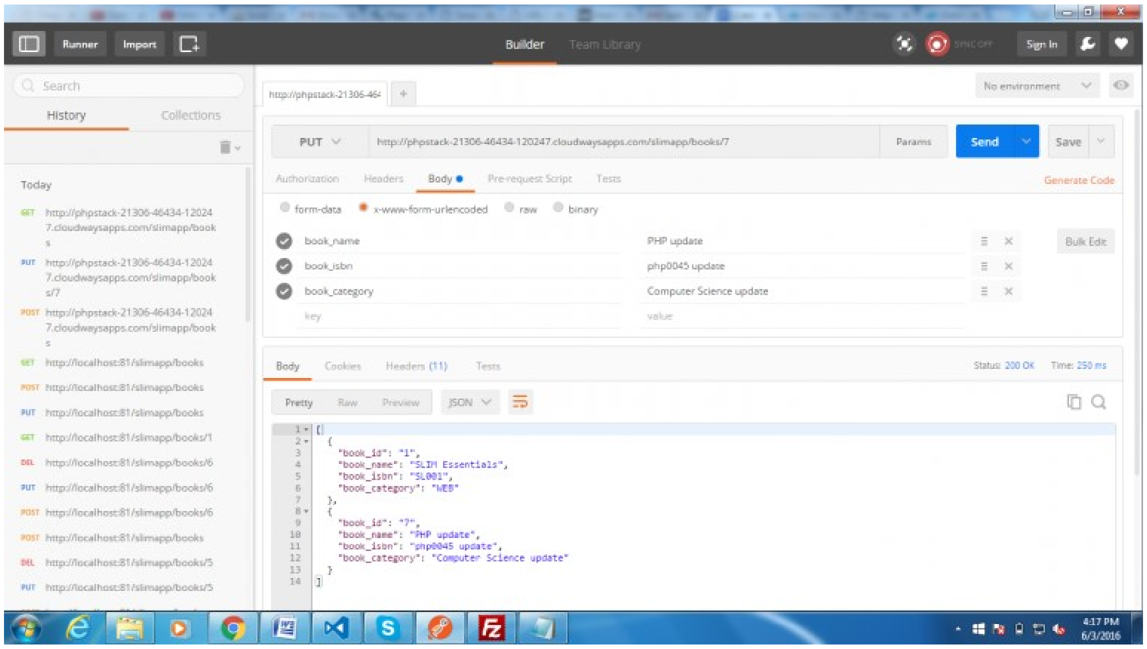
```
$app->put('/books/{book_id}', function($request){  require_once('db.php');  $get_id = $request->getAttribute('book_id');  $query = "UPDATE library SET book_name = ?, book_isbn = ?, book_category = ? WHERE book_id = $get_id";  $stmt = $connection->prepare($query);  $stmt->bind_param("sss",$book_name,$book_isbn,$book_category);  $book_name = $request->getParsedBody()['book_name'];  $book_isbn = $request->getParsedBody()['book_isbn']
 $book_category = $request->getParsedBody()['book_category'];  $stmt->execute(); });
```

```
$app->put('/books/{book_id}', function($request){

 require_once('db.php');

 $get_id = $request->getAttribute('book_id');

 $query = "UPDATE library SET book_name = ?, book_isbn = ?, book_category = ?
WHERE book_id = $get_id";

 $stmt = $connection->prepare($query);

 $stmt->bind_param("sss",$book_name,$book_isbn,$book_category);
```

```php

 $book_name = $request->getParsedBody()['book_name'];

 $book_isbn = $request->getParsedBody()['book_isbn'];

 $book_category = $request->getParsedBody()['book_category'];

 $stmt->execute();

});
```

```php
$app->put('/books/{book_id}', function($request){ require_once('db.php'); $get_id = $request->getAttribute('book_id'); $query = "UPDATE library SET book_name = ?, book_isbn = ?, book_category = ? WHERE book_id = $get_id"; $stmt = $connection->prepare($query); $stmt->bind_param("sss",$book_name,$book_isbn,$book_category); $book_name = $request->getParsedBody()['book_name']; $book_isbn = $request->getParsedBody()['book_isbn']; $book_category = $request->getParsedBody()['book_category']; $stmt->execute(); });
```

In Postman, add data to update a specific book record.



Today

GET http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books
PUT http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books/7
POST http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books
GET http://localhost:81/slimapp/books
POST http://localhost:81/slimapp/books
PUT http://localhost:81/slimapp/books
GET http://localhost:81/slimapp/books/1
DEL http://localhost:81/slimapp/books/6
PUT http://localhost:81/slimapp/books/6
POST http://localhost:81/slimapp/books/6
POST http://localhost:81/slimapp/books
DEL http://localhost:81/slimapp/books/5
PUT http://localhost:81/slimapp/books/5

PUT http://phpstack-21306-46434-120247.cloudwaysapps.com/slimapp/books/7

book_name	PHP update
book_isbn	php0045 update
book_category	Computer Science update
key	value

Status: 200 OK Time: 250 ms

```json
[
  {
    "book_id": "1",
    "book_name": "SLIM Essentials",
    "book_isbn": "SL001",
    "book_category": "WEB"
  },
  {
    "book_id": "7",
    "book_name": "PHP update",
    "book_isbn": "php0045 update",
    "book_category": "Computer Science update"
  }
]
```

Step 7: Deleting a Book's Record

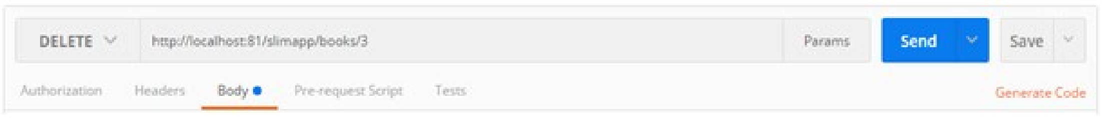
To delete a record with a specific ID, a DELETE call is required.

```php
$app->delete('/books/{book_id}', function($request){  require_once('db.php');  $get_id = $request->getAttribute('book_id');  $query = "DELETE from library WHERE book_id = $get_id";  $result = $connection->query($query); });
```

```php
$app->delete('/books/{book_id}', function($request){

 require_once('db.php');

 $get_id = $request->getAttribute('book_id');

 $query = "DELETE from library WHERE book_id = $get_id";

 $result = $connection->query($query);
});
```

```
$app->delete('/books/{book_id}', function($request){ require_once('db.php'); $get_id = $request->getAttribute('book_id'); $query = "DELETE from library WHERE book_id = $get_id"; $result = $connection->query($query); });
```

On Postman, run the call like this



DELETE http://localhost:81/slimapp/books/3

This is all for the basic REST API in the Slim Framework. However, this API will not work until you add this command at the end of the code.

```
$app->run();
```

```
$app->run();
```

```
$app->run();
```

Conclusion

Creating and using the REST API with Slim framework is very easy. The biggest advantage of the framework is its ease of use and lightweight. The icing on the cake is that it is very easy to learn and a good developer could pick up the framework in a matter of hour. To summarize, Slim receives HTTP requests, review them and invokes the appropriate callback routine for HTTP requests and return the appropriate response(s).

If you need clarification about this article or have any other query about the Slim Framework, do let me know through the comment section.

Share your opinion in the comment section. COMMENT NOW

Shahroze Nawaz

Shahroze is a PHP Community Manager at Cloudways - A Managed PHP Hosting Platform. Besides his work life, he loves movies and travelling. You can email him at shahroze.nawaz@cloudways.com

Get Connected on: Twitter Community Forum

THERE'S MORE TO READ.

PHP | Magento 2 | Magento | 4 Min Read

How To Configure Magento 2 With PHP 7 On...

Fayyaz Khattak
Published on 22nd March

Database | PHP | 7 Min Read

Exclusive MySQL Performance Tuning Tips For Better Database Optimization

Shahroze Nawaz
Published on 19th March

PHP | 7 Min Read

Perform PHP Debugging By Integrating Xdebug With VsCode

Shahroze Nawaz
Published on 18th March

PHP | 6 Min Read

How to Host PHP on Amazon AWS EC2

Ahmed Khan
Published on 13th March

PHP | 10 Min Read

Budget Friendly Yet More Powerful – Learn How to...

Shahroze Nawaz
Published on 13th February

PHP | 2 Min Read

PHP Version History: Brief Timeline of World's Most Used...

Shahroze Nawaz
Published on 4th January

PRODUCT & SOLUTION

WordPress Hosting	WooCommerce Ho...
Magento Hosting	Cloudways Platform
PHP Cloud Hosting	Cloudways API
Laravel Hosting	Breeze – Free Wor...
Drupal Hosting	Add-ons
Joomla Hosting	CloudwaysCDN
PrestaShop Hosting	CloudwaysBot

COMPANY

About us
Testimonials
Terms
Media Kit
Sitemap

SUPPORT

Knowledge base
Contact us
Blog
Community
Feedback
Free Website Migration

QUICK LINKS

Features
Pricing
Partners
Cloud Affiliate...

CLOUDWAYS

52 Springvale, Pope Pius XII Street Mosta MST2653, Malta

Follow Us On