

APPLICATION INTERFACE

Things required to set up an api

- **Postman** : It is an extension which is found on the browser. It is used to test your api calls.

How do you install Postman

Assuming, you are using chrome, go to apps... Locate web store and install postman.

- **Slim**: It is a rest api framework which is used for routing of api calls.

How do you install Slim

Go to <https://github.com/slimphp/Slim/releases> to download the zip file. You can visit <http://www.slimframework.com/> to understand more.

Don't panic I will work you through everything step by step

Lets take a simple project.. Build an api for users that own houses.. i.e. each user is attached to a house with its description.

Starting with the query aspect. A user has a name, password, username, sex, houses, house description, house colour, house location and many more about houses.

To avoid data redundancy, its better to have a user table and a house table. So how does a user claim a house in house table.

1. Create a Database

Database Schema

CREATE DATABASE IF NOT EXISTS apitest;

USE apitest;

CREATE TABLE IF NOT EXISTS users(

user_id INT UNSIGNED AUTO_INCREMENT NOT NULL ,
username VARCHAR (40) NOT NULL ,
password VARCHAR (100) NOT NULL ,
created_at DATETIME NOT NULL,
modified_at DATETIME NOT NULL ,
active_status BOOLEAN DEFAULT 1,
PRIMARY KEY (user_id)
);

CREATE TABLE IF NOT EXISTS houses(

house_id INT UNSIGNED AUTO_INCREMENT NOT NULL ,
user_id INT UNSIGNED NOT NULL ,
house_name VARCHAR (40) NOT NULL ,
house_location VARCHAR (40) NOT NULL ,
house_color VARCHAR (40) NOT NULL ,
created_at DATETIME NOT NULL,
modified_at DATETIME NOT NULL ,
active_status BOOLEAN DEFAULT 1,
PRIMARY KEY (house_id),
CONSTRAINT fk_houses_user_id FOREIGN KEY (user_id) REFERENCES users (user_id)
);

apitest Entity Relation Diagram generated by DbSchema

This can be found in the project..

You can check the project and go through the code ... It is explanatory. At this stage, the slim framework has been downloaded. (vendor directory contains slim stuff).

On the index.php file.. It is where we would be routing our functions with required url.

E.g for the test function .. \$app ->get("/name/:name", "test"); is made.

get is a method, /name/:name is the api call and the test is the function. :name means a name should be included while testing. e.g <http://localhost/apitest/name/mesh>

Methods:

- GET: to request details on a particular resource/set of resources.
- POST: to supply details on a particular resource/set of resources.
- PUT: to update a particular attribute of a specific resource.
- DELETE: to permanently remove a specific resource.

the index page is the real deal.

Work through the code, play with it and feel free to ask questions....

You can manipulate the project to your own taste i.e. use it as your own work ...

Note:

To test your functions using POST