NOTE: Our database might be down because Amazon said we have somehow nearly reached the cap on free tier hourly usage. Make sure our database is actually up before trying to test the API.

This is a quick guide to locally testing the API using postman with Bitnami's WAMP stack.

You can use a different WAMP or LAMP stack, but I don't know if there are differences.

Step 1:

A: Download the files in the "api calls" folder in the "temp" branch on our github repository.

B: Download and install postman from:

https://www.getpostman.com/downloads/

C: Download and install Bitnami's WAMP stack from:

https://bitnami.com/stack/wamp

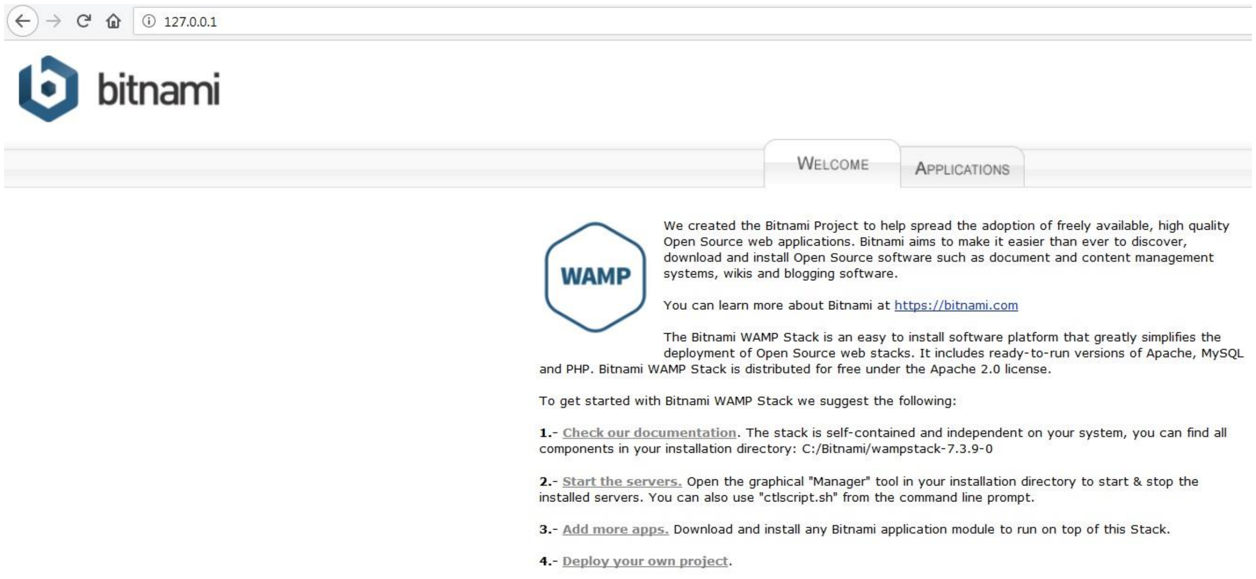
NOTE: Pay attention to file path where you install Bitnami WAMP stack.

D: Download and install MySQL workbench from:

https://dev.mysql.com/downloads/workbench/

Step 2:

A: After installing the WAMP stack, this webpage should open in your browser:



We created the Bitnami Project to help spread the adoption of freely available, high quality Open Source web applications. Bitnami aims to make it easier than ever to discover, download and install Open Source software such as document and content management systems, wikis and blogging software.

You can learn more about Bitnami at https://bitnami.com

The Bitnami WAMP Stack is an easy to install software platform that greatly simplifies the deployment of Open Source web stacks. It includes ready-to-run versions of Apache, MySQL and PHP. Bitnami WAMP Stack is distributed for free under the Apache 2.0 license.

To get started with Bitnami WAMP Stack we suggest the following:

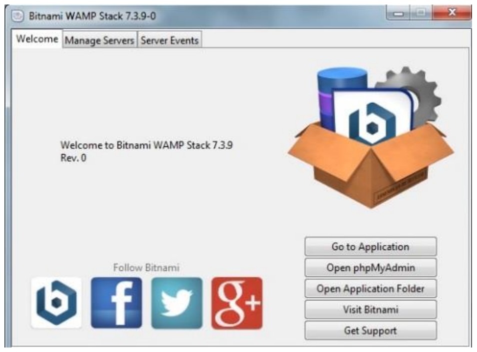
1.- Check our documentation. The stack is self-contained and independent on your system, you can find all components in your installation directory: C:/Bitnami/wampstack-7.3.9-0

2.- Start the servers. Open the graphical "Manager" tool in your installation directory to start & stop the installed servers. You can also use "ctlscript.sh" from the command line prompt.

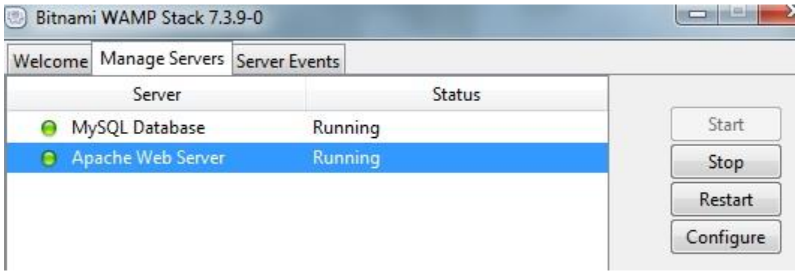
3.- Add more apps. Download and install any Bitnami application module to run on top of this Stack.

4.- Deploy your own project.

This program should open:



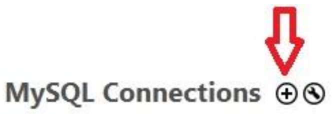
By clicking on the "Manage Servers" tab, you should see this:



Server	Status
MySQL Database	Running
Apache Web Server	Running

Make sure the Apache Web Server is running.

B: In MySQL workbench, create a new connection using this button:

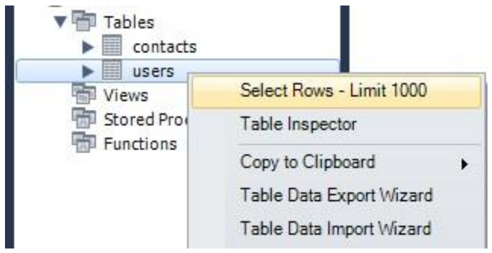


C: In MySQL workbench, enter this information for the database:

- hostname: whodoyaknow.cgp1kdh6zde2.us-east-1.rds.amazonaws.com
- port: 3306
- name: ContactManager
- username: AdminRCL
- password: PRTZ100!

Once entered, make sure you can connect to the database.

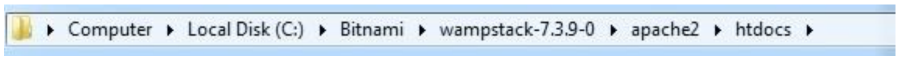
Open the database, then tables, and right click on one of the tables, then use select rows, the open show in the picture to view table data:



There should already be some dummy data in both the users and contacts tables.

Step 3:

A: Follow this file path if you installed WAMP in default file path:



Computer ▸ Local Disk (C:) ▸ Bitnami ▸ wampstack-7.3.9-0 ▸ apache2 ▸ htdocs ▸

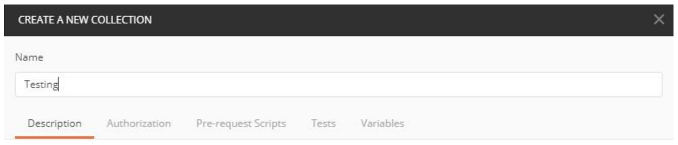
Otherwise, follow the file path you specified when installing WAMP.

B: Create a folder in the htdocs folder, name the folder how you want. For our demonstration, the folder will be named: project

C: Put the API calls you downloaded from github into the project folder (the folder you just created in Step 3: B)

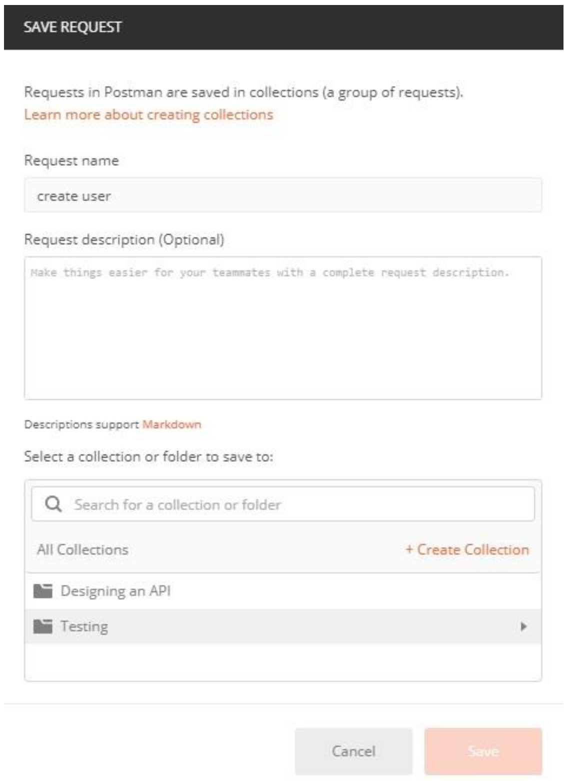
Step 4:

A: Open postman, click new button at top left, click collection button, enter the name you want to use, for our demonstration, we will name it: Testing



Name: Testing

B: After creating the collection, now click the new button at top left again, click the request button, enter the name of the API call you want to use in the request name, as shown, and click on Testing (the collection you just created in Step 4: A), then save:


SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).
Learn more about creating collections

Request name: create user

Request description (Optional): Make things easier for your teammates with a complete request description.

Descriptions support Markdown

Select a collection or folder to save to:

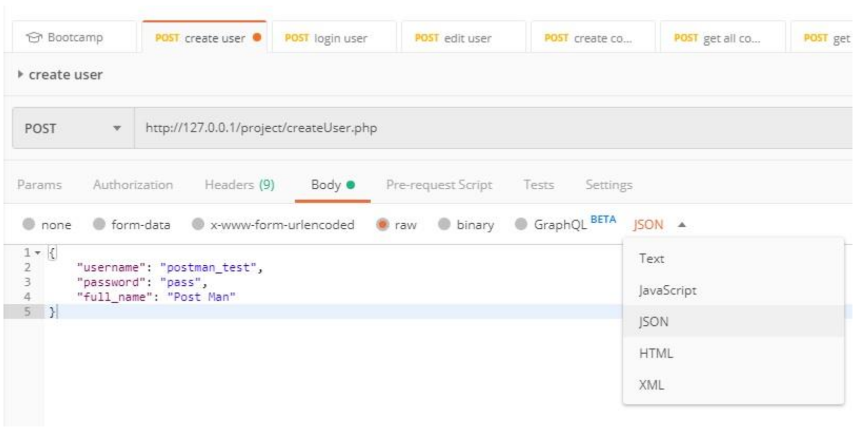
Step 5:

A: Click on "GET" and change to "POST".

B: Enter this in the bar next to POST: http://127.0.0.1/project/createUser.php

NOTE: if you named the folder something other than project, replace project with your folder name.

C: Click on "body", then click on "raw", then click on "text" and select "JSON", as shown:



POST http://127.0.0.1/project/createUser.php

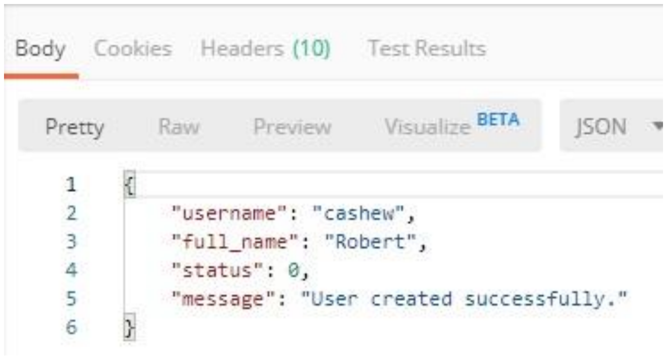
```
{
    "username": "postman_test",
    "password": "pass",
    "full_name": "Post Man"
}
```

D: Open “API documentation” document that you got from “api calls” folder on github.

E: Enter required input for the API call you want to test in JSON format.

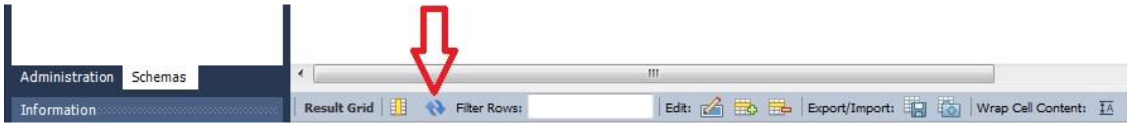
NOTE: An example of what to enter for “createUser.php” is shown in Step 5: C.

F: Click “Send”, you should get some text in the second text box. You can format it if you want by clicking on “HTML” and selecting to “JSON”. The message should show that you succeeded. If status > 0, that means there was an error and the message should display what was wrong. The JSON text may have been entered incorrectly, maybe a required input is missing, maybe there was a problem connecting to the database, maybe the username already exists within the table. If the API call succeeds, you should see something similar to:



```
{
    "username": "cashew",
    "full_name": "Robert",
    "status": 0,
    "message": "User created successfully."
}
```

G: Go to MySQL workbench, click on refresh, as shown to view changes you made:


Administration
Schemas
Information
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content: