**Introduction to API**

The purpose of routes is to capture particular action to do. Example, retrieving student records, classes for the endpoint. Plural and Noun description for endpoint/route.

Get, put, post, delete are known as request methods. Get means retrieving records, post means adding records, put/patch means updating records.

Status Code Meanings:

* 200-299 – success
* 300-399 – redirect
* 400-499 – request error
* 500-onwards – server errors

Next is to make a way to retrieve records from the database. Create a folder named modules and then file named get.php and post.php. Inside get.php you will create a class named get and class named post inside post.php. The code inside the get and post file are methods and functions that will retrieve records from the database.

**1. Database Connection and Retrieving Records**

First process is to create a database in php localhost. Using xampp control panel to activate both Apache and MySQL. After that, go to php admin to finally create your desired database. Create a table inside the database with your own desired categories and values.

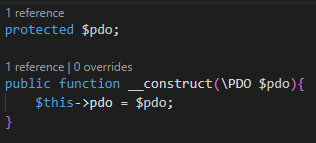
Create Folder named Config. Create 2 files inside it named Index.php and Database.php. Index.php allows us to protect the files inside the config folder so that not anyone can access your own password.

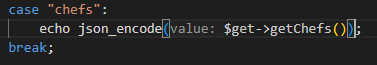
Write down configurations for database in database.php . Add headers, headers are meta data or additional configuration setting for your request and response from your database. Required in order for us to restrict or control the data that will be transferring from the client towards the back-end and vice versa.

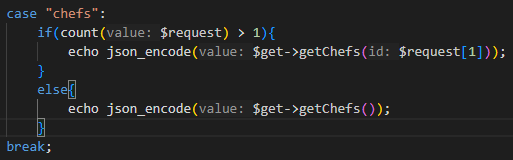
Defining constant variables. Variables that are available at the entirety of the project. Composed of database names and so on that will be used later.

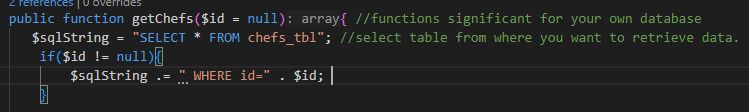
Create a class connection. This will serve as the class that is responsible for connecting to our database. Create first the connectionString. connectionString is what we used to connect into database, where driver, database name. Same with process of java when connecting to database.

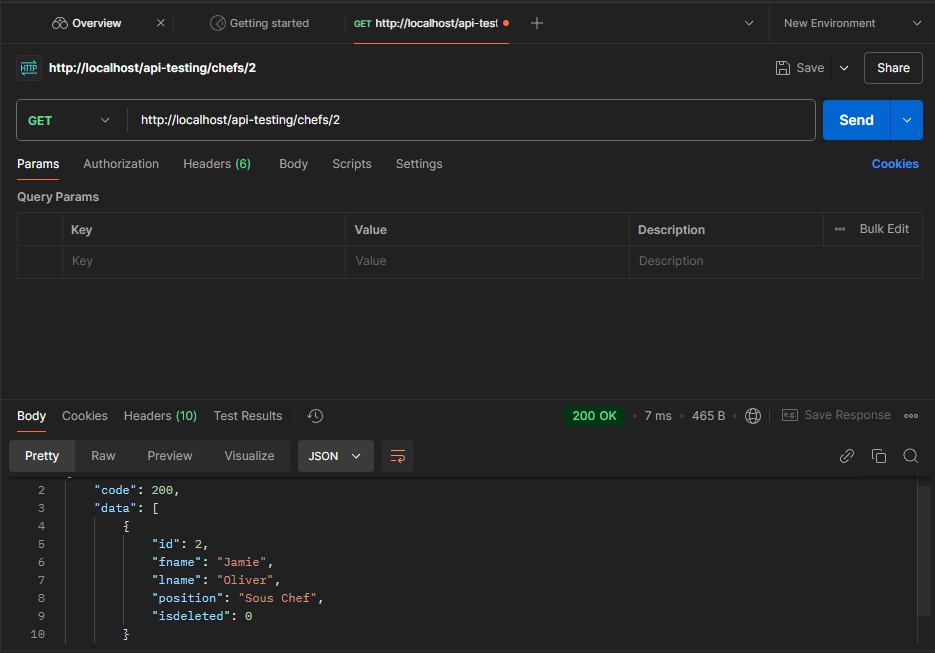
Fetch mode is how database returns data. By default, it returns values in an array index format but when we use fetch\_assoc it will return the data in a column value pair format. Example, in default [loudel, manaloto, middle, extension]. In fetch\_assoc, [“firstname”=>”loudel”,”lastname”=>”manaloto”].

In routes, import the database.php and should be put at top of other imported files. Check if there’s pdo object if the code is right or there’s an error. Create $db = new connection (comes from database.php class). Then, create a $pdo = $db->connect() (to call in database.php). Pass the $pdo to post and get by putting $pdo inside the parenthesis $post = new Post($pdo) and $get = new Get($pdo). It should have a constructor to pass the values as arguments.

Add this code to both Get.php and Post.php to pass the $pdo to post and get in routes.php. Add necessary functions for your desired API. Next is to retrieve records. Try and Catch is used to handle exceptions. $errmsg = $e->getMessage() (To capture error message from retrieval process). Try and catch is a way to retrieve error in the database. To check if it’s successful in retrieving records we need to run it in postman. Since it can’t print array values since the format of the records is on string, we need to convert the routes of the case chefs in order to print it in string values. Using json\_encode to convert it to json string.

What if you only want a specific record to be retrieved? We will be using a condition where the request which is an array will be specified with an index that will be put in the if statement. If the request input is 1 it will display the corresponding value 1 from the records (can be id). Else block will execute all records if there’s no input request value/default.

We also modify in get where we assign an id with a value of null in the getChefs function. This id will then put inside the if condition where once a value exist and match the id, it will retrieve that particular record.

Result

Summary of 1st Video

* Create folder named config. Created 2 files inside the folder named index.php and database.php. Inside the database.php we added headers and configuration to handle requests and also defined constants like the server, database name, username and password. Last part is created a class connection and defined the connection string the options of pdo and created a method connect() for returning a new pdo object. Which is responsible for connecting into database.
* Imported into routes.php (database.php). Created instantiation of class and pdo object and passed on post and get as argument. Since we defined an argument, we created a constructor inside get class and post class.
* Created endpoint chefs where it will echo the getChefs function from get.php. Then, in get.php function will be defined where we will retrieve record. Used try and catch to create a condition once a record was found or not. Inside try there’s an if-else condition where once a record was found it will be stored inside $result and will be pushed using array\_push command. It will also return code and data once condition is true. Once else it will display error message.