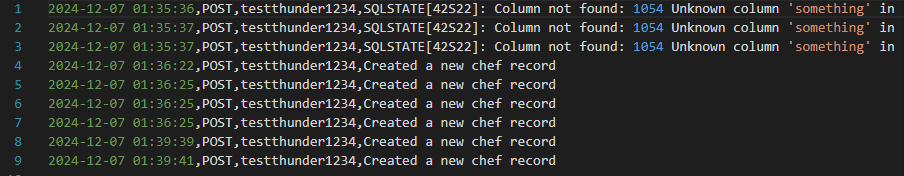
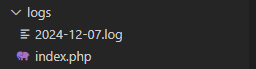
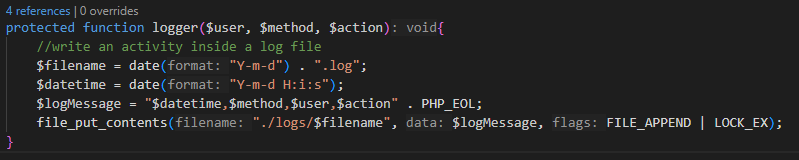
**6. Logger**

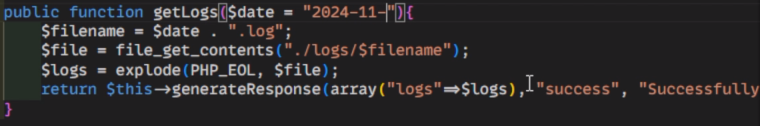
Create a protected function named logger and pass $user, $method and $message. We will save these parameters in a .log file. We also need to save the date and time but we can’t use the command date\_create because it will output a lot of information that’s not necessary (timezone and timezone type). We will make it simple just by date function and put format Year-Month-Day Hour-Minute-Second. You can customize it based on what you prefer (saved on $datetime). We’re going to create $logMessage where inside it we will concatenate $datetime, $method, $user, $message/action. Also we will concatenate $logMessage with PHP\_EOL. The purpose of PHP\_EOL is to add newline to our log so that it won’t be confusing to see the log history.We’re just going to save it in text file, we will use the command file\_put\_contents and put the file name “example.log”. $logMessage, FILE\_APPEND which if it finds a filename that exists, it will append the data to the file instead of overwriting it. We will also add LOCK\_EX which is an exclusive lock on the file while proceeding to the writing. We will uncomment the $filename, the purpose of the variable is to name the log based on the current date and then will be concatenated with “.log”. To check, you change the value in file\_put\_contents from “example.log” to $filename. It will be like this:

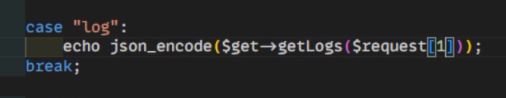


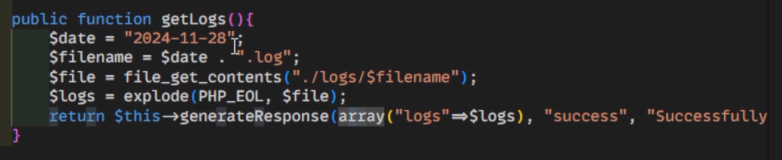


Since the log folder is in root folder, we will make a new folder named logs that will store all the logs to prevent too much file inside the root folder. We will make modifications in file\_put\_content where we will add “./logs/$filename” so that it will be saved under logs folder.

Final code for function logger.

We will also create a log for retrieving records so we will do that in get.php. We will create a public function named getLogs(). We will just copy the $filename command from logger function. Then, we will create $file that will store file\_get\_contents and will be passing the directory/folder location to retrieve. We will create $logs that contains explode with parameters command PHP\_EOL to separate common symbols in the record and $file to get the contents. We will create an instance generateResponse to make it more detailed. The only difference from what we did from get and post is that the data to be passed is in array(“logs”=>$logs), remarks and message will be the same and the status code will be set in constant 200.  
(OPTIONAL) Since there are different dates from logs, we will need to input specific date to retrieve record. There are 2 options, first is to store it in variable named $date then put the date in string format (called dynamic). Then, change the date to $date in filename. Second option is to pass $date inside the method getLogs(). Then we will assign a default date in string format then were going to add $request[1] in case “log” in routes.php. If incase you want to retrieve all log file, we need to display all directories/files then we will loop through to pass it.





If you want to have a separate log for errors you can do it so that you can easily retrieve error records. You can add parameter for logger status code for example from 200-299 which is success, you can save it on success action log and when its between 400-499 you can save it on failed action log. (OPTIONAL)

**Another Version for Retrieval (more object-oriented version)**

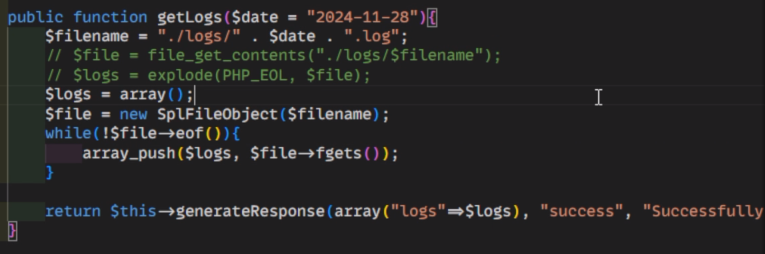
$filename = “./logs/” . $date . “.log”;

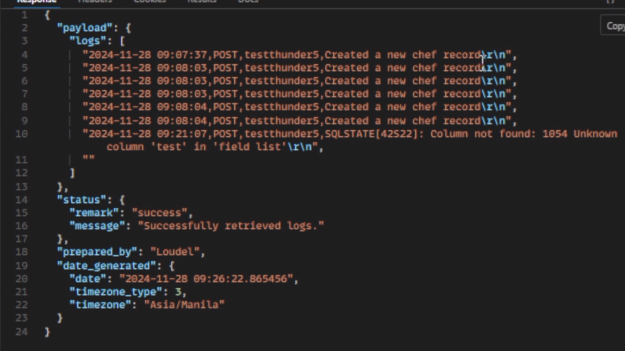
$logs = array();

$file = new SplFileObject($filename);

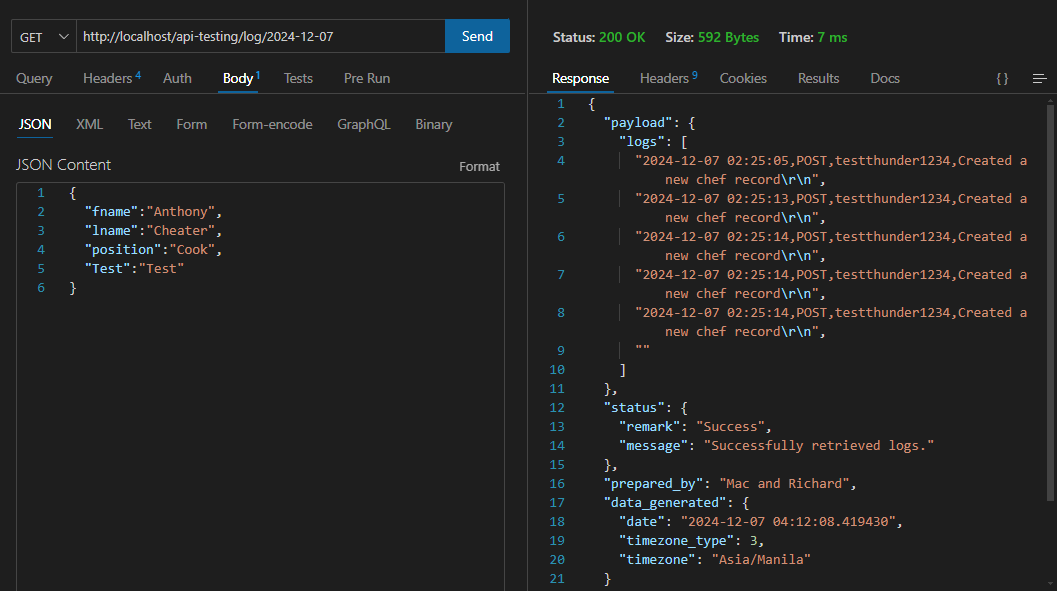
while(!$file->eof()){

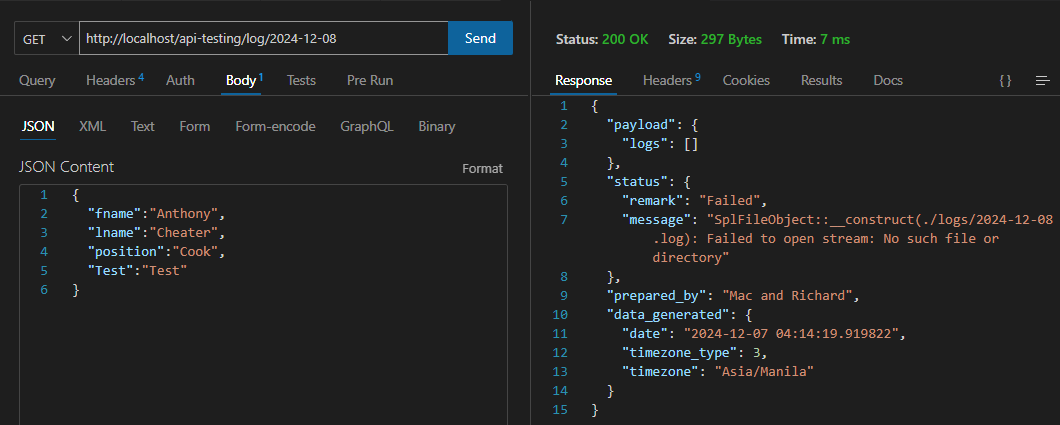
array\_push($logs, $file->fgets());

}

This can be used if the text file is so massive like one hundred thousand lines of code. If it’s like 1000 lines of code, file\_get\_contents and explode is enough. It will also include \r\n:

Apply file-handling error like try catch incase log file doesn’t exist. (code idea only)



If There’s no log in specific date: