

Bus schedule Project APIs using PHP and Mysql .

In short we can come on a common stage of magic that API is used to communicate with server having a standard format (JSON/XML) of data to be transfer from server to client and vice versa . In this article we will be using standard format of data as JSON, PHP as server language and MySql for storing data.

Overview :-

We have a four different table in the database to communicate with server and get data from database using AJAX call. Table for bus details, driver details, schedule for buses with drivers. So for sure we will be creating routes for listing admin access and insert new user record. There is a folder server which will hold all codes, files & directory structure of API server using PHP.

```
|— server/
|—   common/
|—       config.php — file used for configuration of base url and web page title
|—       controller.php — file used for connecting to the database. And do all server side CRUD
function from mysql database. It has the database details also.
|—       pageheader.php — file used for configuring page header for common section
|—       pagefooter.php — file used for configuring page footer for common section
|—       logout.php — destroy the session details from server
|—       encipher.php — contains function for encrypting password.

|— modal/
|—     buses.php — file that will list user in output as JSON data.
|—     drivers.php — this file will take drivers data and store to database.
|—     schedule.php — file that will assign the driver to bus data from database.
|—     buses.php — file that will create, edit, read, delete all bus data from database.
```

Common Folder:-

1. config.php :- web application configuration
2. controller.php :- controller class communicate with data and do CRUD operations

API Folder:-

1. buses.php :- To create new bus POST method with bus number, bus capacity, bus make, bus model. Same for edit with bus id from database.
2. drivers.php :- By sending POST data with driver first name, last name, username, password, social security number, email, same for edit with driver id
3. schedule.php :- To schedule we need POST data with bus id, driver from data base, also we need start time and data, end time and date of the journey.

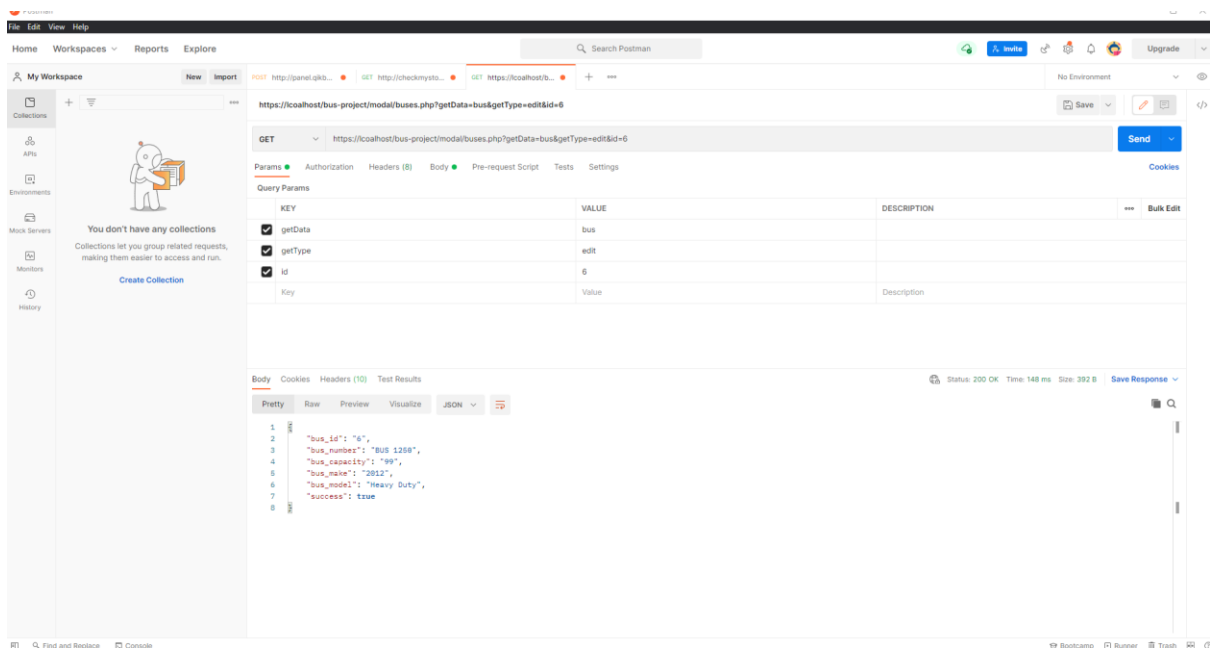
1. Bus list API :- modal/buses.php

1.1. Source code :—

```
// All get method view data
if (($_SERVER['REQUEST_METHOD'] === "GET") && isset($_GET['getData']) && $_GET['getData'] == 'bus' && $_GET['getType'] == 'edit') {
    $id = $_GET['id'];
    $rows = $db_handle->runOneQuery("SELECT * FROM `bus` WHERE bus_id = '$id' LIMIT 0,1");
    if ($rows) {
        $rows['success'] = true;
    }
}

echo json_encode($rows);
}
```

1.2. Response from server : —



Now if all set open browser and type url: <https://localhost/bus-project/modal/buses.php?getData=bus&getType=edit&id=1> (for me host is localhost) and then response will as follows.

JSON response of bus details.

```
{
  "bus_id": "1",
  "bus_number": "BUS 91",
  "bus_capacity": "60",
  "bus_make": "2000",
  "bus_model": null,
  "success": true
}
```

2. Driver API :- modal/drivers.php

1.1. Source code : —

```
// All get method view data
if (($SERVER['REQUEST_METHOD'] === "GET") && isset($_GET['getData']) && $_GET['getData'] == 'drivers' && $_GET['getType'] == 'edit') {
    $id = $_GET['id'];
    $rows = $db_handle->runOneQuery("SELECT * FROM `users` WHERE id = '$id' LIMIT 0,1");
    if ($rows) {
        $rows['success'] = true;
    }
}

echo json_encode($rows);
}
```

1.2 Request and Response

The screenshot shows the Postman interface with a GET request to `https://localhost/bus-project/modal/drivers.php?getData=drivers&getType=edit&id=9`. The query parameters are `getData=drivers`, `getType=edit`, and `id=9`. The response is a JSON object with the following details:

```
{
  "id": "9",
  "name": "Sam",
  "username": "Sam",
  "password": "282cb962ac99675b964087152d234d70",
  "sa_number": "161215",
  "email": "sams@gmail.com",
  "driver_id": null,
  "status": "0",
  "success": true
}
```

Now if all set open browser and type url <https://localhost/bus-project/modal/drivers.php?getData=drivers&getType=edit&id=9> (for me host is localhost) and then response will as follows.

```
{
  "id": "9",
  "fname": "Sam",
  "lname": "Ral",
  "username": "sam@123",
  "password": "202cb962ac59075b964b07152d234b70",
  "ss_number": "161615",
  "email": "sams@gmail.com",
  "driver_id": null,
  "status": "0",
  "success": true
}
```

3. Schedule API :- modal/schedule.php

1.1. Source code : —

```
// All get method view data
if (($_SERVER['REQUEST_METHOD'] === "GET") && isset($_GET['getData']) && $_GET['getData'] == 'schedule' && $_GET['getType'] == 'edit') {
    $id = $_GET['id'];
    $rows = $db_handle->runOneQuery("SELECT * FROM `schedule` WHERE schedule_id = '$id' LIMIT 0,1");
    if ($rows) {
        $rows['success'] = true;
    }

    echo json_encode($rows);
}
```

1.2 Request and Response

The screenshot shows a Postman interface with a GET request to the endpoint `https://localhost/bus-project/modal/schedule.php?getData=schedule&getType=edit&id=4451`. The query parameters are defined as follows:

KEY	VALUE	DESCRIPTION
getData	schedule	
getType	edit	
id	4451	

The response is a JSON object with the following structure:

```
{
  "schedule_id": "4451",
  "start_date": "2022-07-31",
  "end_date": "2022-08-06",
  "start_time": "16:15",
  "end_time": "16:14",
  "driver_id": "7",
  "bus_id": "4",
  "status": "0",
  "success": true
}
```

Now if all set open browser and type url <https://localhost/bus-project/modal/schedule.php?getData=schedule&getType=edit&id=4451> (for me host is localhost) and then response will as follows.

```
{
    "schedule_id": "4451",
    "start_date": "2022-07-31",
    "end_date": "2022-08-06",
    "start_time": "16:13",
    "end_time": "16:14",
    "driver_id": "7",
    "bus_id": "4",
    "status": "0",
    "success": true
}
```

4. Search API :- modal/schedule.php (Using driver name or bus number)

1.1. Source code : —

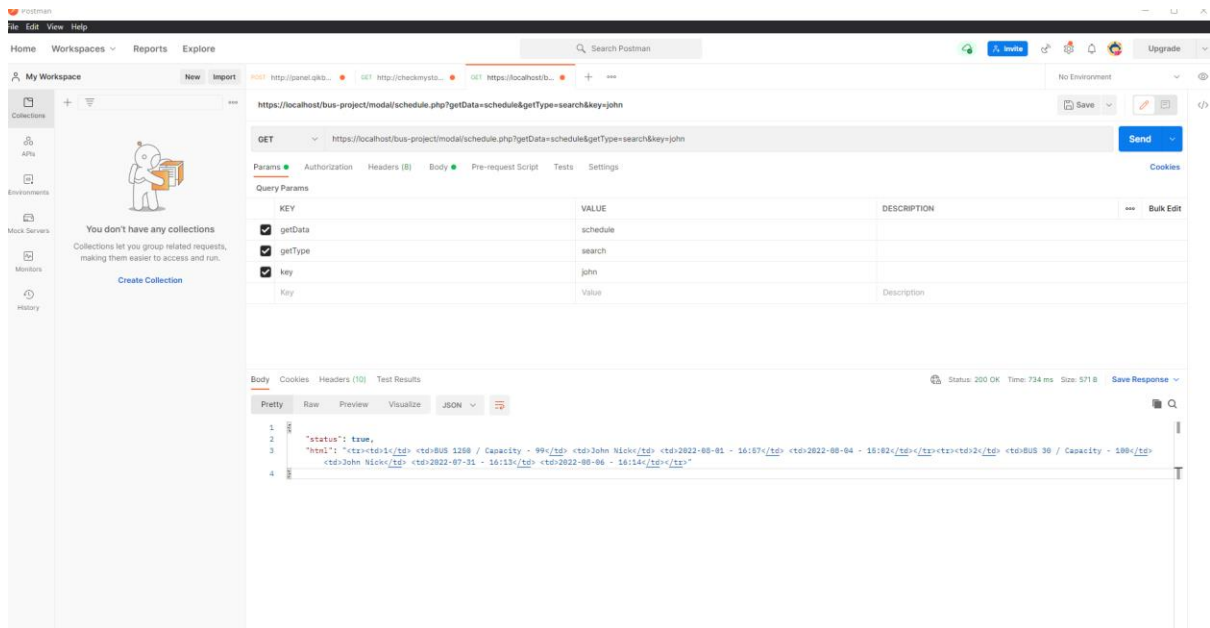
```
// All get method view data
if (($SERVER['REQUEST_METHOD'] === "GET") && isset($_GET['getData']) && $_GET['getData'] == 'schedule' && $_GET['getType'] == 'search') {
    $key = $_GET['key'];
    $sql_driver = "SELECT bus.bus_number,bus.bus_capacity,users.fname,users.lname,schedule.* FROM `schedule` JOIN bus ON bus.bus_id = schedule.bus_id JOIN users ON users.id = schedule.driver_id WHERE users.fname LIKE '$key%' OR users.lname LIKE '$key%' ";
    $sql_bus = "SELECT bus.bus_number,bus.bus_capacity,users.fname,users.lname,schedule.* FROM `schedule` JOIN bus ON bus.bus_id = schedule.bus_id JOIN users ON users.id = schedule.driver_id WHERE bus.bus_number LIKE '$key%'";
    // echo $sql_bus;
    if ($db_handle->numRows($sql_driver) > 0) {
        $rows = $db_handle->runQuery($sql_driver);
    } else if ($db_handle->numRows($sql_bus) > 0) {
        $rows = $db_handle->runQuery($sql_bus);
    } else {
        $rows = [];
    }

    $i = 1;
    $html = '';
    foreach ($rows as $row) {
        $html .= "<tr>";
        $html .= "<td>". $i . "</td>";
        $html .= "<td>". $row['bus_number'] . " / Capacity - " . $row['bus_capacity'] . "</td>";
        $html .= "<td>". $row['fname'] . " - " . $row['lname'] . "</td>";
        $html .= "<td>". $row['start_date'] . " - " . $row['start_time'] . "</td>";
        $html .= "<td>". $row['end_date'] . " - " . $row['end_time'] . "</td>";
        $html .= "</tr>";
        $i++;
    }

    if ($rows) {
        $response['status'] = true;
        $response['html'] = $html;
    } else {
        $response['status'] = false;
    }

    echo json_encode($response);
}
```

1.2 Request and Response



Now if all set open browser and type url <https://localhost/bus-project/modal/schedule.php?getData=schedule&getType=search&key=john> (for me host is localhost) and then response will as follows.

```
{
  "status": true,
  "html": "<tr><td>1</td> <td>BUS 1250 / Capacity - 99</td> <td>John Nick</td> <td>2022-08-01 - 16:57</td> <td>2022-08-04 - 15:02</td></tr><tr><td>2</td> <td>BUS 30 / Capacity - 100</td> <td>John Nick</td> <td>2022-07-31 - 16:13</td> <td>2022-08-06 - 16:14</td></tr>"
}
```

4. Search API :- modal/schedule.php (Using schedule departure and arrival date)

1.1. Source code : —

```

// All get method view data
if (($SERVER['REQUEST_METHOD'] === "GET") && isset($_GET['getData']) && $_GET['getData'] == 'schedule' && $_GET['getType'] == 'search_date') {
    $from_date = $_GET['from_date'];
    $to_date = $_GET['to_date'];
    $sql_date = "SELECT bus.bus_number,bus.bus_capacity,users.fname,users.lname,schedule.* FROM `schedule` JOIN bus ON bus.bus_id = schedule.bus_id JOIN users ON
    users.id = schedule.driver_id WHERE schedule.start_date >= '$from_date' OR schedule.end_date <= '$to_date'";
    if ($db_handle->numRows($sql_date) > 0) {
        $rows = $db_handle->runQuery($sql_date);
    } else {
        $rows = [];
    }

    $i = 1;
    $html = '';
    foreach ($rows as $row) {
        $html .= "<tr>";
        $html .= "<td>" . $i . "</td>";
        $html .= "<td>" . $row['bus_number'] . " / Capacity - " . $row['bus_capacity'] . "</td>";
        $html .= "<td>" . $row['fname'] . " " . $row['lname'] . "</td>";
        $html .= "<td>" . $row['start_date'] . " - " . $row['start_time'] . "</td>";
        $html .= "<td>" . $row['end_date'] . " - " . $row['end_time'] . "</td>";
        $html .= "</tr>";
        $i++;
    }

    if ($rows) {
        $response['status'] = true;
        $response['html'] = $html;
    } else {
        $response['status'] = false;
    }

    echo json_encode($response);
}

```

1.2 Request and Response

The screenshot shows a Postman interface with a GET request to `https://localhost:bus-project/model/schedule.php?getData=schedule&getType=search_date&from_date=2022-08-01&to_date=2022-08-03`. The request parameters are:

KEY	VALUE	DESCRIPTION
getData	schedule	
getType	search_date	
from_date	2022-08-01	
to_date	2022-08-03	
Key	Value	Description

The response is a JSON object with the following structure:

```

{
  "status": true,
  "html": "<tr><td>1</td><td>BUS 1258 / Capacity - 99</td><td>John Nick</td><td>2022-08-01 - 16:57</td><td>2022-08-04 - 15:52</td></tr><tr><td>2</td><td>BUS 38 / Capacity - 100</td><td>San Raj</td><td>2022-07-31 - 22:26</td><td>2022-08-02 - 23:26</td></tr>"
}

```

Now if all set open browser and type url https://localhost/bus-project/modal/schedule.php?getData=schedule&getType=search_date&from_date=2022-08-01&to_date=2022-08-03 (for me host is localhost) and then response will as follows.

```
{
  "status": true,
  "html": "<tr><td>1</td> <td>BUS 1250 / Capacity - 99</td> <td>John Nick</td> <td>2022-08-01 - 16:57</td> <td>2022-08-04 - 15:02</td></tr><tr><td>2</td> <td>BUS 30 / Capacity - 100</td> <td>Sam Ral</td> <td>2022-07-31 - 22:26</td> <td>2022-08-02 - 23:25</td></tr>"
}
```