

# e-API Dashboard Tutorials

---

## Introduce

This tutorial will show you how use the Postman to send the request and get employees response.

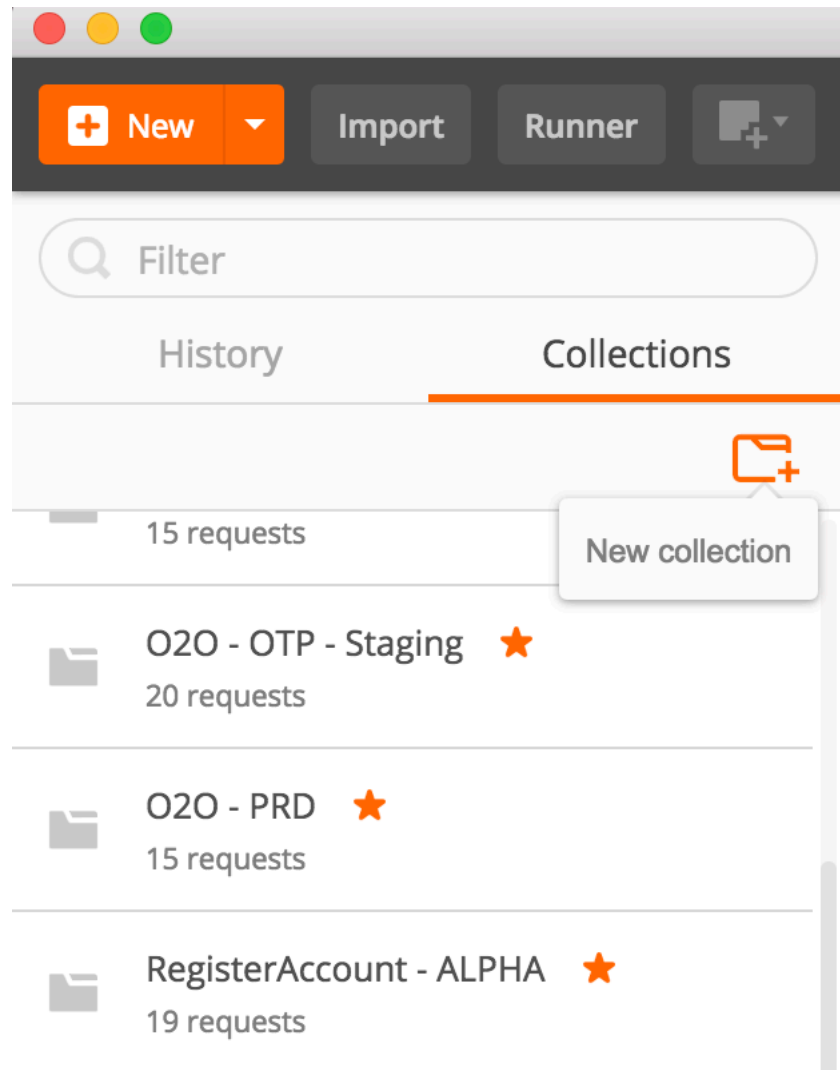
### 1. Get all employees data

This API will retrieving all employees data from the table and return information that included name, salary and age.

|                  |  |
|------------------|--|
| URL              | http://sangbui.com/api/v1/employees  |
| Method           | GET  |
| URL Params       | <none>   |
| Header           | Content-Type: application/json<br>Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==  |
| Data Params      | <none>   |
| Success Response | Code: <b>200 OK</b><br>Content:<br><pre>[   {     "id": "&lt;id&gt;",     "employee_name": "&lt;name&gt;",     "employee_salary": "&lt;salary&gt;",     "employee_age": "&lt;age&gt;"   },   {     "id": "&lt;id&gt;",     "employee_name": "&lt;name&gt;",     "employee_salary": "&lt;salary&gt;",     "employee_age": "&lt;age&gt;"   } ]</pre> |
| Error Response   | Code: <b>404 Not Found</b>   |
| Sample Call      | <none>   |
| Notes            | <none>   |

The above table is the API requirements, it will include all information that needed for the Postman tool.

1. Open the Postman, create new Collection by click on **New Collection** icon. Enter the collection name.



2. Enter the URL at “**Enter request URL**” - as the above requirement, the URL should be [“http://sangbui.com/api/v1/employees”](http://sangbui.com/api/v1/employees).

If you click on **Send** button, it will show the error message that required you to have an authorization. We move to step 3 to set the header and authorization information.

3. Click on the tab **Header**. Enter 2 keys and values as the above table.

| Key   | Value                          |
|---|--------------------------------|
| <input checked="" type="checkbox"/> Content-Type  | application/json               |
| <input checked="" type="checkbox"/> Authorization | Basic d2VlX2FwcDpjaGFuZ2VpdA== |
| New key   | Value                          |

Now you can try to click on **Send** button again, it will show the successful code **200 OK** and list all employees data.

```
Body Cookies (1) Headers (8) Test Results Status: 200 OK Time: 179 ms Size: 1.69 KB
Pretty Raw Preview JSON
1- [
2- {
3-   "id": "1",
4-   "employee_name": "Nguyen Trung Nam",
5-   "employee_salary": "800",
6-   "employee_age": "25"
7- },
8- {
9-   "id": "2",
10-  "employee_name": "Phan Nhan Tai",
11-  "employee_salary": "450",
12-  "employee_age": "23"
13- },
14- {
15-  "id": "3",
16-  "employee_name": "Bui Tuan Minh",
17-  "employee_salary": "550",
18-  "employee_age": "30"
19- },
20- {
21-  "id": "4",
22-  "employee_name": "Nguyen Van Anh",
23-  "employee_salary": "950",
24-  "employee_age": "29"
25- },
26- {
27-  "id": "5",
28-  "employee_name": "Nguyen Thi Dieu",
29-  "employee_salary": "1200",
30-  "employee_age": "27"
31- },
32- {
```

4. Click on [X] button, to close the tab, it will show the popup and asks you to save the request, click on **Save** button and enter the Request name.

Select the Collection name that we have created at the step 1. That's done for the first tutorial!

## SAVE REQUEST



Requests in Postman are saved in collections (a group of requests).

[Learn more about creating collections](#)

Request name

Get all employees data

Request description (Optional)

Adding a description makes your docs better

Descriptions support Markdown

Select a collection or folder to save to:

- All Collections + Create Collection
- API Localhost
- Employee - STAGING >
- ForgetPassword API

Cancel

Save

With the second API, we will not get the list of employees but just one record by the id. At the URL we will input the employee\_id, so it will show the data of that id.

## 2. Get an employee by ID

|                  |   |
|------------------|---|
| URL              | http://sangbui.com/api/v1/employees/<employee_id>   |
| Method           | GET   |
| URL Params       | <employee_id>   |
| Header           | Content-Type: application/json<br>Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==   |
| Data Params      | <none>  |
| Success Response | Code: <b>200 OK</b><br>Content:<br>[<br>{<br>"id": "<id>",<br>"employee_name": "<name>",<br>"employee_salary": "<salary>",<br>"employee_age": "<age>"<br>}<br>] |
| Error Response   | Code: <b>404 Not Found</b>  |
| Sample Call      | http://sangbui.com/api/v1/employees/1   |
| Notes            | In our database, the employee_id from <b>1 to 15</b> will be stable and will not removed, so you can use these IDs for testing and get the sample data.         |

Click on the [+] icon to create the new tab (Request).

1. Set the request **Header** as the first API
2. At the URL, **adding the employ\_id** that you want to see the details:  
`http://sangbui.com/api/v1/employees/<employee_id>`

The screenshot shows a REST client interface for a request titled "Get an employee by id". The method is GET and the URL is `http://sangbui.com/api/v1/employees/2`. The response status is 200 OK, with a time of 226 ms and a size of 399 B. The response body is displayed in JSON format:

```
1 {
2   "id": "2",
3   "employee_name": "Phan Nhan Tai",
4   "employee_salary": "450",
5   "employee_age": "23"
6 }
```

### 3. Add new employee

This API will support for adding the new employee to the database. The `employee_id` will be auto increment. Check the new submitted records at the home page. It will show 10 latest records and all value from the employee table.

|             |  |
|-------------|--|
| URL         | <code>http://sangbui.com/api/v1/employees</code>   |
| Method      | POST   |
| URL Params  | <none>   |
| Header      | Content-Type: application/json<br>Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==  |
| Data Params | <pre>{   "employee_name": "&lt;name&gt;",   "employee_salary": "&lt;salary&gt;",   "employee_age": "&lt;age&gt;" }</pre> |

|                  |   |
|------------------|---|
| Success Response | Code: <b>200 OK</b><br>Content:<br>{<br>"status": 1,<br>"status_message": "Employee added successfully!"<br>} |
| Error Response   | Code: <b>404 Not Found</b>  |
| Sample Call      | {<br>"employee_name" : "An",<br>"employee_salary": "1000",<br>"employee_age": "29"<br>}                       |
| Notes            | Check the new submitted records ID and details at the home page.  |

The big difference on this API is the **Method** name, it is the POST method instead of the GET as before. So with the POST method we will need to add request Body

1. Add the **Header** as previous API
2. Select the method: **POST**
3. Add the Data Params

```
{
  "employee_name" : "<name>",
  "employee_salary": "<salary>",
  "employee_age": "<age>"
}
```

The screenshot displays a REST client interface for a POST request to the endpoint `http://sangbui.com/api/v1/employees`. The request body is a JSON object with the following structure:

```
1 {
2   "employee_name": "An",
3   "employee_salary": "1000",
4   "employee_age": "29"
5 }
```

The response status is `200 OK`, with a response time of `170 ms` and a size of `457 B`. The response body is a JSON object:

```
1 {
2   "status": 1,
3   "status_message": "Employee added successfully!"
4 }
```

With the method **PUT** and **DELETE**, it will be the same as POST method, so now you can make a try!!



## 4. Update an employee information

This API will help to update an employee information by the employee\_id.


|                  |  |
|------------------|--|
| URL              | http://sangbui.com/api/v1/employees  |
| Method           | PUT  |
| URL Params       | <none>   |
| Header           | Content-Type: application/json<br>Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==  |
| Data Params      | <pre>{   "id": "&lt;id&gt;",   "employee_name": "&lt;new_name&gt;",   "employee_salary": "&lt;new_salary&gt;",   "employee_age": "&lt;new_age&gt;" }</pre> |
| Success Response | <b>Code: 200 OK</b><br>Content:<br><pre>{   "status": 1,   "status_message": "Employee updated successfully." }</pre>                                      |
| Error Response   | <b>Code: 404 Not Found</b>   |
| Sample Call      | <pre>{   "id" : 7,   "employee_name" : "Nam",   "employee_salary": "800",   "employee_age": "25" }</pre>   |
| Notes            | <none>   |

## 5. Delete an employee

This API will support to delete an employee by the employee\_id.

|                  |   |
|------------------|---|
| URL              | http://sangbui.com/api/v1/employees   |
| Method           | DELETE  |
| URL Params       | <none>  |
| Header           | Content-Type: application/json<br>Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==                                 |
| Data Params      | {<br>"id" : <id><br>}   |
| Success Response | <b>Code: 200 OK</b><br>Content:<br>{<br>"status": 1,<br>"status_message": "Employee deleted successfully."<br>} |
| Error Response   | <b>Code: 404 Not Found</b>  |
| Sample Call      | {<br>"id" : 5<br>}  |
| Notes            | <none>  |

The complete test Collection should be as the below structure, and all request will get the valid response as **200 OK**.

|   |                             |
|---|-----------------------------|
|  | API Localhost<br>5 requests |
| <b>GET</b>  | Get all employees           |
| <b>GET</b>  | Get an employee by id       |
| <b>POST</b>   | Add new employee            |
| <b>PUT</b>  | Update an employee          |
| <b>DEL</b>  | Delete an employee          |

Let me know if you have any questions or concerns. Cheers!