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.

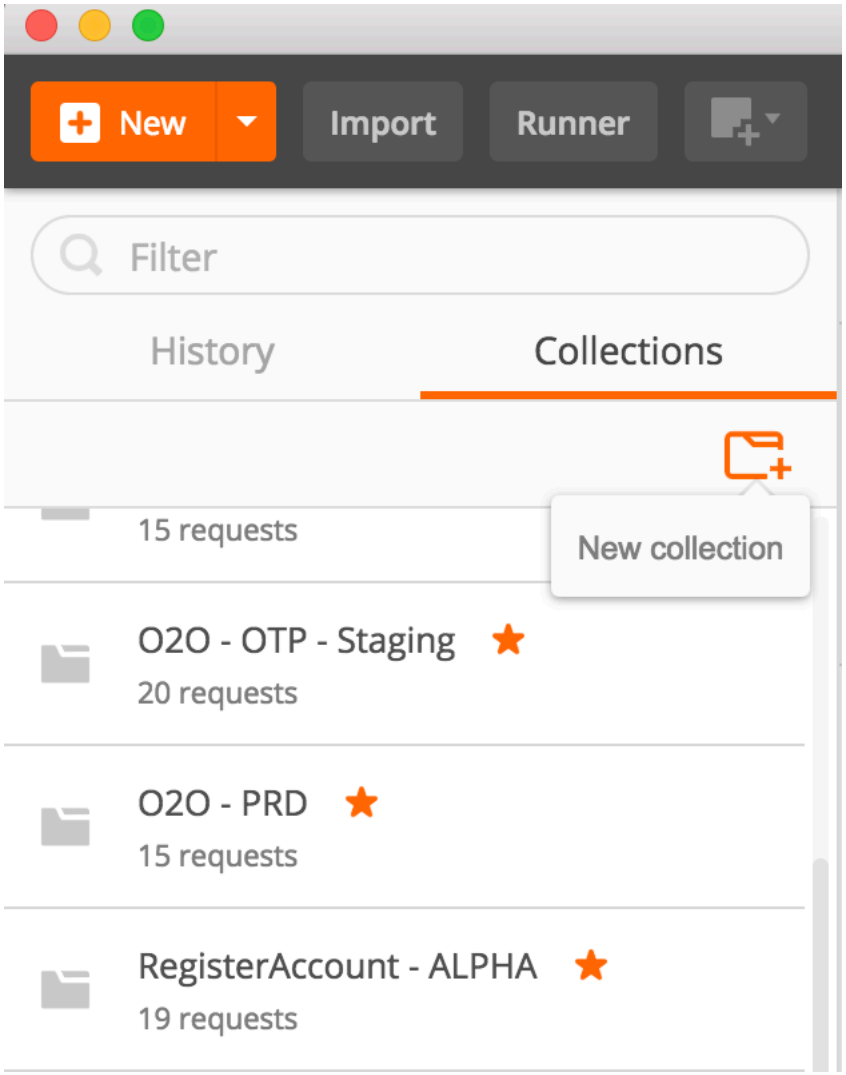
URL	http://sangbui.com/api/v1/employees
Method	GET
URL Params	<none>
Header	Content-Type: application/json Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==
Data Params	<none>

Success Response	Code: **200 OK**
	Content:
	[
	{
	"id": "<id>",
	"employee_name": "<name>",
	"employee_salary": "<salary>",
	"employee_age": "<age>"
	},
	{
	"id": "<id>",
	"employee_name": "<name>",
	"employee_salary": "<salary>",
	"employee_age": "<age>"
	}
	]

Error Response	Code: **404 Not Found**
Sample Call	<none>
Notes	<none>

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

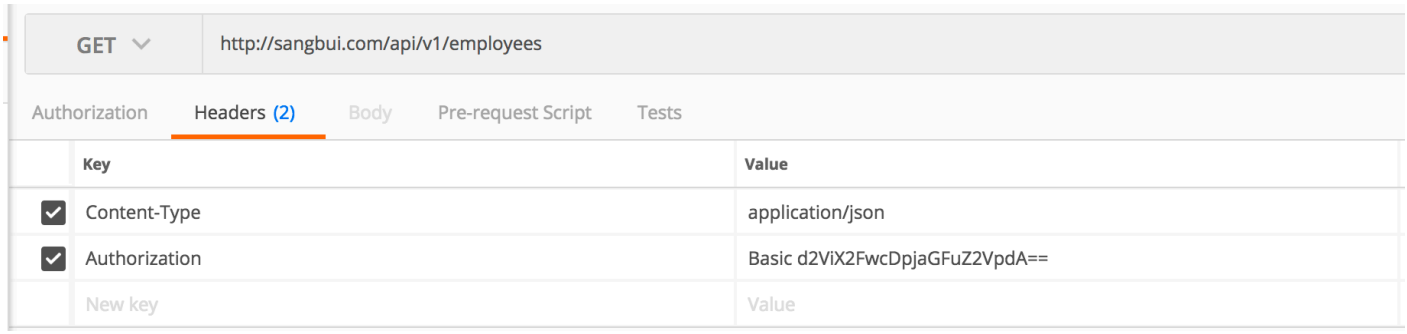
1. Open the Postman, create new Collection by click on **New Collection** icon. Enter the collection name.



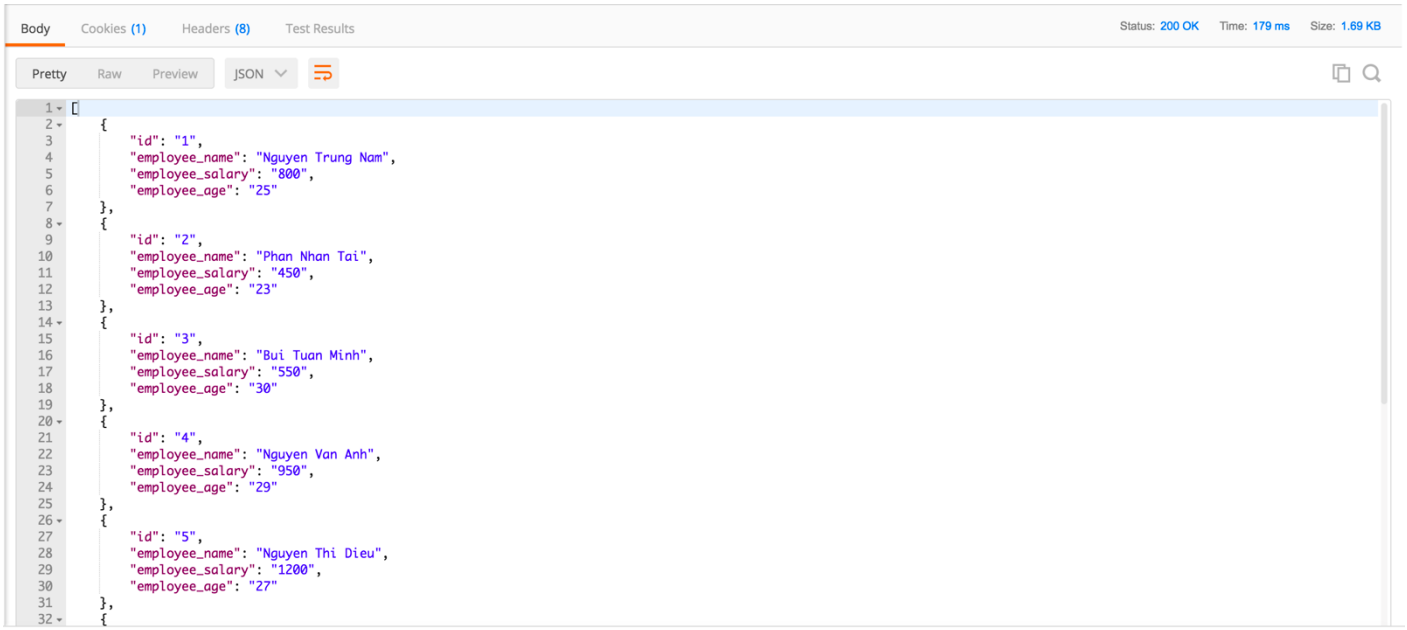
2. Enter the URL at "**Enter request URL**" - as the above requirement, the URL should be "http://sangbui.com/api/v1/employees".

   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.

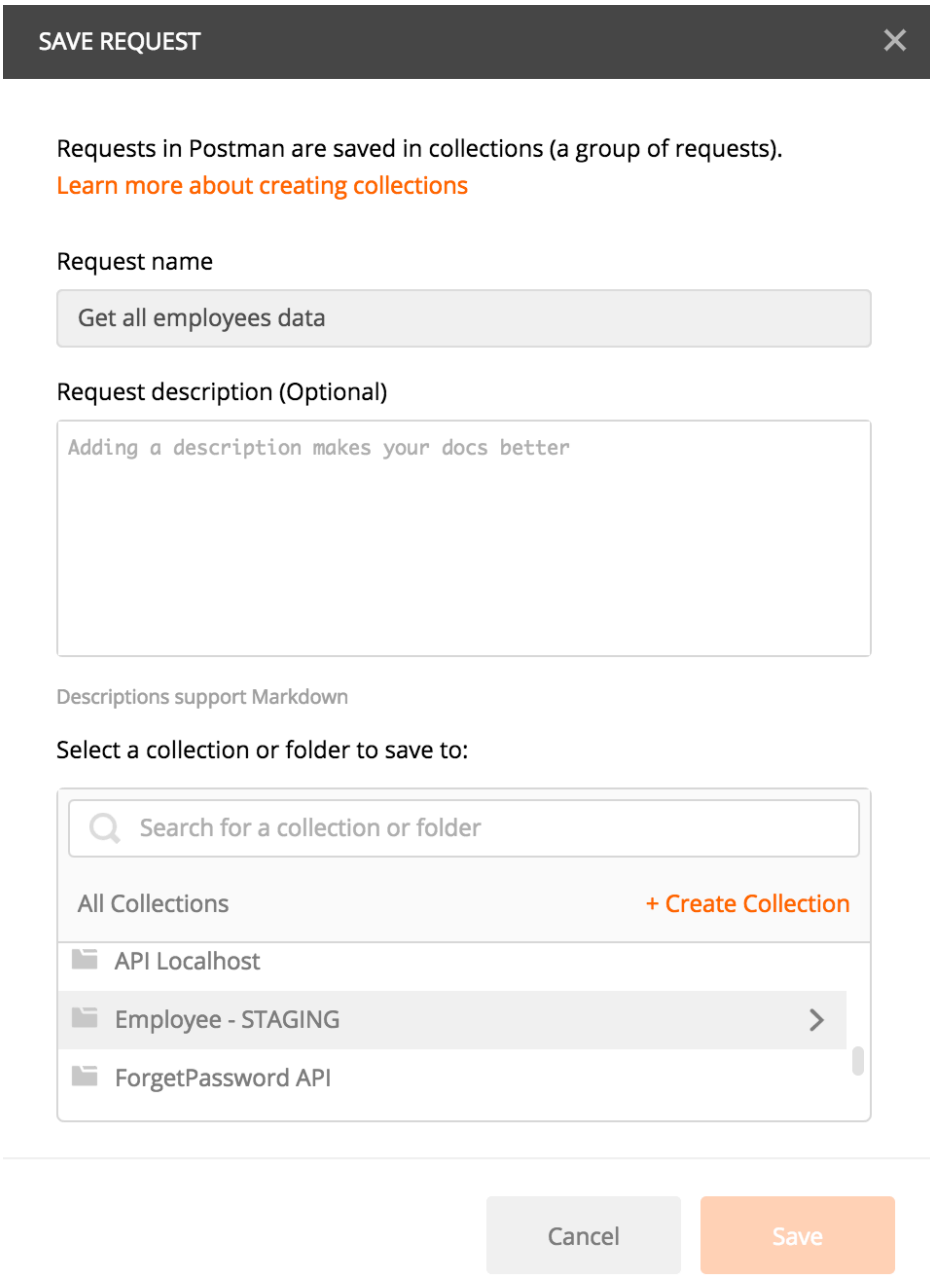


Now you can try to click on **Send** button again, it will show the successful code **200 OK** and list all employees data.



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:
Search for a collection or folder
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

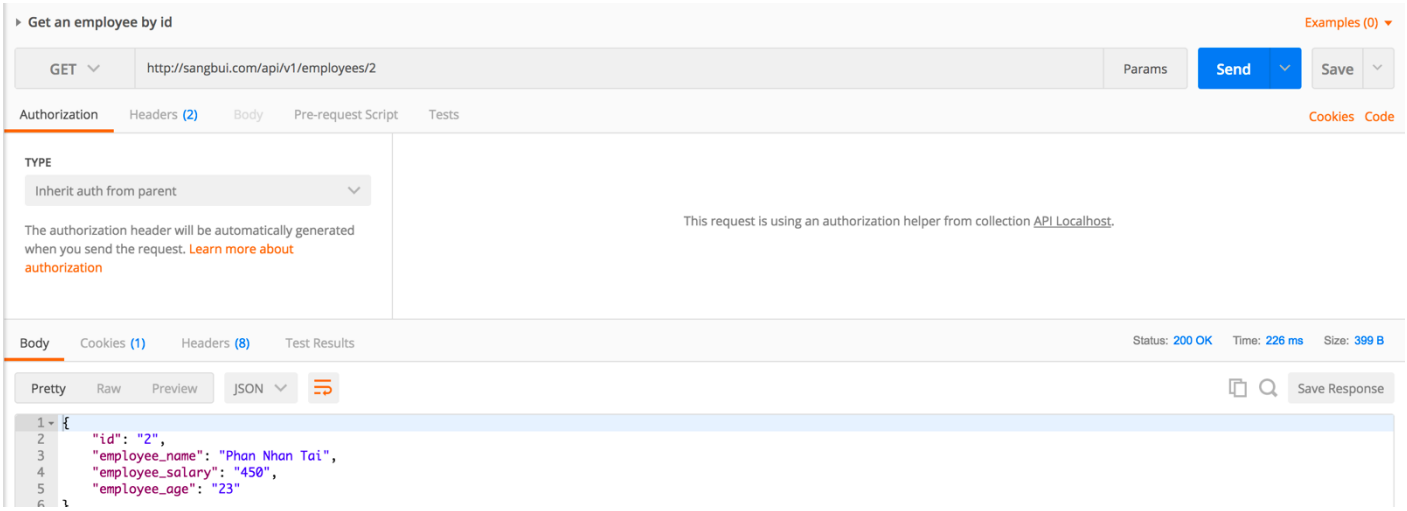
URL	http://sangbui.com/api/v1/employees/**<employee_id>**
Method	GET
URL Params	<employee_id>
Header	Content-Type: application/json Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==
Data Params	<none>

Success Response	Code: **200 OK**
	Content:
	[
	{
	"id": "<id>",
	"employee_name": "<name>",
	"employee_salary": "<salary>",
	"employee_age": "<age>"
	}
	]

Error Response	Code: **404 Not Found**
Sample Call	http://sangbui.com/api/v1/employees/**1**
Notes	In our database, the employee_id from **1 to 15** 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>**



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	http://sangbui.com/api/v1/employees
Method	POST
URL Params	<none>
Header	Content-Type: application/json Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==

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

Success Response	Code: **200 OK**
	Content:
	{
	"status": 1,
	"status_message": "Employee added successfully!"
	}

Error Response	Code: **404 Not Found**

Sample Call	{
	"employee_name" : "An",
	"employee_salary": "1000",
	"employee_age": "29"
	}

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

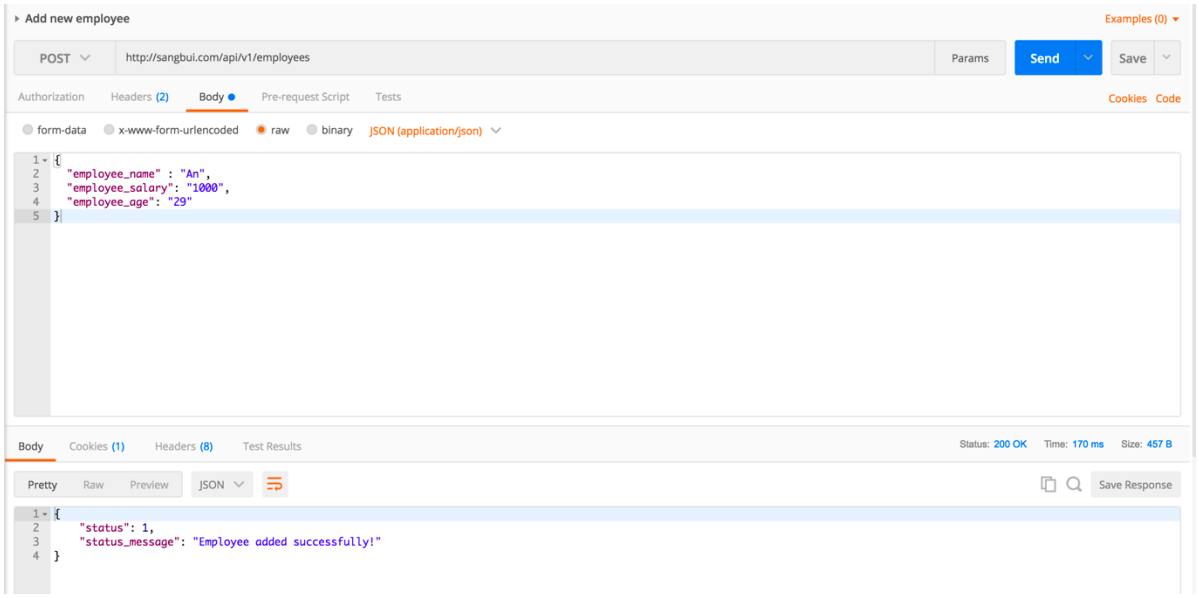
```
{

  "employee_name" : "<name>",

  "employee_salary": "<salary>",

  "employee_age": "<age>"

}
```



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 Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==

Data Params	{
	"id": "<id>",
	"employee_name": "<new_name>",
	"employee_salary": "<new_salary>",
	"employee_age": "<new_age>"
	}

Success Response	**Code: 200 OK**
	Content:
	{
	"status": 1,
	"status_message": "Employee updated successfully."
	}

Error Response	**Code: 404 Not Found**

Sample Call	{
	"id" : 7,
	"employee_name" : "Nam",
	"employee_salary": "800",
	"employee_age": "25"
	}

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 Authorization: Basic d2ViX2FwcDpjaGFuZ2VpdA==

Data Params	{
	"id" : <id>
	}

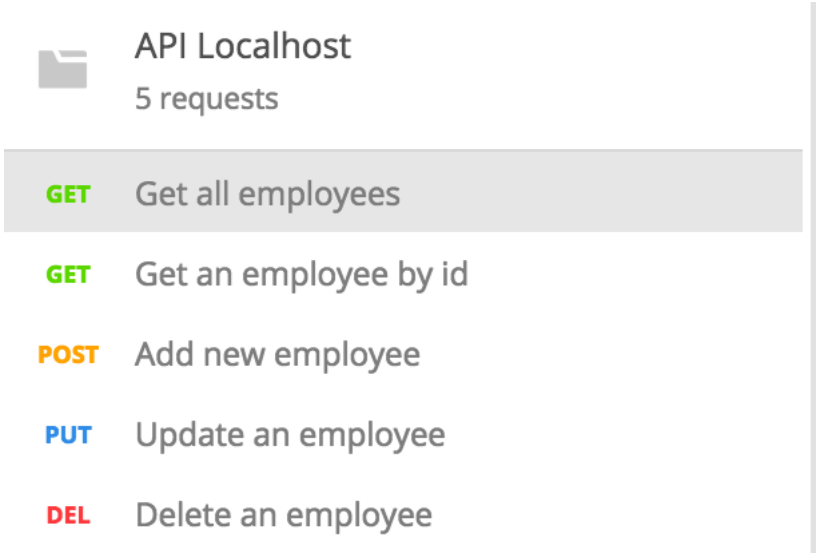
Success Response	**Code: 200 OK**
	Content:
	{
	"status": 1,
	"status_message": "Employee deleted successfully."
	}

Error Response	**Code: 404 Not Found**

Sample Call	{
	"id" : 5
	}

Notes	<none>

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



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