SESSION 7

Flask REST API

Request Get, Insert, Patch, and Delete Data

Data Science Program

Purwadhika
Startup and Coding School

SESSION 7

Flask REST API

Request Get Data

Purwadhika
Startup and Coding School

REQUEST TO GET DATA

1. Open Jupyter Notebook
2. Import module “request”

```
# import library yang ingin digunakan
import requests
```

3. Make a request to get data on the employee table in the flask_mysql database. Change the result of the request into json form before it is displayed

```
# GET ALL EMPLOYEE
# request ini di tujukan untuk mengambil seluruh data employee
result = requests.get('http://localhost:5000/employee')
# karena response dari API dalam bentuk json
# kita harus menggunakan function json untuk memproses datanya
result = result.json()
print(result)
```

Purwadhika
Startup and Coding School

REQUEST TO GET DATA

```python
# import library yang ingin digunakan
import requests
```

```python
# GET ALL EMPLOYEE
# request ini di tujukan untuk mengambil seluruh data employee
result = requests.get('http://localhost:5000/employee')
# karena response dari API dalam bentuk json
# kita harus menggunakan function json untuk memproses datanya
result = result.json()
print(result)
```

```
[{'gender': 'M', 'id': 1, 'married': 0, 'name': 'Rochafi Alvin', 'username': 'rochafi'}, {'gender': 'F', 'id': 2, 'married': 1,
'name': 'Alvin Rochafi', 'username': 'alvin'}, {'gender': 'M', 'id': 3, 'married': 1, 'name': 'John Wick', 'username': 'john'}]
```

Purwadhika
Startup and Coding School

SESSION 7

Flask REST API

Request Insert Data

Purwadhika
Startup and Coding School

REQUEST TO POST DATA

1. Create a variable as a dictionary type that contains the variables in the table of the database.

```
# POST ONE EMPLOYEE

# data yang akan disimpan
employee = {
    "username" : "deno",
    "name" : "Ryan Node",
    "gender" : "M",
    "married" : True
}
```

Purwadhika
Startup and Coding School

REQUEST TO POST DATA

2. Make a request to post to the localhost or employee database. After that, display the response that was previously created on the route.

```
# melakukan proses request ke API dan mendapatkan response
# response dari API akan disimpan ke variable result
result = requests.post(
    'http://localhost:5000/employee',
    data = employee
)

# karena response dikirim dalam bentuk json
# kita gunakan function json untuk meng-extract
result = result.json()

print(result)
```

Purwadhika
Startup and Coding School

REQUEST TO POST DATA

```
# UPDATE ONE EMPLOYEE

# data yang akan disimpan
employee = {
    "username" : "notebook",
    "name" : "Ryan Note Edited",
    "gender" : "M",
    "married" : False
}

# melakukan proses request ke API dan mendapatkan response
# response dari API akan disimpan ke variable result
result = requests.patch(
    'http://localhost:5000/employee',
    data = employee
)

# karena response dikirim dalam bentuk json
# kita gunakan function json untuk meng-extract
result = result.json()

print(result)
```

```
{'message': 'Update Success'}
```

Purwadhika
Startup and Coding School

SESSION 7

Flask REST API

Request Patch Data

Purwadhika
Startup and Coding School

REQUEST TO PATCH DATA

1. Create a variable as dictionary type that contains the appropriate variables in the table in the database. The contents of these variables are the latest updates to the "id" which will later be addressed

```
# UPDATE ONE EMPLOYEE WITH ID

# data yang akan disimpan
employee = {
    "username" : "notebook",
    "name" : "Ryan Note Edited",
    "gender" : "M",
    "married" : False
}
```

Purwadhika
Startup and Coding School

REQUEST TO PATCH DATA

2. Make a request to patch (update) to the localhost or employee database with "id" as the benchmark to be updated. After that, display the response that was previously created on the router.

```
# melakukan proses request ke API dan mendapatkan response
# response dari API akan disimpan ke variable result
# melakukan update untuk employee dengan id = 9
result = requests.patch(
    'http://localhost:5000/employee/9',
    data = employee
)

# karena response dikirim dalam bentuk json
# kita gunakan function json untuk meng-extract
result = result.json()

print(result)
```

Purwadhika
Startup and Coding School

REQUEST TO PATCH DATA

```
# UPDATE ONE EMPLOYEE WITH ID

# data yang akan disimpan
employee = {
    "username" : "notebook",
    "name" : "Ryan Note Edited",
    "gender" : "M",
    "married" : False
}

# melakukan proses request ke API dan mendapatkan response
# response dari API akan disimpan ke variable result
# melakukan update untuk employee dengan id = 9
result = requests.patch(
    'http://localhost:5000/employee/9',
    data = employee
)

# karena response dikirim dalam bentuk json
# kita gunakan function json untuk meng-extract
result = result.json()

print(result)
```

```
{'message': 'Update Success', 'user_id': '9'}
```

Purwadhika
Startup and Coding School

SESSION 7

Flask REST API

Request Delete Data

Purwadhika
Startup and Coding School

REQUEST TO DELETE DATA

1. Lakukan request untuk menghapus (delete) data dari database / localhost dengan menjadikan “id” sebagai referensi data yang akan dihapus

```
# DELETE ONE EMPLOYEE WITH ID

# melakukan proses request ke API dan mendapatkan response
# response dari API akan disimpan ke variable result
# melakukan delete untuk employee dengan id = 9
result = requests.delete(
    'http://localhost:5000/employee/9'
)
```

2. Tampilkan response yang telah dibuat sebelumnya

```
# karena response dikirim dalam bentuk json
# kita gunakan function json untuk meng-extract
result = result.json()

print(result)
```

Purwadhika
Startup and Coding School

REQUEST TO DELETE DATA

```
# DELETE ONE EMPLOYEE WITH ID

# melakukan proses request ke API dan mendapatkan response
# response dari API akan disimpan ke variable result
# melakukan delete untuk employee dengan id = 9
result = requests.delete(
    'http://localhost:5000/employee/9'
)

# karena response dikirim dalam bentuk json
# kita gunakan function json untuk meng-extract
result = result.json()

print(result)
```

```
{'message': 'Delete Success', 'user_id': '9'}
```

Purwadhika
Startup and Coding School