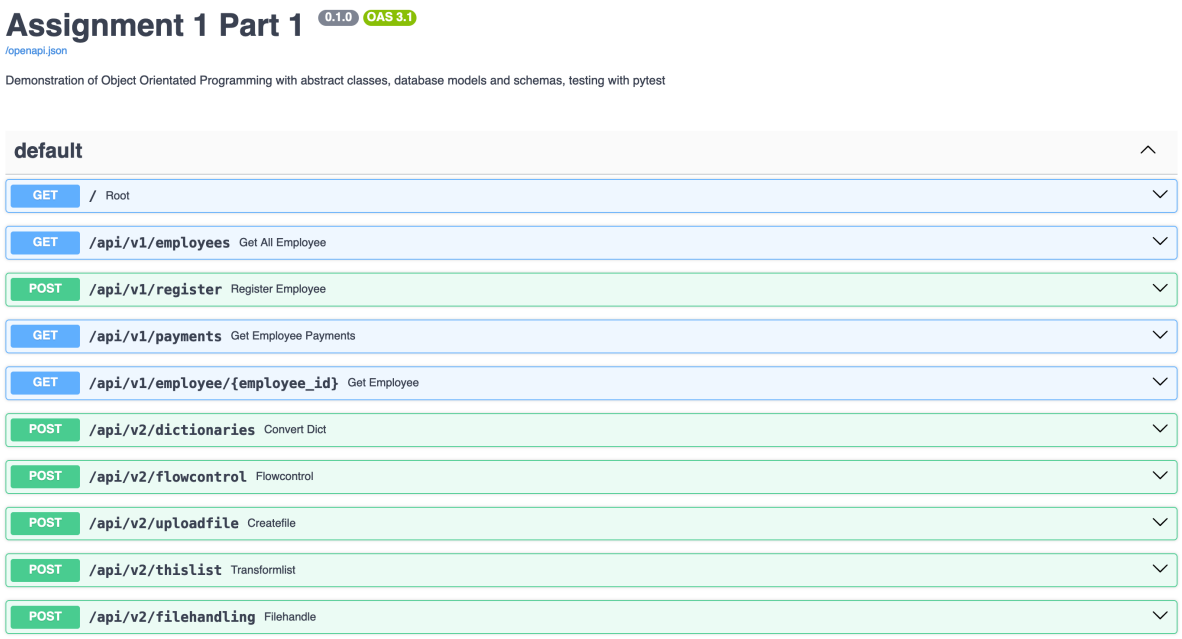
Assignment 1
Uvicorn, Response Body, and pytest output documentation

Table of Contents

Assignment 1
Uvicorn, Response Body, and pytest output documentation

Part 1



This exercise contains randomly generated data, produced by loadData.py. For running pytest, there is a separate database that needs to be empty for each run.

Part 1.1 Response Body

```
{
  "a": 100,
  "b": 200,
  "c": 300
}
INFO:     127.0.0.1:50900 - "POST /api/v2/dictionaries HTTP/1.1" 200 OK
```

Part 1.2 Response Body

```
in this case n = 4
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
in this case n = 1
*
* *
*
INFO:     127.0.0.1:50904 - "POST /api/v2/flowcontrol HTTP/1.1" 200 OK
```

Part 1.3 Response Body

```
[
  "e",
  "d",
  "c",
  "z",
  "b",
  "a",
  "o"
]
INFO:     127.0.0.1:50916 - "POST /api/v2/thislist HTTP/1.1" 200 OK
```

Assignment 1

Uvicorn, Response Body, and pytest output documentation

Part 1.4 Response Body

```
---
+++
@@ -1,15 +1,15 @@
-Lorem
-ipsum
-dolor
-sit
-amet,
-qui
-minim
-labore
-adipisicing
-minim
-sint
-cillum
-sint
-consectetur
-cupidatat.
+ 1: Lorem
+ 2: ipsum
+ 3: dolor
+ 4: sit
+ 5: amet,
+ 6: qui
+ 7: minim
+ 8: labore
+ 9: adipisicing
+10: minim
+11: sint
+12: cillum
+13: sint
+14: consectetur
+15: cupidatat.
INFO:     127.0.0.1:50992 - "POST /api/v2/filehandling HTTP/1.1" 200 OK
```

Part 1.5 Response Body

Note on responses: all contracts are concrete implementations of the abstract class Contract and the model type contains the details in JSON. The code that calculates the payment: `temp = ContractModel(**contract).execute_contract().get_payment()`

The failure of the get to find the employee is a problem I did not solve, but rather did a sequential search instead. This is intended to fulfil the print requirement of the exercise.

The following is generated with the registration of a new employee with inadequate salary:

```
{
  "detail": "salary below threshold"
}
Sqlite Connection successful
INFO:     127.0.0.1:49217 - "POST /api/v1/register HTTP/1.1" 400 Bad Request
```

Print employee and calculate payment:

```
{
  "employee": {
    "emp_number": "KLD2439",
    "full_name": "Darnbrook, Koenraad",
    "salarly": 11653100,
    "contract_type": {
      "contract": {
        "ctype": "salaried",
        "salary": 14468700,
        "hours": 78
      }
    },
    "updated_at": "2018-11-30T13:49:13"
  },
  "contract_type": "Salaried Contract",
  "payment": "$225,711.72"
}
Sqlite Connection successful
```

Assignment 1

Uvicorn, Response Body, and pytest output documentation

```
couldn't find 9e1550e-0604-4d8a-a640-47114ada2278 - will try sequential search
comparing fd6d756c-67ab-48e5-8942-173a34851955 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 27ec76f4-a1df-4f0a-9ffc-2a779f1bbdb0 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing bab12f50-9000-4f48-a647-94b5a51b5859 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 3711f522-7f48-401c-9417-972c4453441e to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 39fda92f-46ee-44c0-8974-4ce99da064e3 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 365c3687-c299-4881-af25-fbed2efae554 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 1646ddac-d7bb-4a90-8dc8-aa3ab75caa55 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing a5dc5a97-ab62-49b3-a2f4-666949ac96f5 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 3622a419-f0f2-4b0d-b2de-f3dd75c57ff5 to 9e1550e-0604-4d8a-a640-
47114ada2278
comparing 9e1550e-0604-4d8a-a640-47114ada2278 to 9e1550e-0604-4d8a-a640-
47114ada2278
found it
INFO:     127.0.0.1:49229 - "GET /api/v1/employee/9e1550e-0604-4d8a-a640-
47114ada2278 HTTP/1.1" 200 OK
```

Part 1.6 Response Body

See 1.5

Part 1.7 Testing

```
platform darwin -- Python 3.12.3, pytest-8.2.1, pluggy-1.5.0
rootdir: /Volumes/WDmini/PythonWorkspace/30 AWS
configfile: pytest.ini
plugins: asyncio-0.23.7, anyio-4.4.0
asyncio: mode=Mode.STRICT
collected 7 items

test_Part1.py ....
[ 57%]
test_api.py ...
[100%]


============================== 7 passed in 1.60s ===============================
```

Assignment 1
Uvicorn, Response Body, and pytest output documentation

Part 2

Part 2.1

An example of a Dict[‘message key’, Message] is commented in code. I used a List[Message] with an id for primary key.
The autoincrement and overwrite protection:

```
INFO:     attempt to get message with message_id 1
ERROR:    attempt to create message with id 1 - already exists.
Auto-incrementing - new id is 4
INFO:     127.0.0.1:50536 - "POST /api/create HTTP/1.1" 201 Created
```

Part 2.2

```
INFO:     get all messages (count: 3)
INFO:     127.0.0.1:50531 - "GET /api/get_messages HTTP/1.1" 200 OK
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     get all messages (count: 0)
ERROR:    no messages
INFO:     127.0.0.1:50499 - "GET /api/get_messages HTTP/1.1" 404 Not Found
```

Part 2.3

```
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     attempt to get message with message_id 6
ERROR:    message 6 not found
INFO:     127.0.0.1:50504 - "GET /api/get/6 HTTP/1.1" 404 Not Found
```

Part 2.4

```
INFO:     127.0.0.1:50674 - "PATCH /api/update/5 HTTP/1.1" 404 Not Found
INFO:     old message id=2 to_address='Ahmed@unitec.ac.nz'
from_address='Fran@unitec.ac.nz' copy_to='' subject='This is the subject of the
message' text='This is the text of the message'
attachment='file://thisherefile.txt'
INFO:     new message id=2 to_address='dca@gmail.com'
from_address='Fran@unitec.ac.nz' copy_to='' subject='This is the subject of the
message' text='This is the text of the message' attachment='new attachment and new
to_address'
INFO:     127.0.0.1:50689 - "PATCH /api/update/2 HTTP/1.1" 200 OK
```

Part 2.5

```
ERROR:    attempt to delete message 5  - not found
INFO:     127.0.0.1:50696 - "DELETE /api/delete/5 HTTP/1.1" 404 Not Found
INFO:     successful attempt to delete message id=1 to_address='Ahmed@unitec.ac.nz'
from_address='jam@gmail.com' copy_to='' subject='Lorem ipsum' text='Lorem ipsum
dolor sit amet, qui minim labore adipisicing minim sint cillum sint consectetur
cupidatat.' attachment=''
INFO:     127.0.0.1:50700 - "DELETE /api/delete/1 HTTP/1.1" 204 No Content
```

Part 2.6

```
==================================== test session starts ====================================
platform darwin -- Python 3.12.3, pytest-8.2.1, pluggy-1.5.0
rootdir: /Volumes/WDmini/PythonWorkspace/30 AWS
configfile: pytest.ini
plugins: asyncio-0.23.7, anyio-4.4.0
asyncio: mode=Mode.STRICT
collected 6 items

test_api.py ......

===================================== 6 passed in 0.65s =====================================
```

Part 3

This exercise contains randomly generated data, produced by loadData.py. For running pytest, there is a separate database that needs to be empty (truncated tables) for each run.

Part 3.1Response Body


```
{
  "owner": {
    "first_name": "David",
    "last_name": "Allen",
    "middle_name": "C",
    "email": "dca@example.com"
  },
  "cars": [
    {
      "id": 28,
      "owner_id": 26,
      "car_id": 12,
      "colour": "no colour",
      "vin": "very long vin",
      "plate_number": "IKFI9876",
      "purchased_dt": "2024-06-05"
    },
    {
      "id": 29,
      "owner_id": 26,
      "car_id": 13,
      "colour": "no colour",
      "vin": "very long vin",
      "plate_number": "IKFI9876",
      "purchased_dt": "2024-06-05"
    }
  ]
}
```


```
Sqlite Connection successful
INFO:     127.0.0.1:50418 - "POST /api/v1/register/ HTTP/1.1" 201 Created
```

Part 3.2 Response Body


```
{
  "owner": {
    "first_name": "Garner",
    "last_name": "Artingstall",
    "middle_name": "Bassford",
    "email": "gbassfordnl@yale.edu"
  },
  "cars": [
    {
      "id": 14,
      "owner_id": 22,
      "car_id": 19,
      "colour": "Turquoise",
      "vin": "MK12 7O7T OZOT XYF2 B77",
      "plate_number": "KFD77867",
      "purchased_dt": "2010-04-08"
    }
  ]
}
```


```
Sqlite Connection successful
INFO:     127.0.0.1:50426 - "GET /api/v1/owners-full-details22 HTTP/1.1" 200 OK
```

Part 3.3 Response Body


```
{
  "id": 27,
  "make": "Volkswagen",
  "model": "Quantum",
```

```
  "style": "convertible",
  "year": "1984",
  "updated_at": "2024-06-05T07:12:28",
  "created_at": "2024-06-05T07:12:28"
}

Sqlite Connection successful
INFO:     127.0.0.1:50430 - "POST /api/v1/carregister/ HTTP/1.1" 201 Created
```


Part 3.4 Response Body


```
{
  "car": {
    "make": "Dodge",
    "model": "Caliber",
    "style": "EV",
    "year": "2009"
  },
  "owners": [
    {
      "owner": {
        "first_name": "Angele",
        "last_name": "Gurner",
        "middle_name": "Prevost",
        "email": "aprevostl0@mail.ru"
      },
      "cars": [
        {
          "id": 8,
          "owner_id": 10,
          "car_id": 22,
          "colour": "Fuscia",
          "vin": "AZ70 ALNB PEX0 0AJ6 DXAW SDJE IA7C",
          "plate_number": "KFD67007",
          "purchased_dt": "1991-10-13"
        },
        {
          "id": 23,
          "owner_id": 10,
          "car_id": 14,
          "colour": "Violet",
          "vin": "SI17 7079 6969 5138 876",
          "plate_number": "YY23896",
          "purchased_dt": "2010-03-31"
        }
      ]
    }
  ]
}
Sqlite Connection successful
INFO:     127.0.0.1:50435 - "GET /api/v1/cars-full-details14 HTTP/1.1" 200 OK
```


Part 3.5 Response Body


```
{
  "plate_number": "new plate"
}
Sqlite Connection successful
{'plate_number': 'new plate'}
INFO:     127.0.0.1:50751 - "PATCH /api/v1/ownscar/12 HTTP/1.1" 206 Partial Content
```


Part 3.6

```
================================== test session starts ==================================
platform darwin -- Python 3.12.3, pytest-8.2.1, pluggy-1.5.0
rootdir: /Volumes/WDmini/PythonWorkspace/30 AWS
configfile: pytest.ini
plugins: asyncio-0.23.7, anyio-4.4.0
asyncio: mode=Mode.STRICT
collecting ... Initializing package from assigment 1 part 3 models...
Initializing package from assigment 1 part 3 schemas...
collected 7 items

test_api.py .......

=================================== 7 passed in 1.23s ===================================
```