ДЗ\_3 Postman

=====

=====

1) необходимо залогиниться

POST

http://162.55.220.72:5005/login

login : str (кроме /)

password : str

Приходящий токен необходимо передать во все остальные запросы.

===================

дальше все запросы требуют наличие токена.

===================

2) http://162.55.220.72:5005/user\_info

req. (RAW JSON)

POST

age: int

salary: int

name: str

auth\_token

resp.

{'start\_qa\_salary':salary,

'qa\_salary\_after\_6\_months': salary \* 2,

'qa\_salary\_after\_12\_months': salary \* 2.9,

'person': {'u\_name':[user\_name, salary, age],

'u\_age':age,

'u\_salary\_1.5\_year': salary \* 4}

}

Тесты:

1) Статус код 200

2) Проверка структуры json в ответе.

3) В ответе указаны коэффициенты умножения salary, напишите тесты по проверке правильности результата перемножения на коэффициент.

4) Достать значение из поля 'u\_salary\_1.5\_year' и передать в поле salary запроса http://162.55.220.72:5005/get\_test\_user

===================

3) http://162.55.220.72:5005/new\_data

req.

POST

age: int

salary: int

name: str

auth\_token

Resp.

{'name':name,

'age': int(age),

'salary': [salary, str(salary\*2), str(salary\*3)]}

Тесты:

1) Статус код 200

2) Проверка структуры json в ответе.

3) В ответе указаны коэффициенты умножения salary, напишите тесты по проверке правильности результата перемножения на коэффициент.

4) проверить, что 2-й элемент массива salary больше 1-го и 0-го

===================

4) http://162.55.220.72:5005/test\_pet\_info

req.

POST

age: int

weight: int

name: str

auth\_token

Resp.

{'name': name,

'age': age,

'daily\_food':weight \* 0.012,

'daily\_sleep': weight \* 2.5}

Тесты:

1) Статус код 200

2) Проверка структуры json в ответе.

3) В ответе указаны коэффициенты умножения weight, напишите тесты по проверке правильности результата перемножения на коэффициент.

===================

5) http://162.55.220.72:5005/get\_test\_user

req.

POST

age: int

salary: int

name: str

auth\_token

Resp.

{'name': name,

'age':age,

'salary': salary,

'family':{'children':[['Alex', 24],['Kate', 12]],

'u\_salary\_1.5\_year': salary \* 4}

}

Тесты:

1) Статус код 200

2) Проверка структуры json в ответе.

3) Проверить что занчение поля name = значению переменной name из окружения

4) Проверить что занчение поля age в ответе соответсвует отправленному в запросе значению поля age

===================

6) http://162.55.220.72:5005/currency

req.

POST

auth\_token

Resp. Передаётся список массив объектов.

[

{"Cur\_Abbreviation": str,

"Cur\_ID": int,

"Cur\_Name": str

}

…

{"Cur\_Abbreviation": str,

"Cur\_ID": int,

"Cur\_Name": str

}

]

Тесты:

1) Можете взять любой объект из присланного списка, используйте js random.

В объекте возьмите Cur\_ID и передать через окружение в следующий запрос.

===================

7) http://162.55.220.72:5005/curr\_byn

req.

POST

auth\_token

curr\_code: int

Resp.

{

"Cur\_Abbreviation": str

"Cur\_ID": int,

"Cur\_Name": str,

"Cur\_OfficialRate": float,

"Cur\_Scale": int,

"Date": str

}

Тесты:

1) Статус код 200

2) Проверка структуры json в ответе.

===============

\*\*\*

1) получить список валют

2) итерировать список валют

3) в каждой итерации отправлять запрос на сервер для получения курса каждой валюты

4) если возвращается 500 код, переходим к следующей итреации

5) если получаем 200 код, проверяем response json на наличие поля "Cur\_OfficialRate"

6) если поле есть, пишем в консоль инфу про фалюту в виде response

{

"Cur\_Abbreviation": str

"Cur\_ID": int,

"Cur\_Name": str,

"Cur\_OfficialRate": float,

"Cur\_Scale": int,

"Date": str

}

7) переходим к следующей итерации