DZ\_3 Postman

=====

1) you need to login

POST

http://162.55.220.72:5005/login

login : str (except /)

password : str

The incoming token must be passed to all other requests.

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

further all requests require the presence of a token.

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

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}

}

Tests:

1) Status code 200

2) Checking the json structure in the response.

3) The answer contains the coefficients of salary multiplication, write tests to check the correctness of the result of multiplication by the coefficient.

4) Get the value from the 'u\_salary\_1.5\_year' field and pass it to the salary field of the request 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)]}

Tests:

1) Status code 200

2) Checking the json structure in the response.

3) The answer contains the coefficients of salary multiplication, write tests to check the correctness of the result of multiplication by the coefficient.

4) check that 2nd element of salary array is greater than 1st and 0th

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

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}

Tests:

1) Status code 200

2) Checking the json structure in the response.

3) The answer contains the multiplication coefficients of weight, write tests to check the correctness of the result of multiplication by the coefficient.

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

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}

}

Tests:

1) Status code 200

2) Checking the json structure in the response.

3) Check that the value of the field name = the value of the variable name from the environment

4) Check that the value of the age field in the response matches the value of the age field sent in the request

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

6) http://162.55.220.72:5005/currency

req.

POST

auth\_token

Resp. A list of an array of objects is passed.

[

{"Cur\_Abbreviation": str,

Cur\_ID: int,

"Cur\_Name": str

}

…

{"Cur\_Abbreviation": str,

Cur\_ID: int,

"Cur\_Name": str

}

]

Tests:

1) You can take any object from the sent list, use js random.

Take the Cur\_ID in the object and pass it through the environment to the next request.

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

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

}

Tests:

1) Status code 200

2) Checking the json structure in the response.

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

\*\*\*

1) get a list of currencies

2) iterate the list of currencies

3) in each iteration, send a request to the server to get the rate of each currency

4) if 500 code is returned, go to the next iteration

5) if we get 200 code, check the response json for the presence of the "Cur\_OfficialRate" field

6) if there is a field, we write information about the currency in the console in the form of response

{

"Cur\_Abbreviation": str

Cur\_ID: int,

"Cur\_Name": str,

Cur\_OfficialRate: float

Cur\_Scale: int,

"Date": str

}

7) move on to the next iteration