**jkfjshfkjdhkhkf**

**Dskjfkjdlfjs;pewiptjks;;daskdp\wpip**

**Task 2. Postman, GIT.**

Postman.

**endpoint\_1**: object\_info\_1

method: GET

request\_params:

   1) age (int)

   2) weight (int)

   3) name (str)

response:

{'name': name,

          'age': age,

          'daily\_food': weight \* 0.012,

          'daily\_sleep': weight \* 2.5}

========

**endpoint\_2**: object\_info\_2

method: GET

request\_params:

   1) age (int)

   2) salary (int)

   3) name (str)

response:

{'start\_qa\_salary': salary,

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

          'qa\_salary\_after\_12\_months': salary \* 2.7,

          'qa\_salary\_after\_1.5\_year': salary \* 3.3,

          'qa\_salary\_after\_3.5\_years': salary \* 3.8,

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

                     'u\_age': age,

                     'u\_salary\_5\_years': salary \* 4.2}

          }

=========

**endpoint\_3**: object\_info\_3

method: GET

request\_params:

   1) age (int)

   2) salary (int)

   3) name (str)

response:

result = {'name': name,

          'age': age,

          'salary': salary,

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

                     'pets': {'cat':{'name':'Sunny',

                                     'age': 3},

                              'dog':{'name':'Luky',

                                     'age': 4}},

                     'u\_salary\_1\_5\_year': salary \* 4}

          }

=========

**endpoint\_4**: object\_info\_4

method: GET

request\_params:

   1) age (int)

   2) name (str)

   3) salary (int)

response:

result = {'name': name,

          'age': int(age),

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

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

**endpoint\_5**: user\_info\_1

method: POST

form\_params:

   1) age (int)

   2) weight (int)

   3) name (str)

response:

{'name': name,

          'age': age,

          'daily\_food': weight \* 0.012,

          'daily\_sleep': weight \* 2.5}

========

**endpoint\_6**: user\_info\_2

method: POST

form\_params:

   1) age (int)

   2) salary (int)

   3) name (str)

response:

{'start\_qa\_salary': salary,

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

          'qa\_salary\_after\_12\_months': salary \* 2.7,

          'qa\_salary\_after\_1.5\_year': salary \* 3.3,

          'qa\_salary\_after\_3.5\_years': salary \* 3.8,

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

                     'u\_age': age,

                     'u\_salary\_5\_years': salary \* 4.2}

          }

=========

**endpoint\_7**: user\_info\_3

method: POST

form\_params:

   1) age (int)

   2) salary (int)

   3) name (str)

response:

result = {'name': name,

          'age': age,

          'salary': salary,

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

                     'u\_salary\_1\_5\_year': salary \* 4}

          }

=========

**endpoint\_8**: user\_info\_4

method: POST

form\_params:

   1) age (int)

   2) name (str)

   3) salary (int)

response:

result = {'name': name,

          'age': int(age),

          'salary': [salary, str(salary \* 2.5), str(salary \* 3.5)]}

**Задание.  
1)** Выполнить в Postman endpoints 1,2,7,8.  
**2)** Сделать репозиторий в Гит  
**3)** Склонировать репозиторий на локальный компьютер.  
**4)** Експортировать тесты endpoints 1,2,7,8 в папку git склонированного репозитория.

**5)** Запушить тесты на внешний репозиторий  
**6)** Сделать в GIT отдельную ветку.  
**7)** Сделать тесты ендроинтов 3, 6 в Postman  
**8)** Кспортировать тесты в новую, созданную ветку GIT.  
**9)** Запушить новую ветку на внешний репозиторий. Тесты должны появиться на внешнем репозитории созданной в ветке  
**10)** Вмержить изменения новой ветки в основную.  
**11)** Запушить изменения основной ветки на внешний репозиторий.  
**12)** Сделать в GIT отдельную ветку.

**13**) Сделать тесты ендроинтов 4,5 в Postman  
**14**) Экспортировать тесты в созданную ветку GIT.  
**15**) Запушить новую ветку на внешний репозиторий. Тесты должны появиться на внешнем репозитории созданной в ветке.  
**16**) Вмержить изменения ветки в основную ветку.  
**17**) Запушить изменения основной ветки на внешний репозиторий.

**Теоретическое задание.**  
**1)** Какие есть виды тестирования. Расписать подробно о каждом.  
**2)** Что такое авторизация, аутентификация, валидация, идентификация.

**3)** Распишите жизненный цикл бага.