Создать запросы в Postman.

Protocol: http

IP: 162.55.220.72

Port: 5005

EP\_1

Method: GET

EndPoint: /get\_method

request url params:

 name: str

 age: int

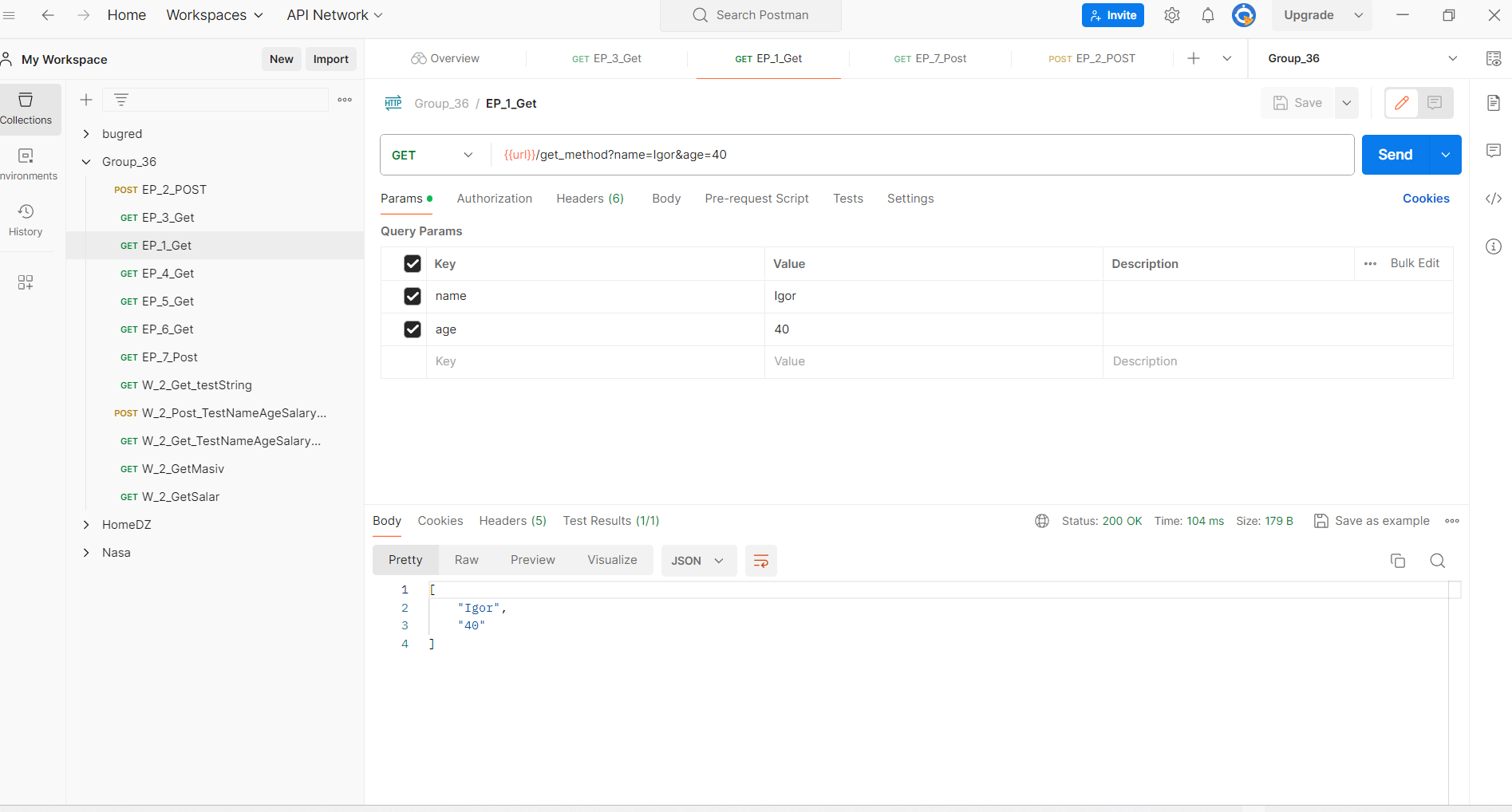
response:

[

    “Str”,

    “Str”

]



EP\_2

Method: POST

EndPoint: /user\_info\_3

request form data:

 name: str

 age: int

 salary: int

response:

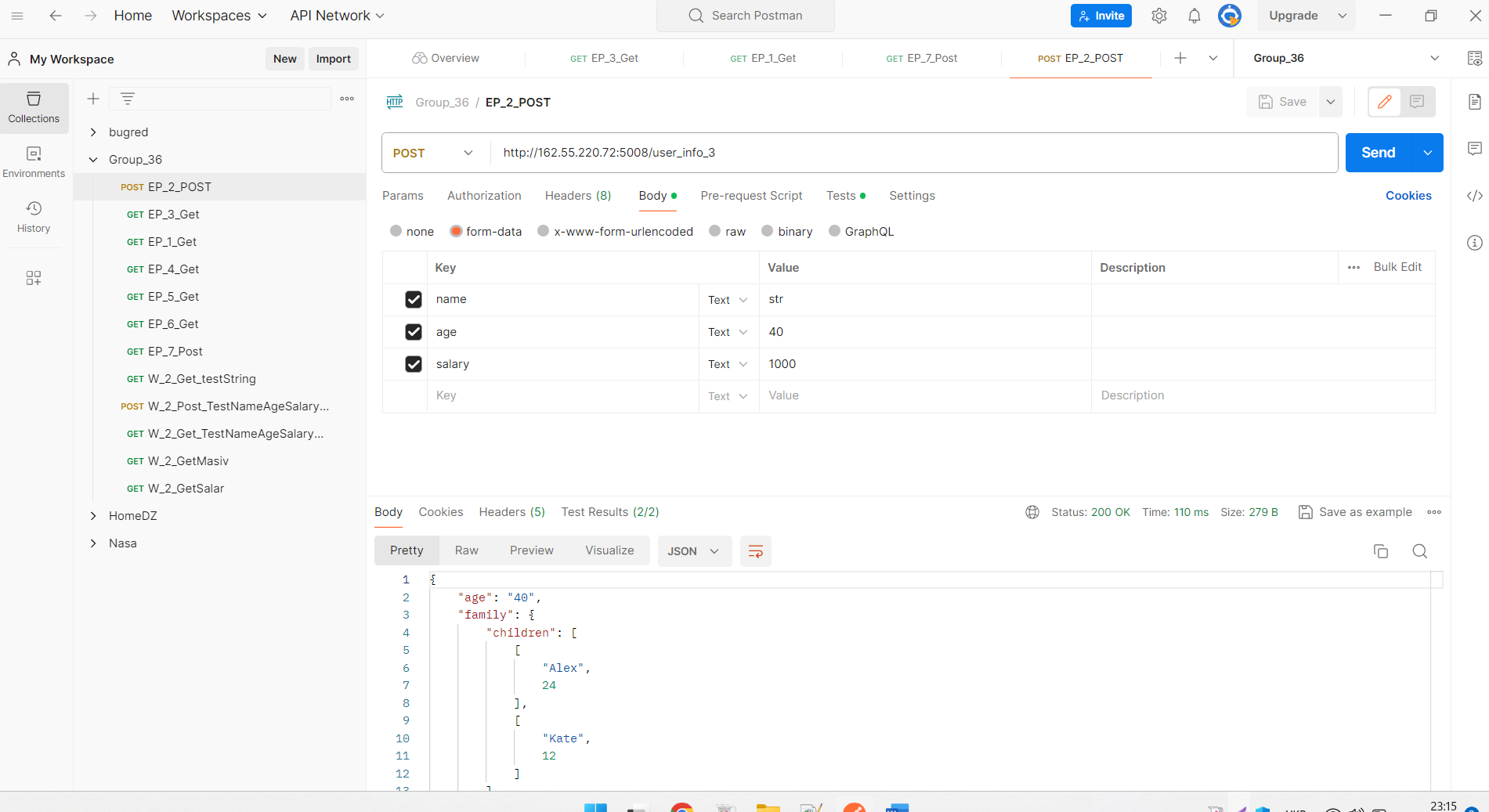
{'name': name,

          'age': age,

          'salary': salary,

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

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



EP\_3

Method: GET

EndPoint: /object\_info\_1

request url params:

 name: str

 age: int

 weight: int

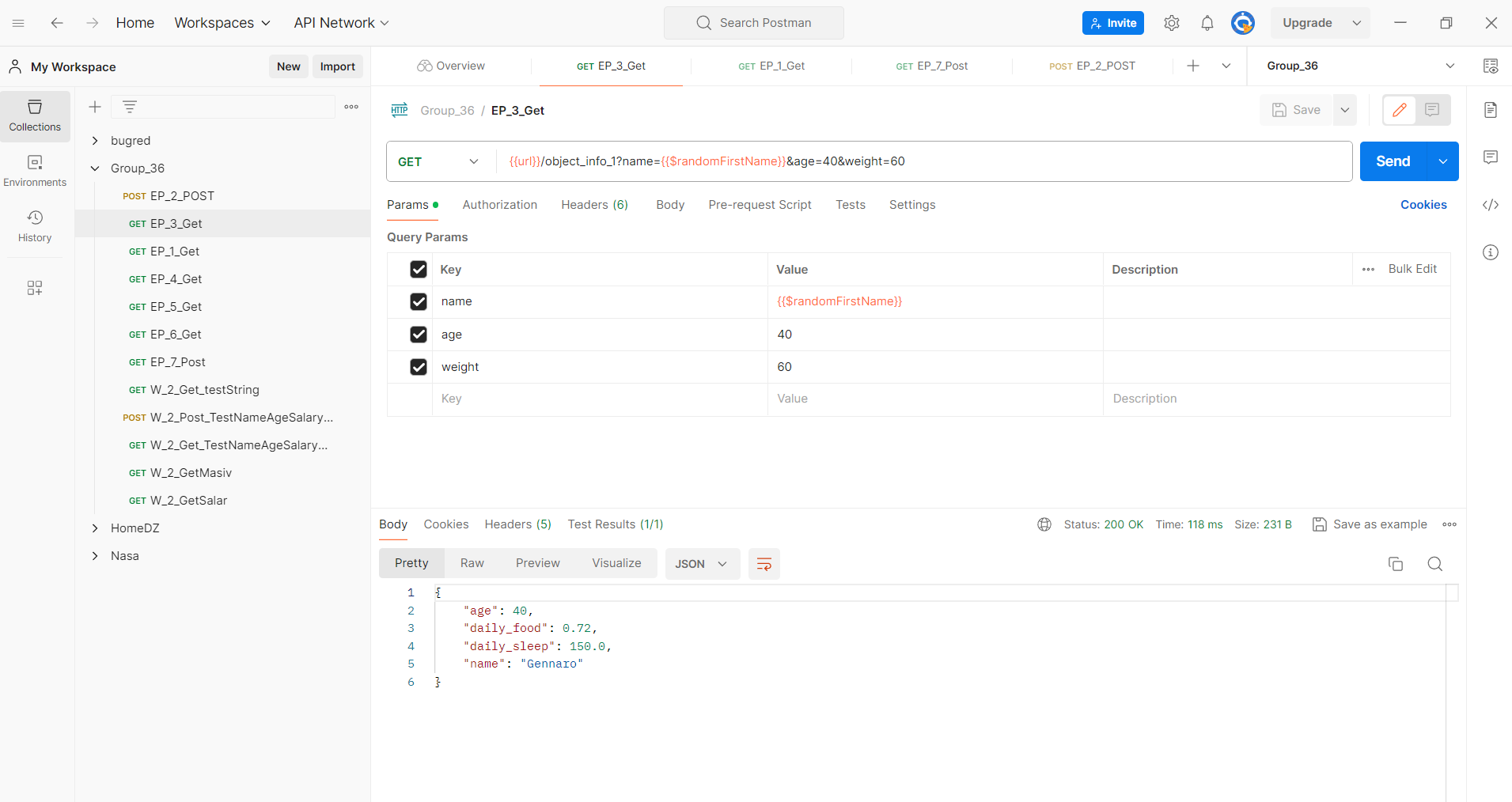
response:

{'name': name,

          'age': age,

          'daily\_food': weight \* 0.012,

          'daily\_sleep': weight \* 2.5}



EP\_4

Method: GET

EndPoint: /object\_info\_2

request url params:

 name: str

 age: int

 salary: int

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,

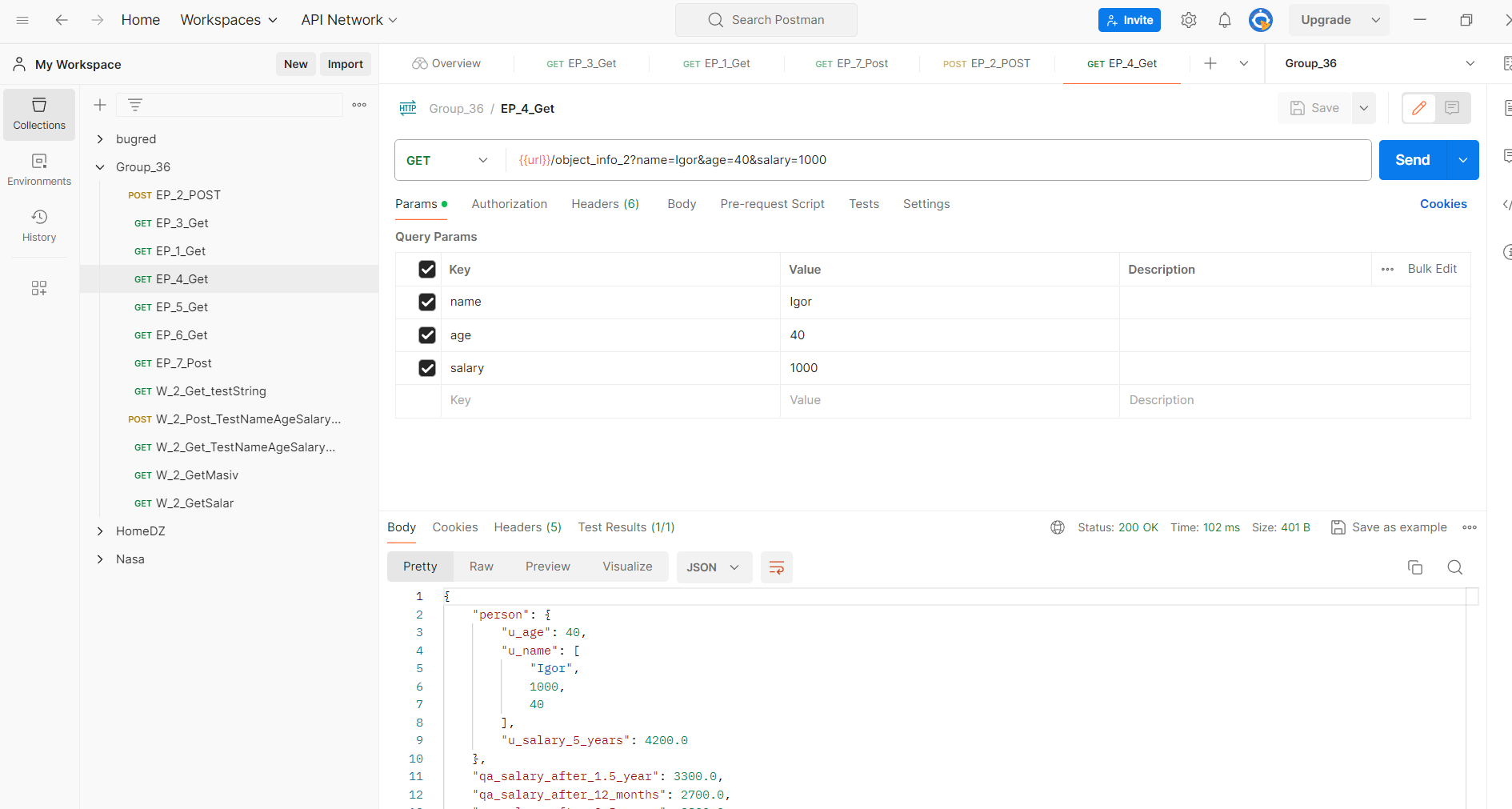
          'qa\_salary\_after\_3.5\_years': salary \* 3.8,

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

                     'u\_age': age,

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

          }



EP\_5

Method: GET

EndPoint: /object\_info\_3

request url params:

 name: str

 age: int

 salary: int

response:

{'name': name,

          'age': age,

          'salary': salary,

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

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

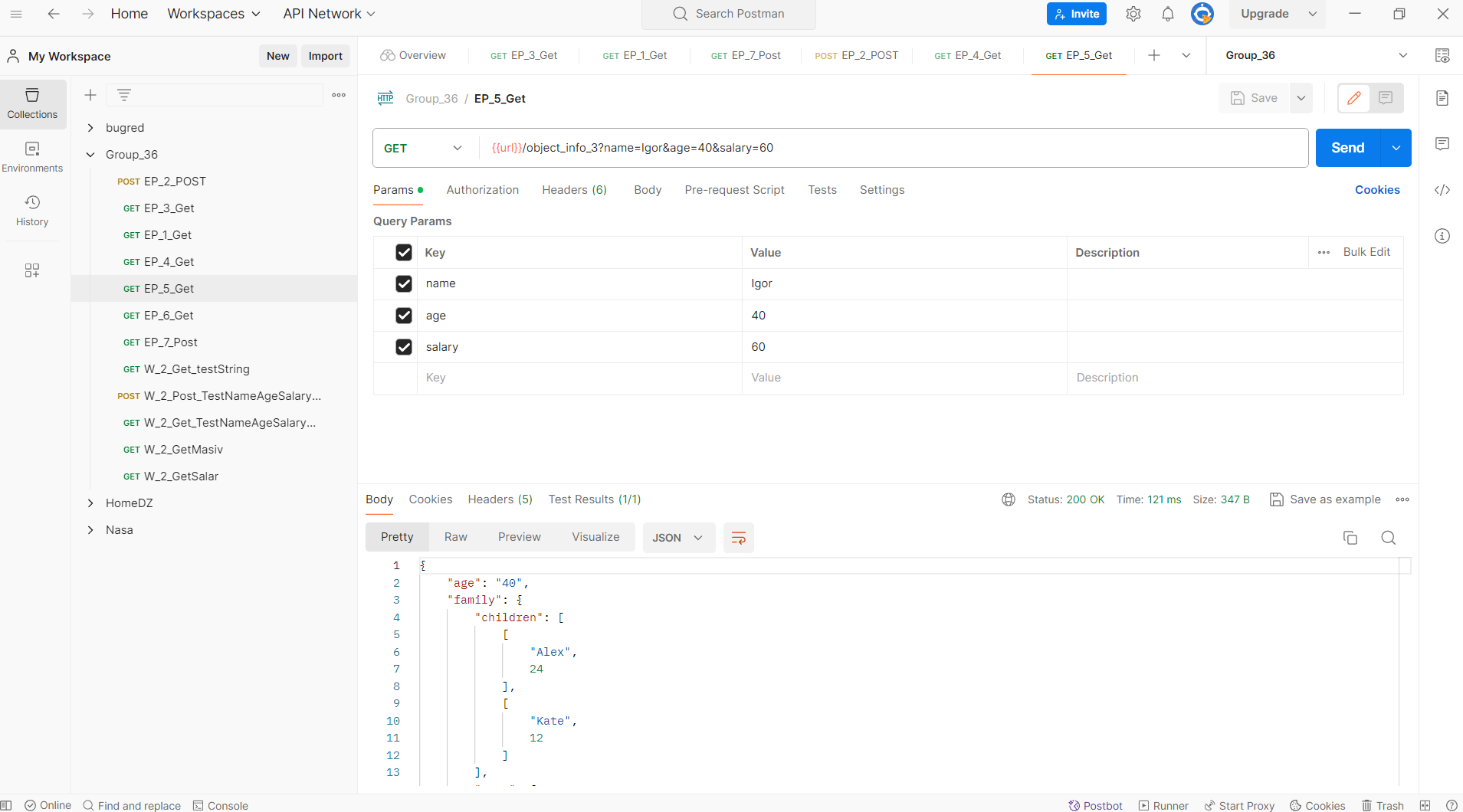
                                     'age': 3},

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

                                     'age': 4}},

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

          }



EP\_6

Method: GET

EndPoint: /object\_info\_4

request url params:

 name: str

 age: int

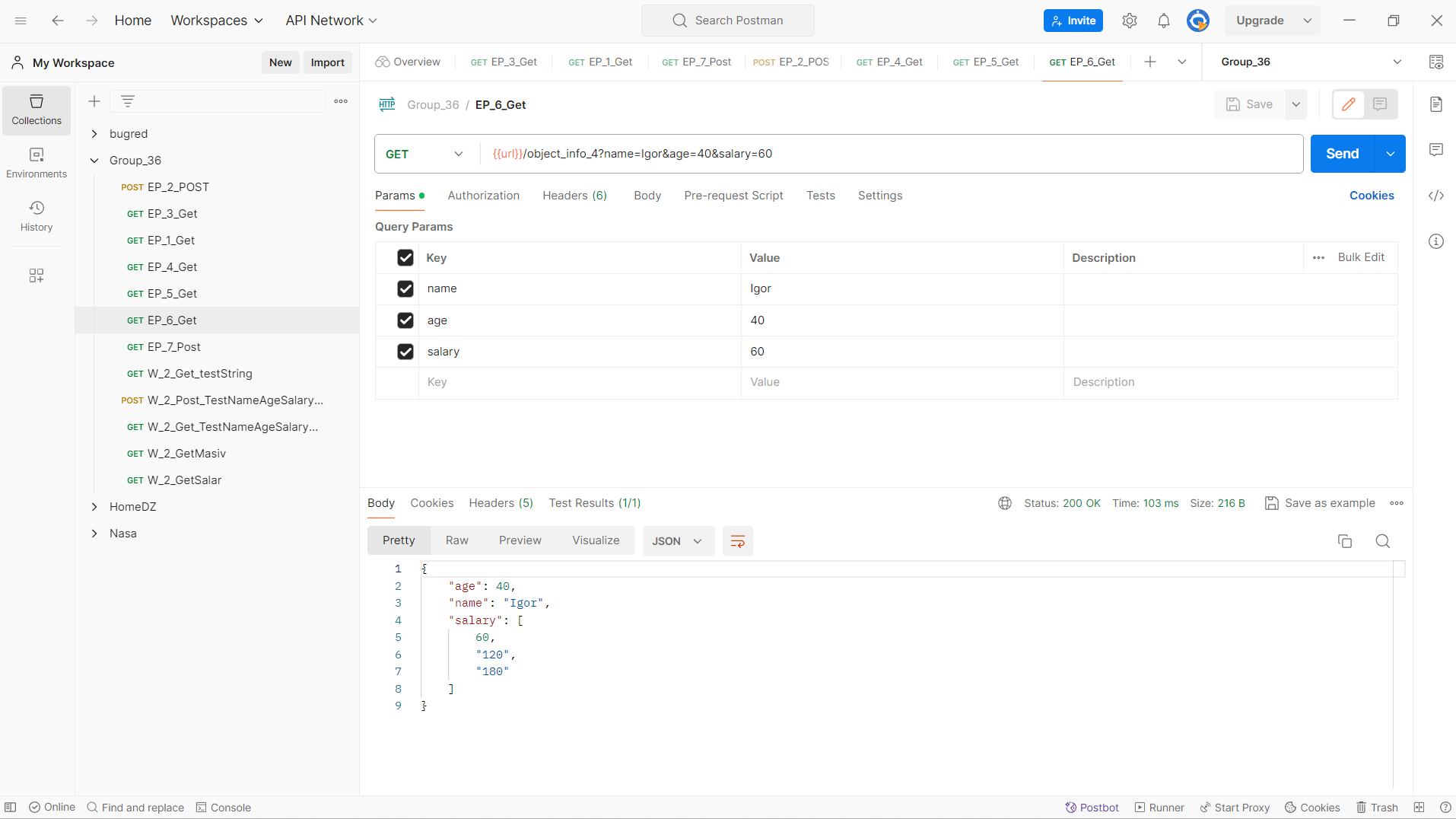
 salary: int

response:

{'name': name,

          'age': int(age),

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



EP\_7

Method: POST

EndPoint: /user\_info\_2

request form data:

 name: str

 age: int

 salary: int

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,

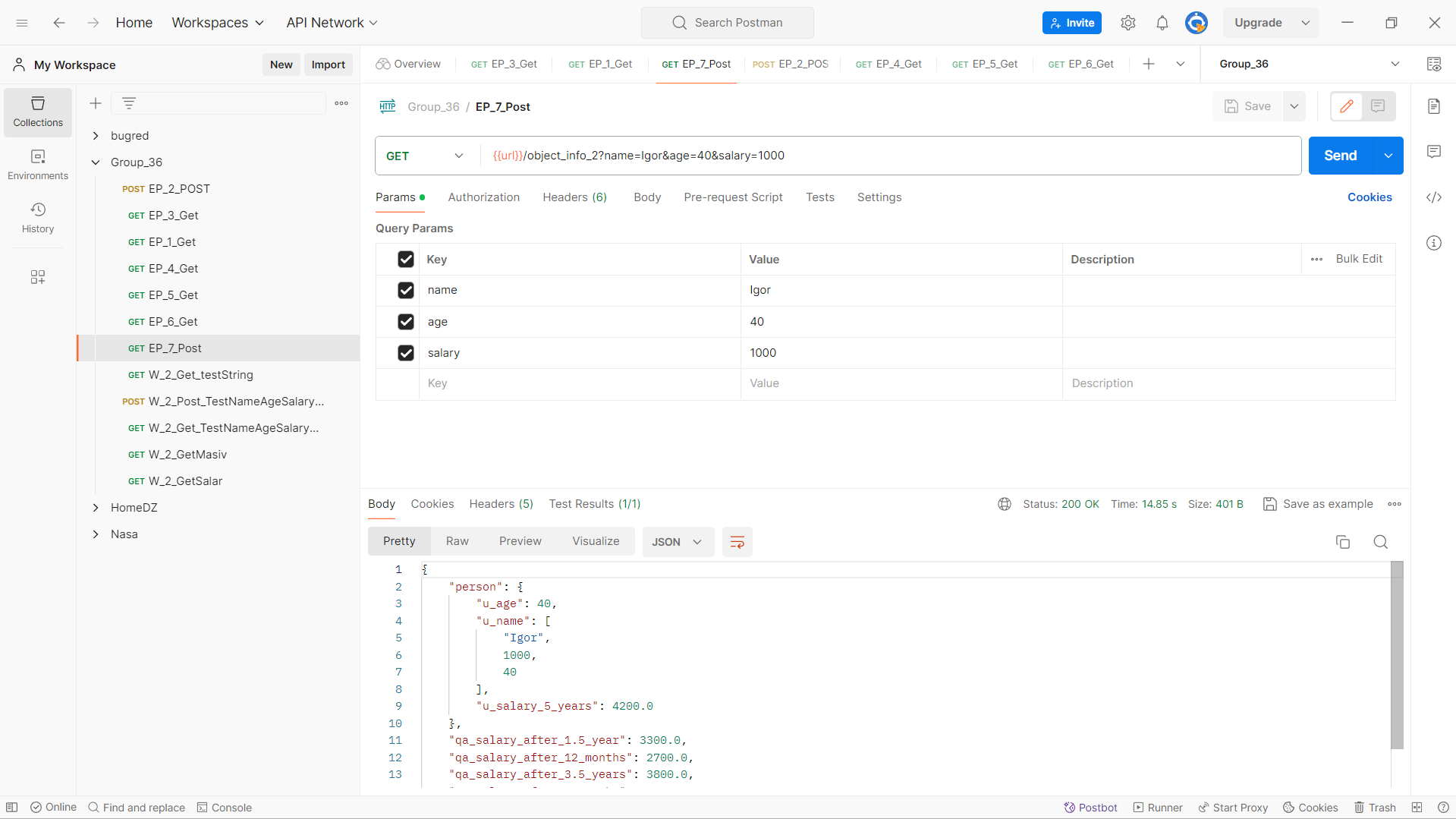
          'qa\_salary\_after\_3.5\_years': salary \* 3.8,

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

                     'u\_age': age,

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

          }



HW\_2 Postman

http://162.55.220.72:5005/first

1. Отправить запрос.

2. Статус код 200

3. Проверить, что в body приходит правильный string.

http://162.55.220.72:5005/user\_info\_3

1. Отправить запрос.

2. Статус код 200

3. Спарсить response body в json.

4. Проверить, что name в ответе равно name s request (name вбить руками.)

5. Проверить, что age в ответе равно age s request (age вбить руками.)

6. Проверить, что salary в ответе равно salary s request (salary вбить руками.)

7. Спарсить request.

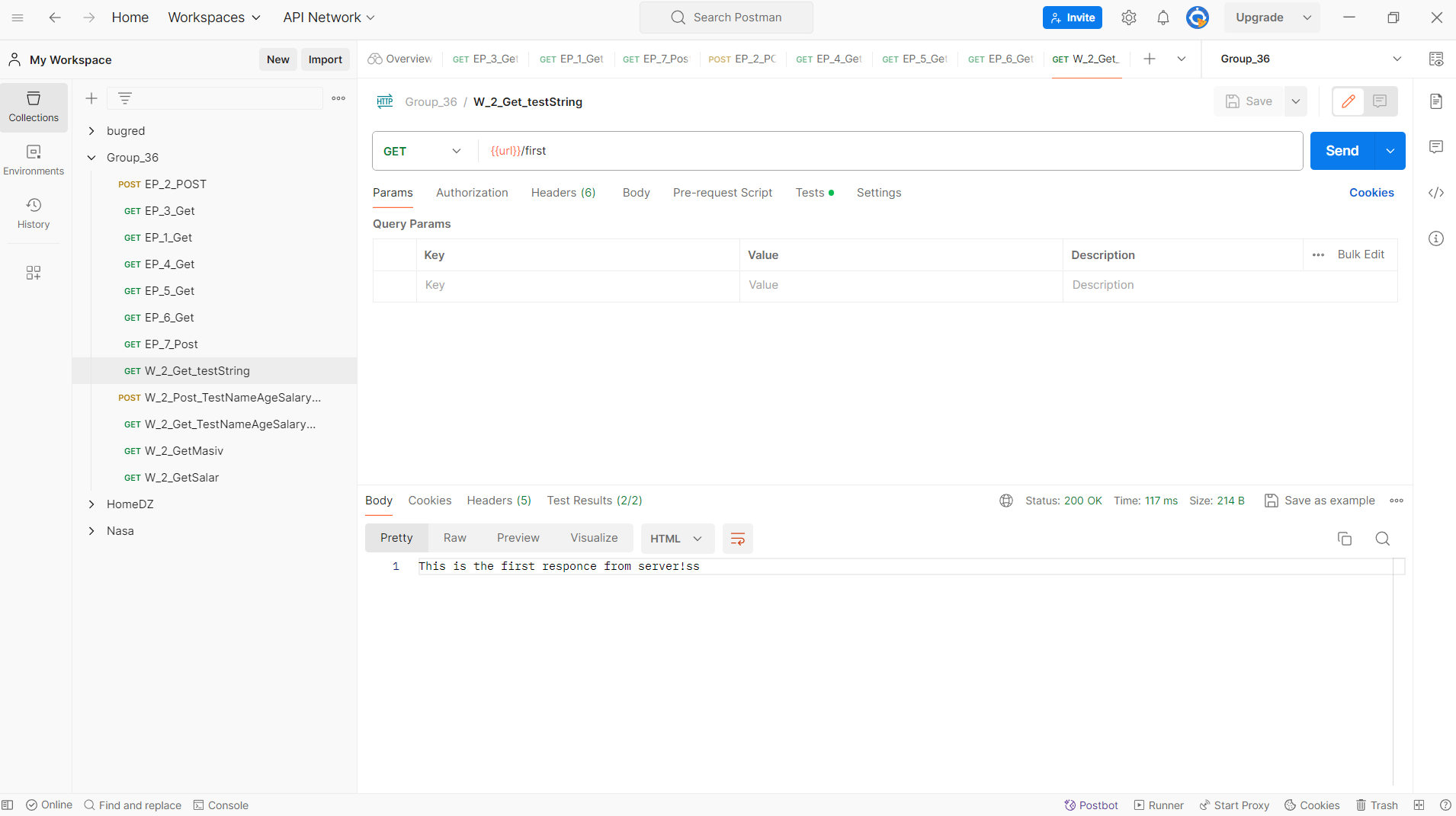
8. Проверить, что name в ответе равно name s request (name забрать из request.)

9. Проверить, что age в ответе равно age s request (age забрать из request.)

10. Проверить, что salary в ответе равно salary s request (salary забрать из request.)

11. Вывести в консоль параметр family из response.

12. Проверить что u\_salary\_1\_5\_year в ответе равно salary\*4 (salary забрать из request)





http://162.55.220.72:5005/user\_info\_3

1. Отправить запрос.

2. Статус код 200

3. Спарсить response body в json.

4. Проверить, что name в ответе равно name s request (name вбить руками.)

5. Проверить, что age в ответе равно age s request (age вбить руками.)

6. Проверить, что salary в ответе равно salary s request (salary вбить руками.)

7. Спарсить request.

8. Проверить, что name в ответе равно name s request (name забрать из request.)

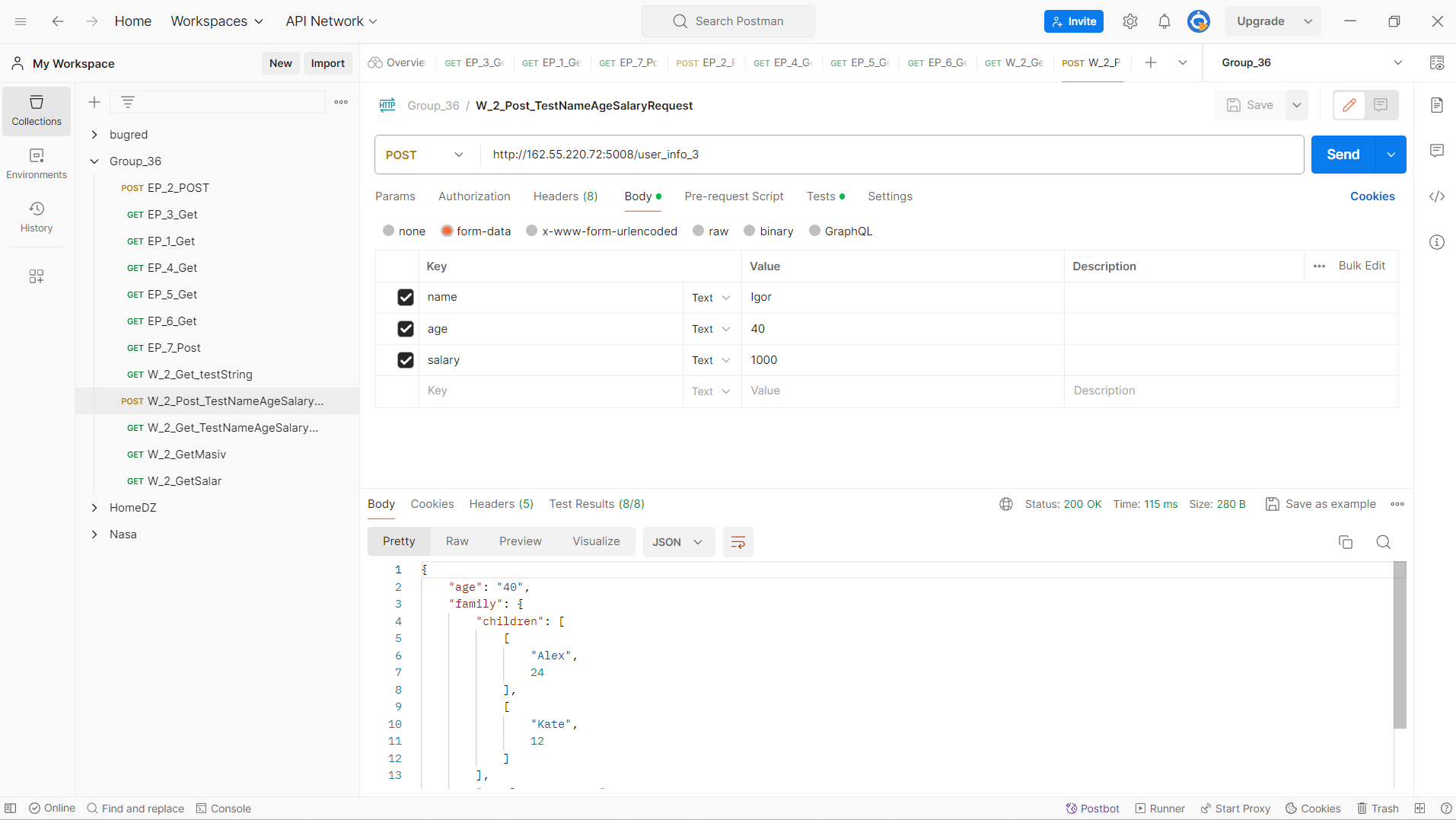
9. Проверить, что age в ответе равно age s request (age забрать из request.)

10. Проверить, что salary в ответе равно salary s request (salary забрать из request.)

11. Вывести в консоль параметр family из response.

12. Проверить что u\_salary\_1\_5\_year в ответе равно salary\*4 (salary забрать из request)





http://162.55.220.72:5005/object\_info\_3

1. Отправить запрос.

2. Статус код 200

3. Спарсить response body в json.

4. Спарсить request.

5. Проверить, что name в ответе равно name s request (name забрать из request.)

6. Проверить, что age в ответе равно age s request (age забрать из request.)

7. Проверить, что salary в ответе равно salary s request (salary забрать из request.)

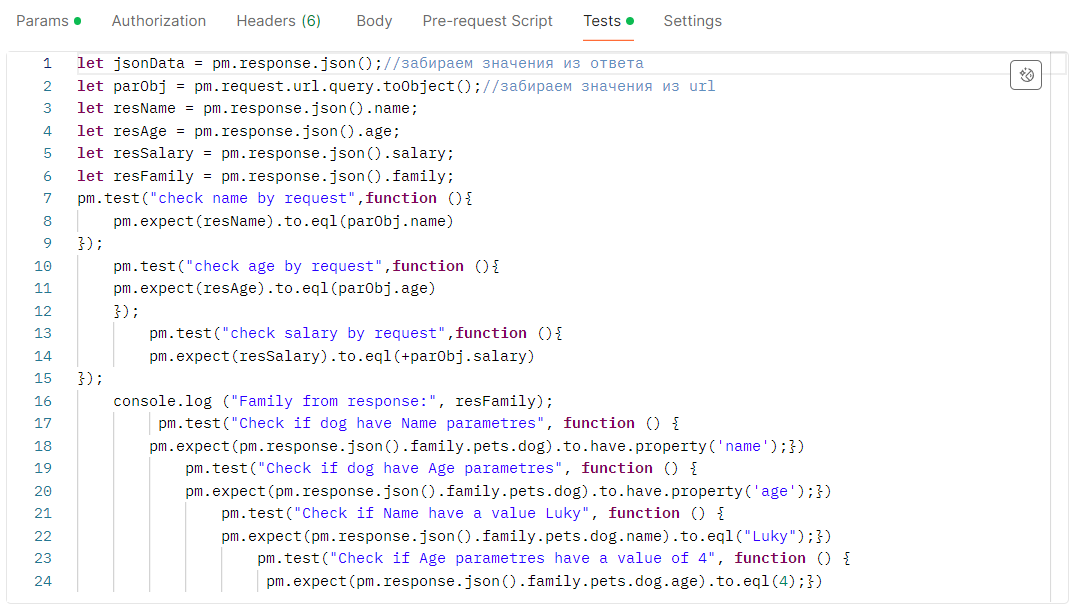
8. Вывести в консоль параметр family из response.

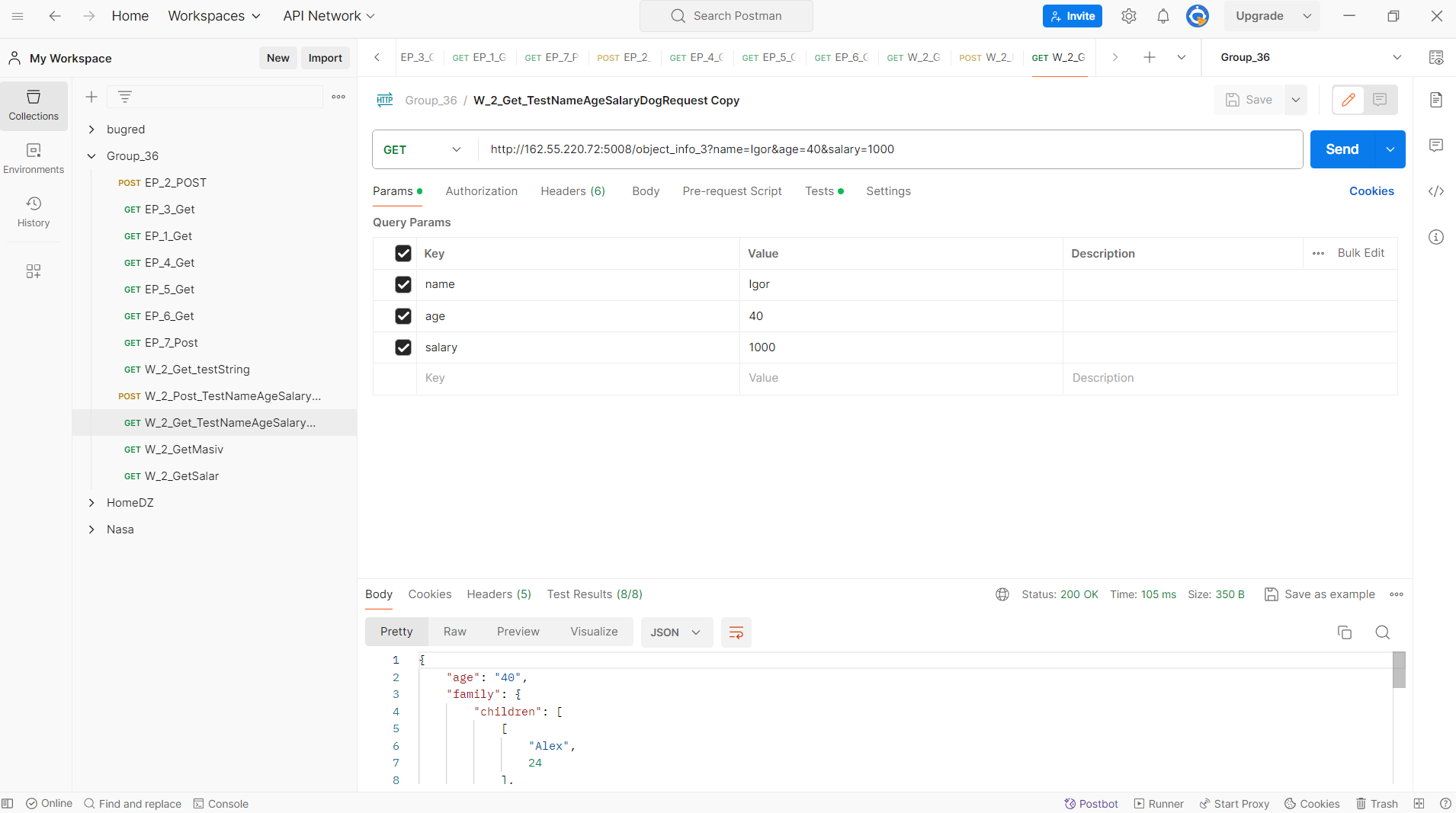
9. Проверить, что у параметра dog есть параметры name.

10. Проверить, что у параметра dog есть параметры age.

11. Проверить, что параметр name имеет значение Luky.

12. Проверить, что параметр age имеет значение 4.





http://162.55.220.72:5005/object\_info\_4

1. Отправить запрос.

2. Статус код 200

3. Спарсить response body в json.

4. Спарсить request.

5. Проверить, что name в ответе равно name s request (name забрать из request.)

6. Проверить, что age в ответе равно age из request (age забрать из request.)

7. Вывести в консоль параметр salary из request.

8. Вывести в консоль параметр salary из response.

9. Вывести в консоль 0-й элемент параметра salary из response.

10. Вывести в консоль 1-й элемент параметра salary параметр salary из response.

11. Вывести в консоль 2-й элемент параметра salary параметр salary из response.

12. Проверить, что 0-й элемент параметра salary равен salary из request (salary забрать из request.)

13. Проверить, что 1-й элемент параметра salary равен salary\*2 из request (salary забрать из request.)

14. Проверить, что 2-й элемент параметра salary равен salary\*3 из request (salary забрать из request.)

15. Создать в окружении переменную name

16. Создать в окружении переменную age

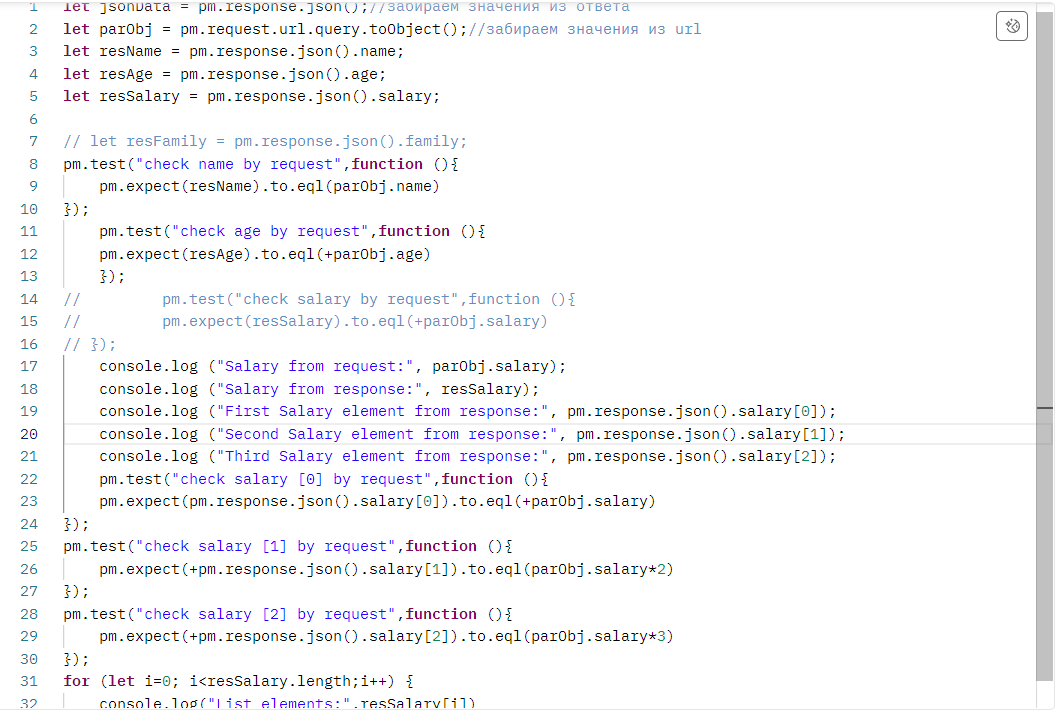
17. Создать в окружении переменную salary

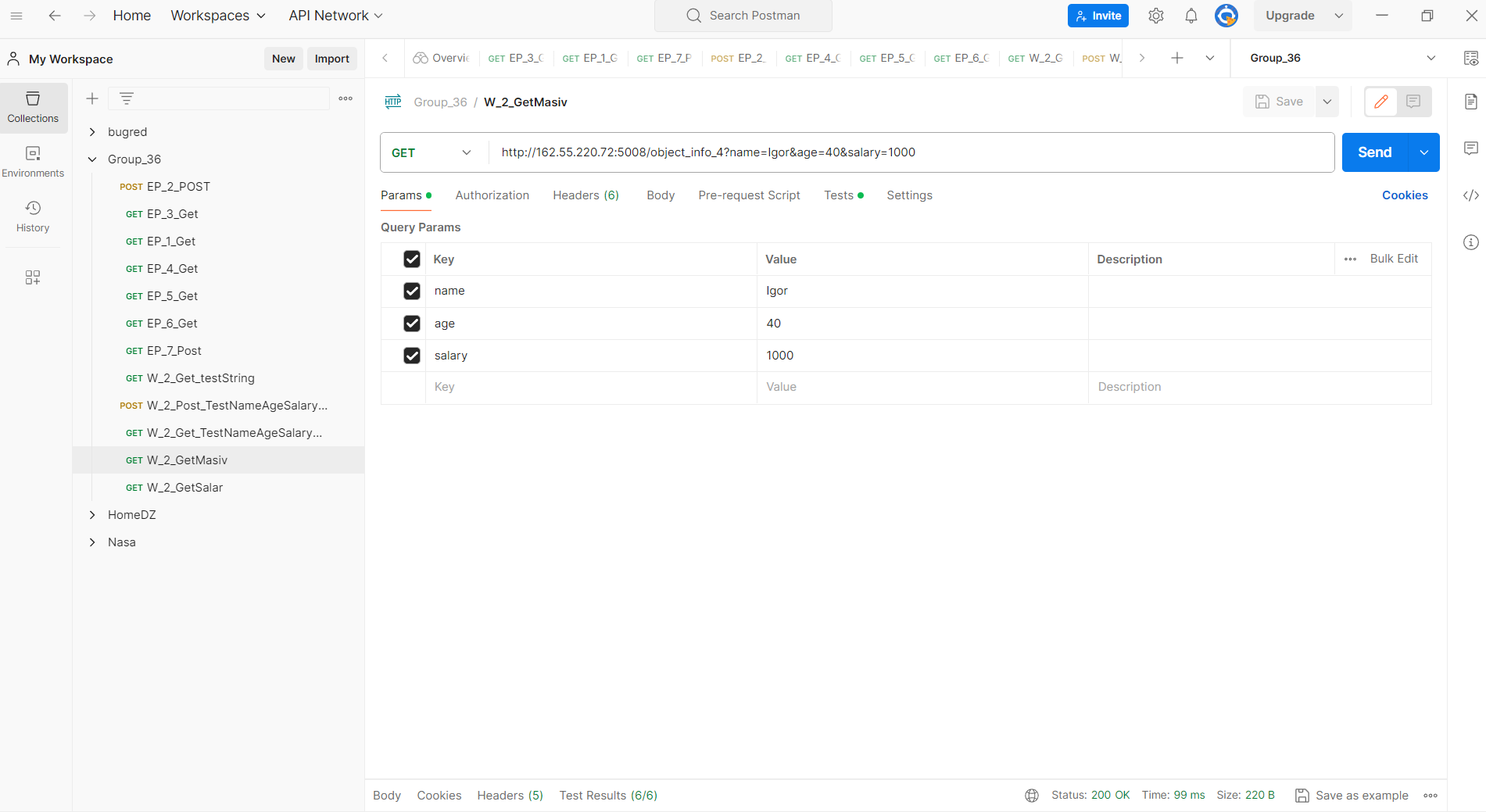
18. Передать в окружение переменную name

19. Передать в окружение переменную age

20. Передать в окружение переменную salary

21. Написать цикл который выведет в консоль по порядку элементы списка из параметра salary





http://162.55.220.72:5005/user\_info\_2

1. Вставить параметр salary из окружения в request

2. Вставить параметр age из окружения в age

3. Вставить параметр name из окружения в name

4. Отправить запрос.

5. Статус код 200

6. Спарсить response body в json.

7. Спарсить request.

8. Проверить, что json response имеет параметр start\_qa\_salary

9. Проверить, что json response имеет параметр qa\_salary\_after\_6\_months

10. Проверить, что json response имеет параметр qa\_salary\_after\_12\_months

11. Проверить, что json response имеет параметр qa\_salary\_after\_1.5\_year

12. Проверить, что json response имеет параметр qa\_salary\_after\_3.5\_years

13. Проверить, что json response имеет параметр person

14. Проверить, что параметр start\_qa\_salary равен salary из request (salary забрать из request.)

15. Проверить, что параметр qa\_salary\_after\_6\_months равен salary\*2 из request (salary забрать из request.)

16. Проверить, что параметр qa\_salary\_after\_12\_months равен salary\*2.7 из request (salary забрать из request.)

17. Проверить, что параметр qa\_salary\_after\_1.5\_year равен salary\*3.3 из request (salary забрать из request.)

18. Проверить, что параметр qa\_salary\_after\_3.5\_years равен salary\*3.8 из request (salary забрать из request.)

19. Проверить, что в параметре person, 1-й элемент из u\_name равен salary из request (salary забрать из request.)

20. Проверить, что что параметр u\_age равен age из request (age забрать из request.)

21. Проверить, что параметр u\_salary\_5\_years равен salary\*4.2 из request (salary забрать из request.)

22. \*\*\*Написать цикл который выведет в консоль по порядку элементы списка из параметра person.



