**Técnica: Pomelo**

**Task #1**

API: <https://age-of-empires-2-api.herokuapp.com/docs/#/>

SUT: Age of Empires

* Mostrar Print Army type de los Britons.

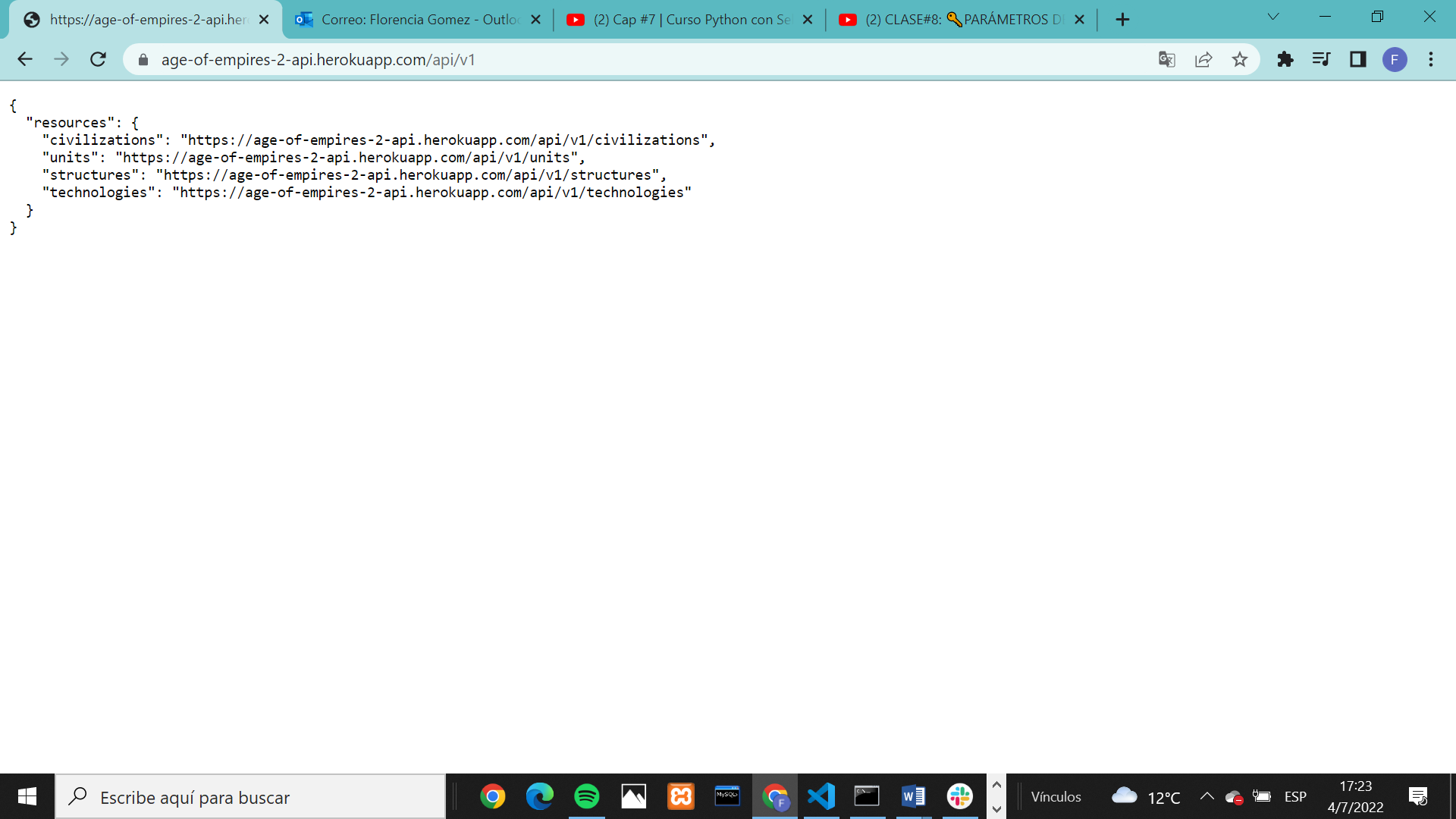
Para test API utilizo Postman.

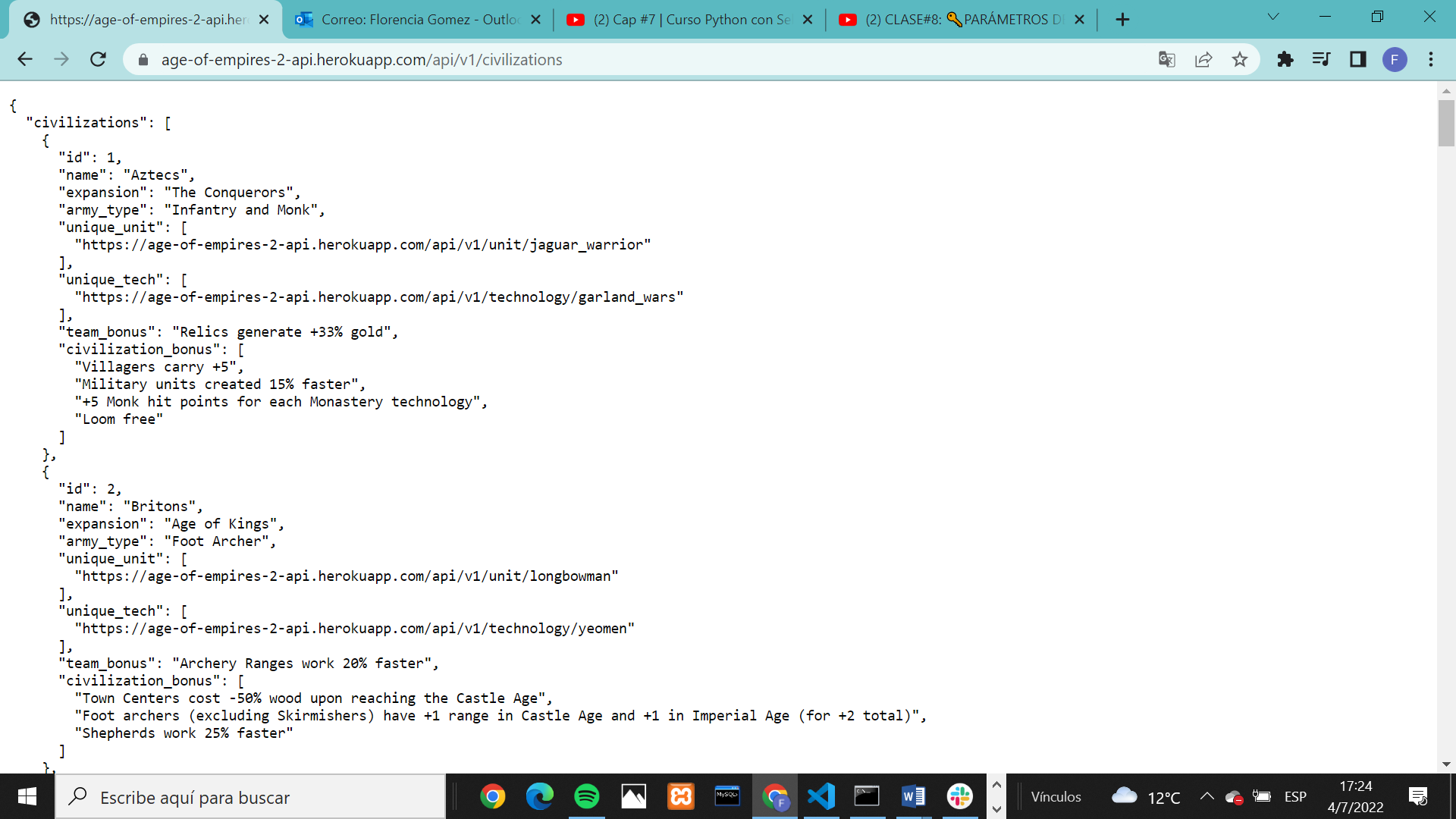
**Aclaraciones:**

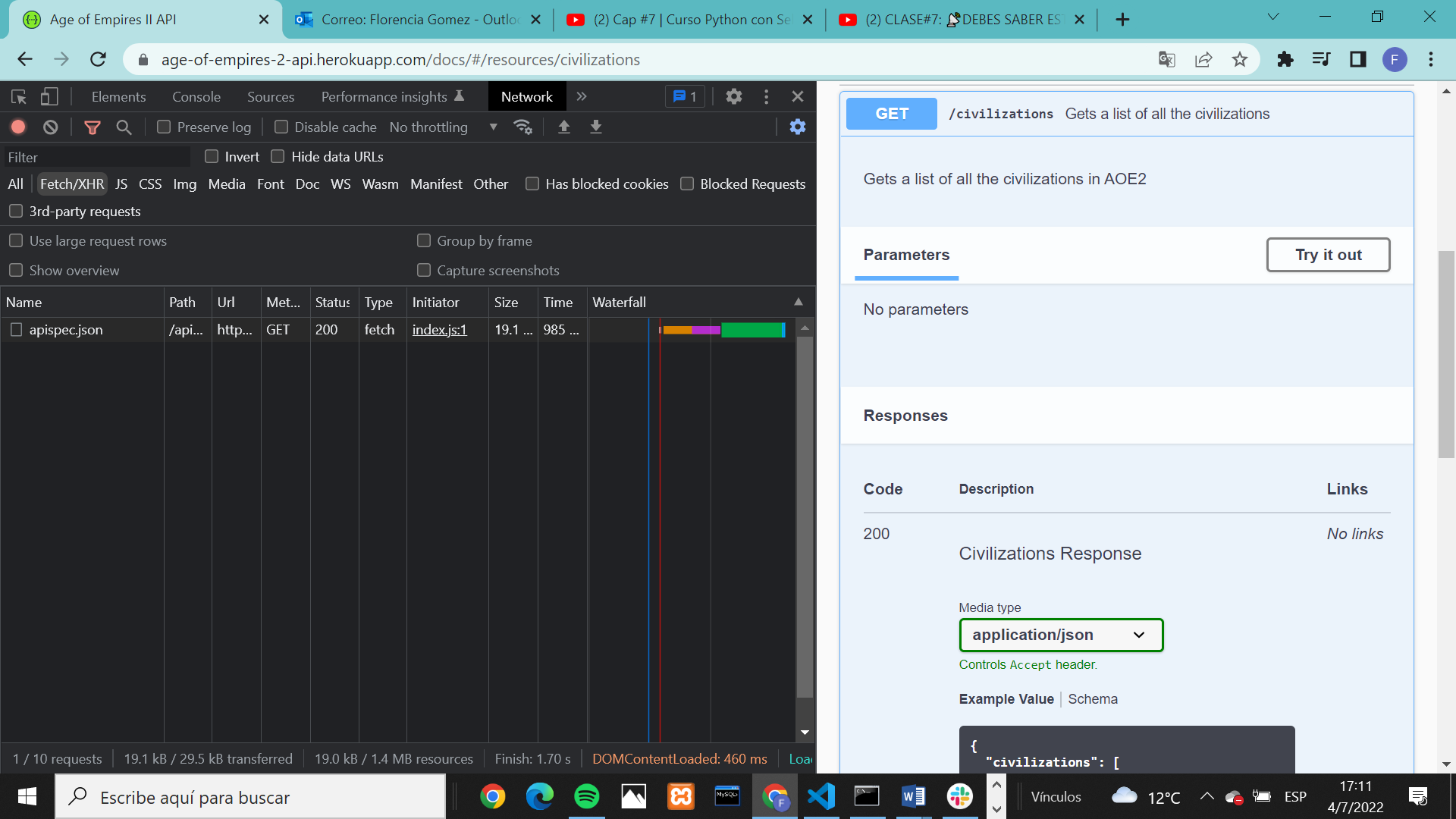
La Api es púbica por tanto no es necesario configurar authentication and authorization (key y token) en el request., surge de la documentación.

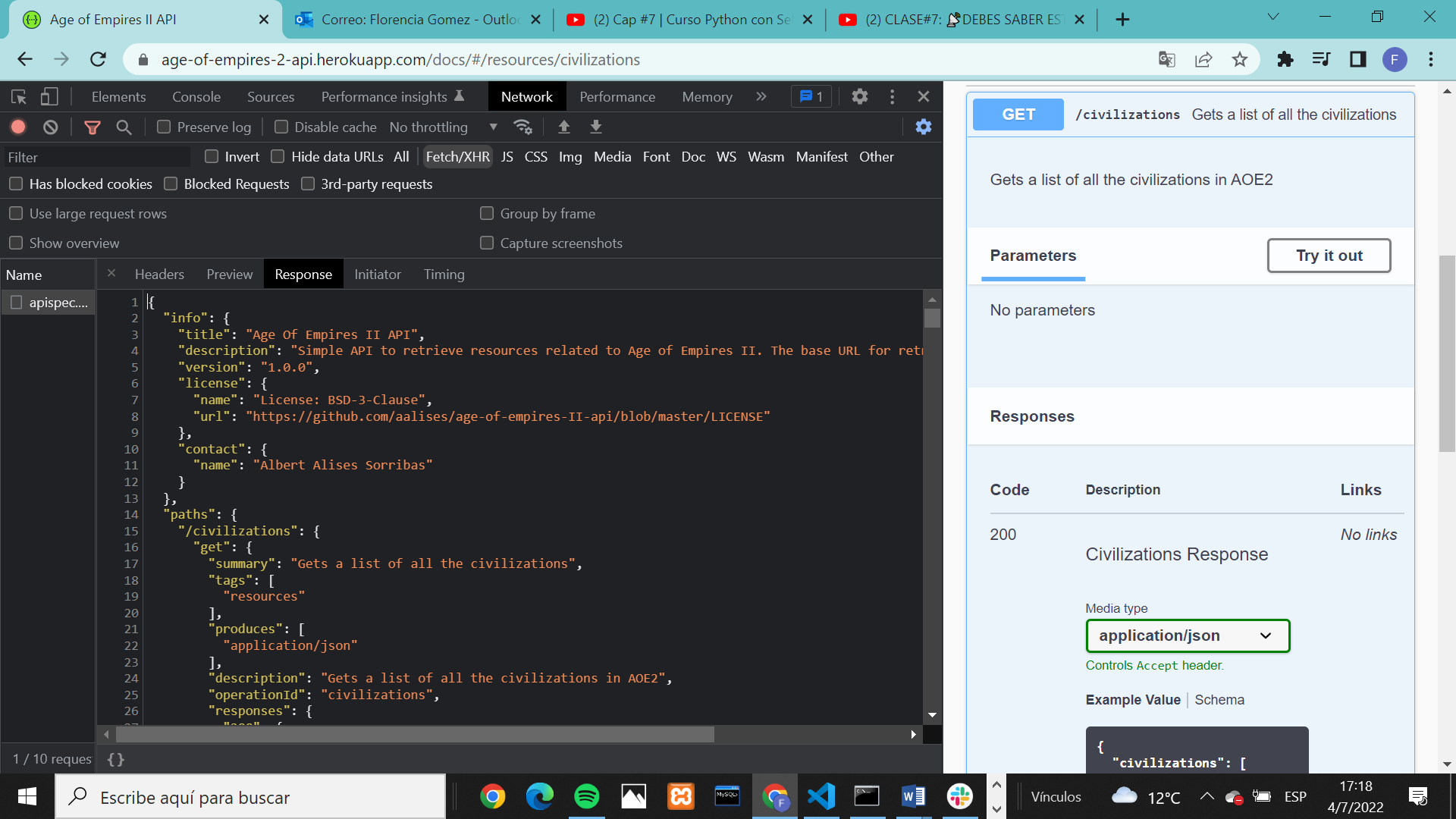
El End point: api/v1 y civilizations.

El meth que voy a utilizar es GET.

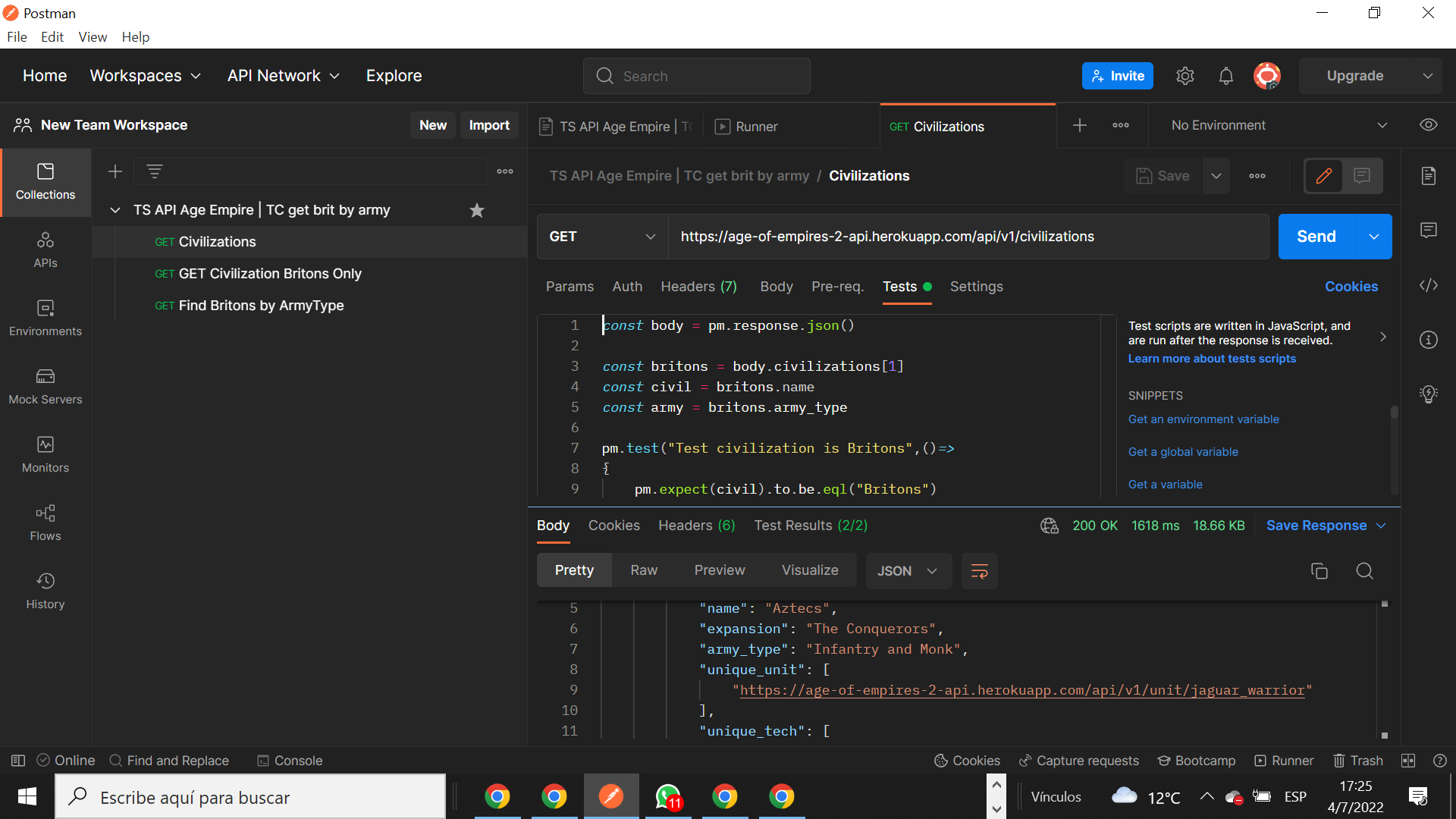








Tests: encuentro 3 formas llegar a los prints.



1. **Desde Civilizations.**

*const* body = pm.response.json()

*const* britons = body.civilizations[1]

*const* civil = britons.name

*const* army = britons.army\_type

pm.test("Test civilization is Britons",()*=>*

{

    pm.expect(civil).to.be.eql("Britons")

})

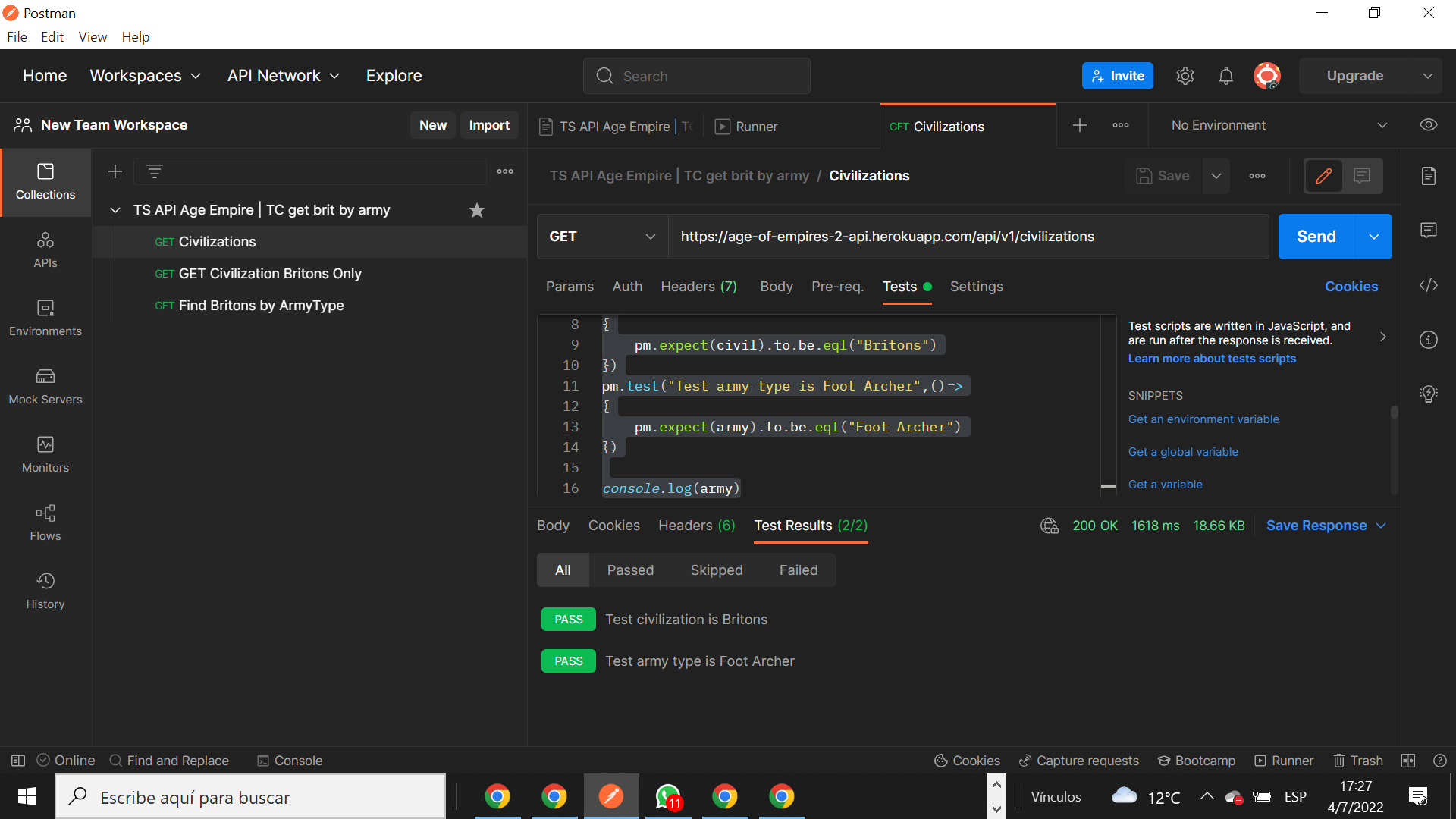
pm.test("Test army type is Foot Archer",()*=>*

{

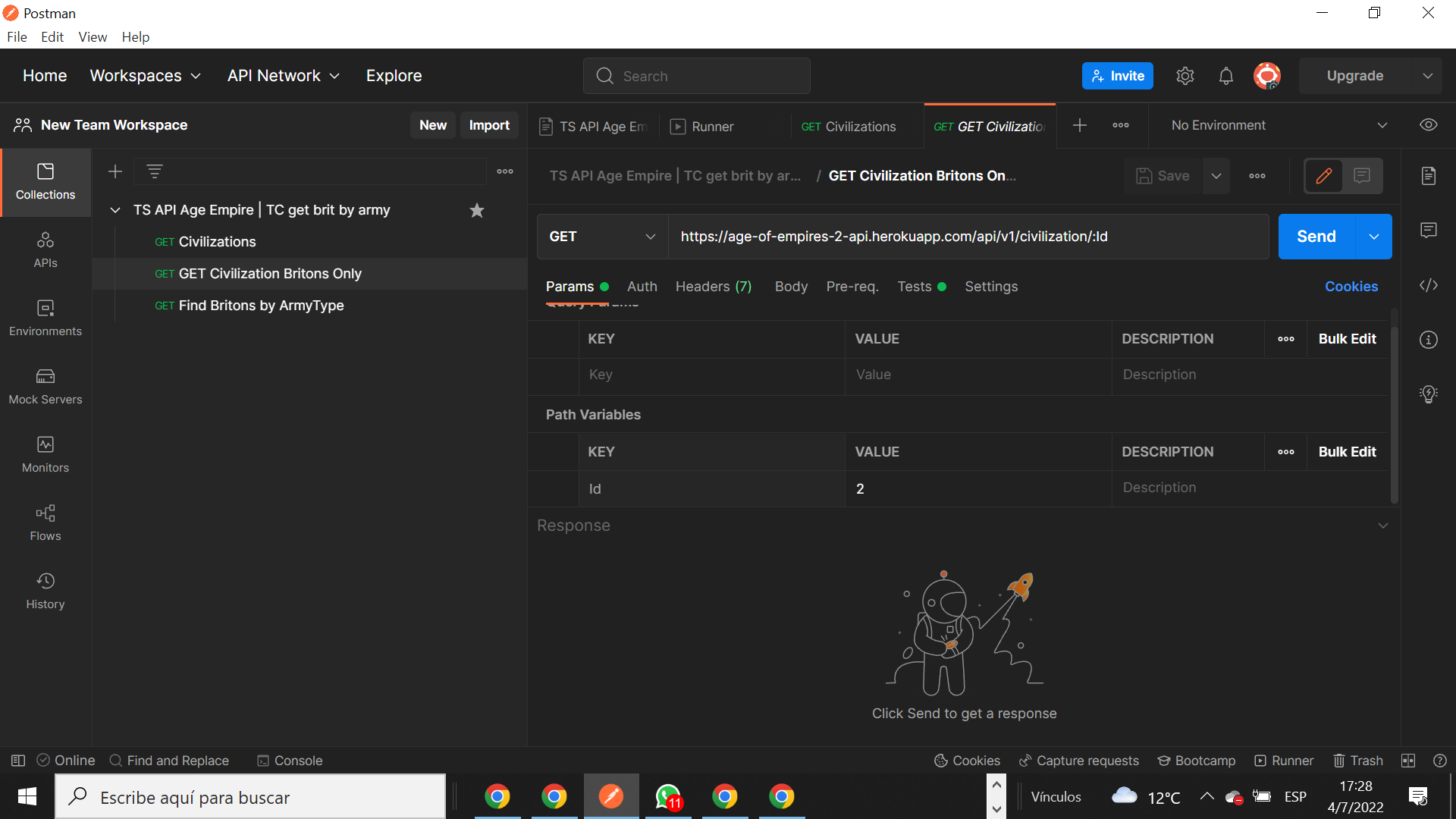
    pm.expect(army).to.be.eql("Foot Archer")

})

*console*.log(army)



1. **Utilizando PATH VARIABLES, sabiendo que el ID=2.**



*const* body = pm.response.json()

*const* civil = body.name

*const* army = body.army\_type

pm.test("Test civilization is Britons",()*=>*

{

    pm.expect(civil).to.be.eql("Britons")

})

pm.test("Test army type is Foot Archer",()*=>*

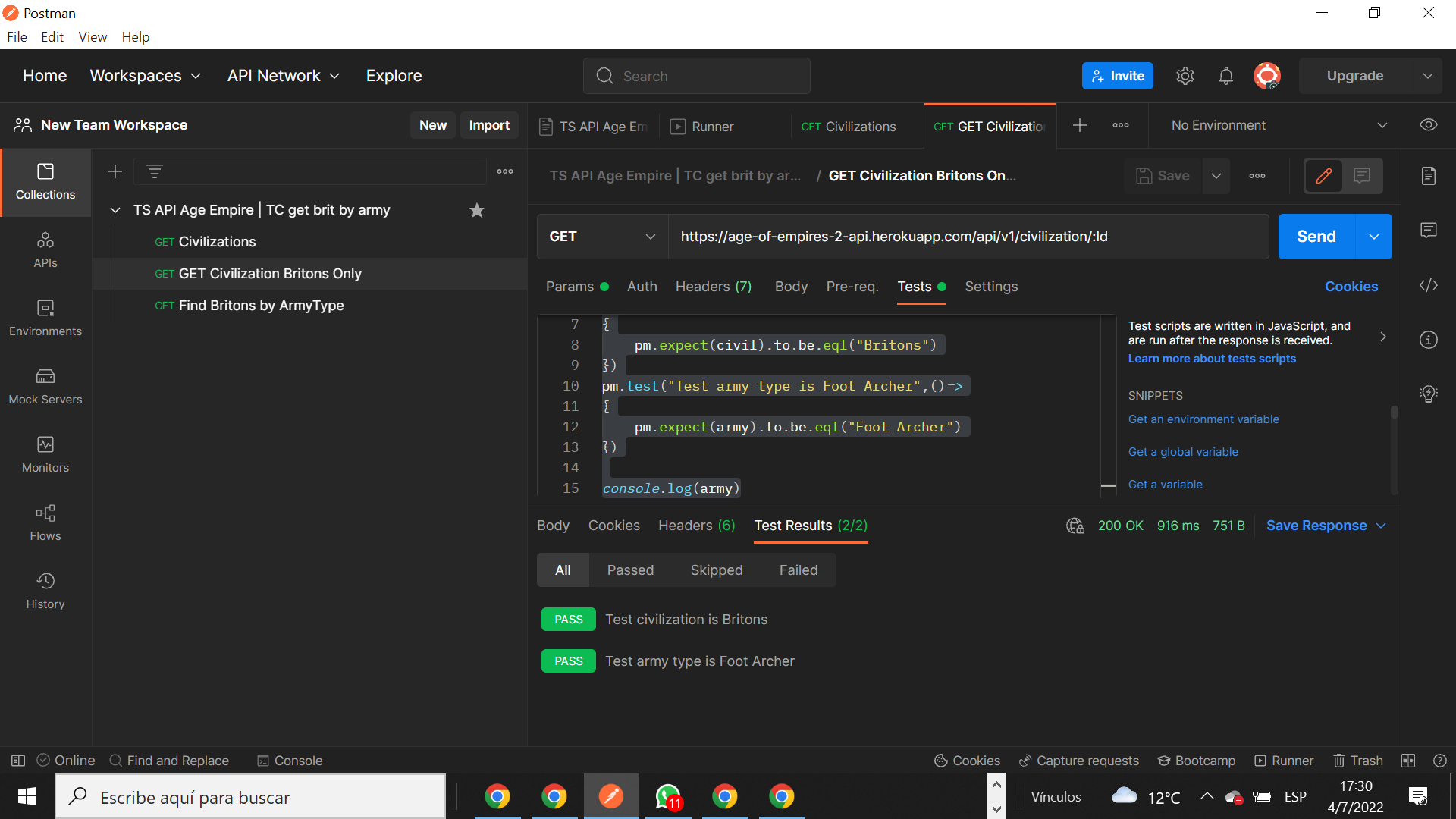
{

    pm.expect(army).to.be.eql("Foot Archer")

})

*console*.log(army)

1. **Buscando Britons pero desde el Armytype (“foot archer”)**
2. *const* body = pm.response.json()
3. *const* civils = body.civilizations
4. *const* britons = civils.find(*i* *=>* i.army\_type === "Foot Archer")
5. *const* name = britons.name
6. *const* army = britons.army\_type
7. pm.test("Test civilization is Britons",()*=>*
8. {
9. pm.expect(name).to.be.eql("Britons")
10. })
11. pm.test("Test army type is Foot Archer",()*=>*
12. {
13. pm.expect(army).to.be.eql("Foot Archer")
14. })
15. *console*.log(army)
16. *const* body = pm.response.json()
17. *const* civils = body.civilizations
18. *const* britons = civils.find(*i* *=>* i.army\_type === "Foot Archer")
19. *const* name = britons.name
20. *const* army = britons.army\_type
21. pm.test("Test civilization is Britons",()*=>*
22. {
23. pm.expect(name).to.be.eql("Britons")
24. })
25. pm.test("Test army type is Foot Archer",()*=>*
26. {
27. pm.expect(army).to.be.eql("Foot Archer")
28. })
29. *console*.log(army)



**Task #2**

1. Abrir google buscar pomelo y hacer click en pomelo.

Para test búsqueda en google utilice Selenium con Python. (en este punto es dable destacar que el test correrá mejor en Cypress utilizando lengh. Cucumber) no lo utilizo porque aún me encuentro aprendiéndolo, pero sé que el mismo nos permitiría tener el test automatizado y correrlo en distintos drivers.

#enviroment

from selenium import webdriver

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

#paraeldriveChorme

driver = webdriver.Chrome (executable\_path=r"C:\dchrome\chromedriver.exe")

driver.get ("https://www.google.com/")

driver.maximize\_window()

#searchpomeloenboxsearch

search= driver.find\_element(by=By.NAME, value="q")

search.send\_keys("pomelo")

time.sleep(1)

options = driver.find\_elements (By.XPATH, "//li[@class='sbct']//b")

for i in range(len(options)):

    element = options[0]

    text = element.text

    if text == "fintech":

        assert text == "fintech"

        element.click()

        time.sleep(1)

pomelo = driver.find\_elements(By.XPATH, "//div[@id='search']//h3")[0].click()

time.sleep(1)

assert "pomelo" in driver.title

driver.quit()

***Pasos:***

**Los pasos están aclarados en comentarios.**

**La mayor dificultad que encontré en este tipo de automatización es que la search box de google contiene un array, el que es necesario llamar desde el xpath y luego iterarlo para que, en la misma, utilizando esta automatización siempre lo encuentre. Más aun teniendo en cuenta que su orden cambia.**

