

## 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:

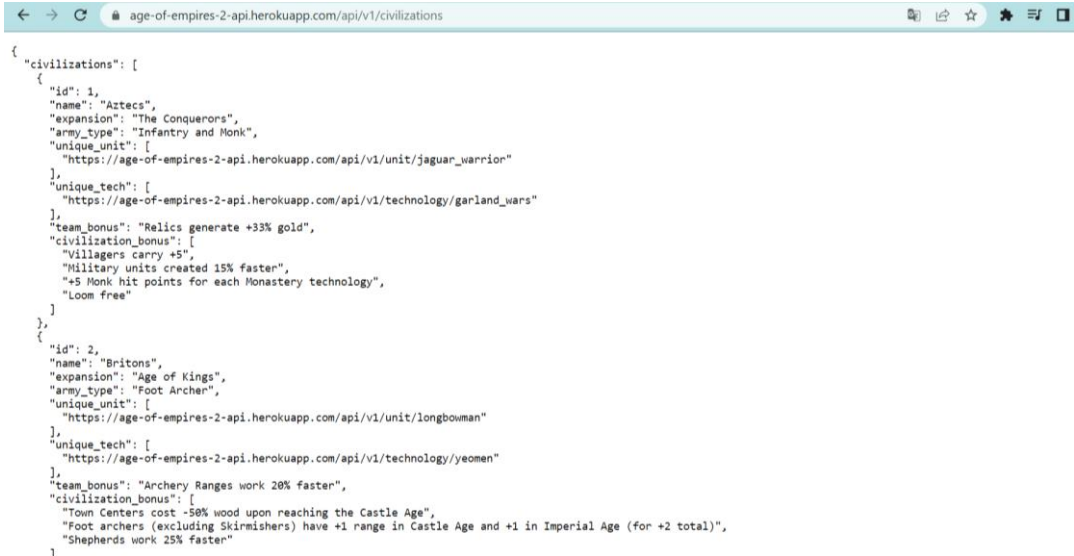
La Api es pública por tanto no es necesario configurar authentication and authorization (key y token) en el request., surge de la documentación.

El End point: api/v1 y civilizations.

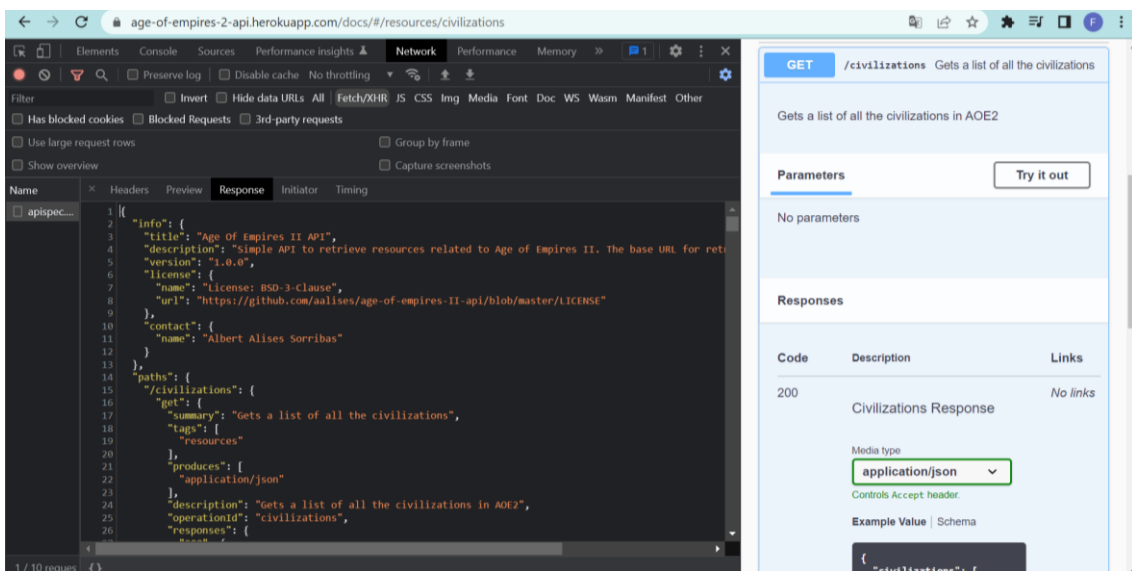
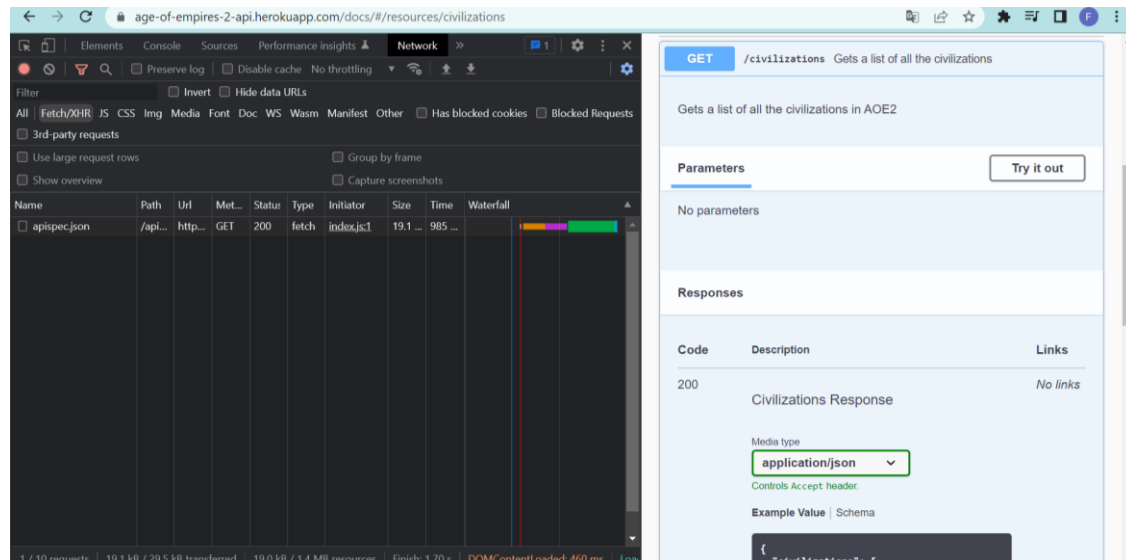
El meth que voy a utilizar es GET.



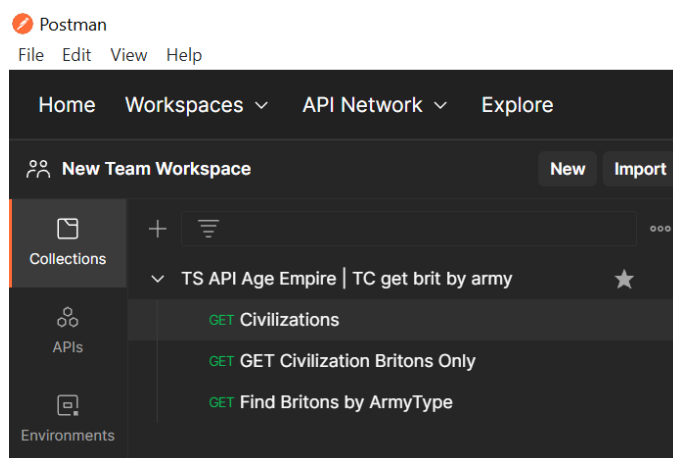
```
{
  "resources": {
    "civilizations": "https://age-of-empires-2-api.herokuapp.com/api/v1/civilizations",
    "units": "https://age-of-empires-2-api.herokuapp.com/api/v1/units",
    "structures": "https://age-of-empires-2-api.herokuapp.com/api/v1/structures",
    "technologies": "https://age-of-empires-2-api.herokuapp.com/api/v1/technologies"
  }
}
```



```
{
  "civilizations": [
    {
      "id": 1,
      "name": "Aztecs",
      "expansion": "The Conquerors",
      "army_type": "Infantry and Monk",
      "unique_unit": [
        "https://age-of-empires-2-api.herokuapp.com/api/v1/unit/jaguar_warrior"
      ],
      "unique_tech": [
        "https://age-of-empires-2-api.herokuapp.com/api/v1/technology/garland_wars"
      ],
      "team_bonus": "Relics generate +33% gold",
      "civilization_bonus": [
        "Villagers carry +5",
        "Military units created 15% faster",
        "+5 Monk hit points for each Monastery technology",
        "Loom free"
      ]
    },
    {
      "id": 2,
      "name": "Britons",
      "expansion": "Age of Kings",
      "army_type": "Foot Archer",
      "unique_unit": [
        "https://age-of-empires-2-api.herokuapp.com/api/v1/unit/longbowman"
      ],
      "unique_tech": [
        "https://age-of-empires-2-api.herokuapp.com/api/v1/technology/yeomen"
      ],
      "team_bonus": "Archery Ranges work 20% faster",
      "civilization_bonus": [
        "Town Centers cost -50% wood upon reaching the Castle Age",
        "Foot archers (excluding Skirmishers) have +1 range in Castle Age and +1 in Imperial Age (for +2 total)",
        "Shepherds work 25% faster"
      ]
    }
  ]
}
```



Tests: encuentro 3 formas llegar a los prints.



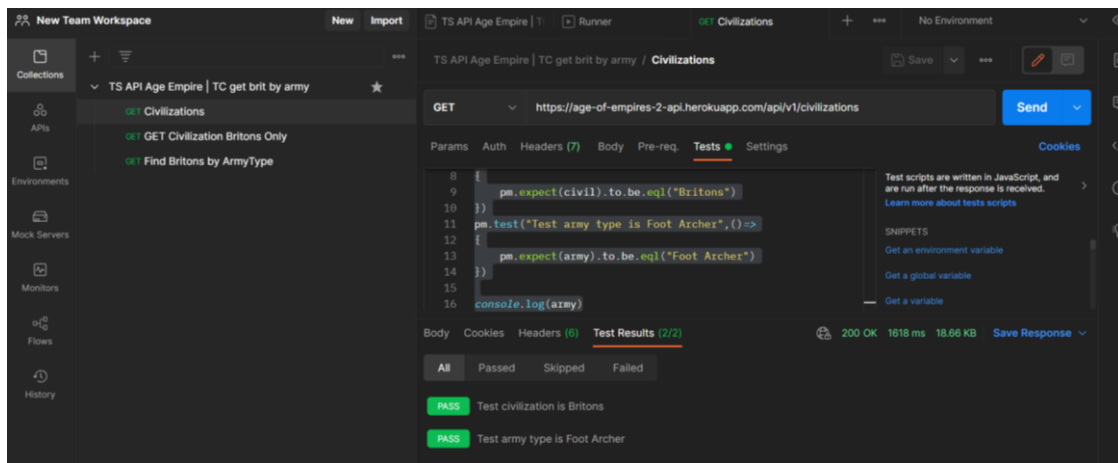
## 1. Desde Civilizations.

```
const body = pm.response.json()

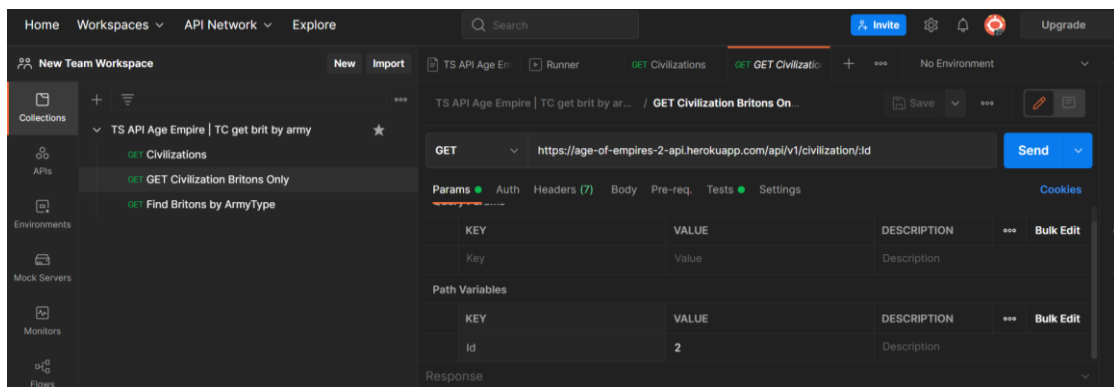
const britons = body.civilizations[1]
const civil = britons.name
const army = britons.army_type

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)
```



## 2. 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)

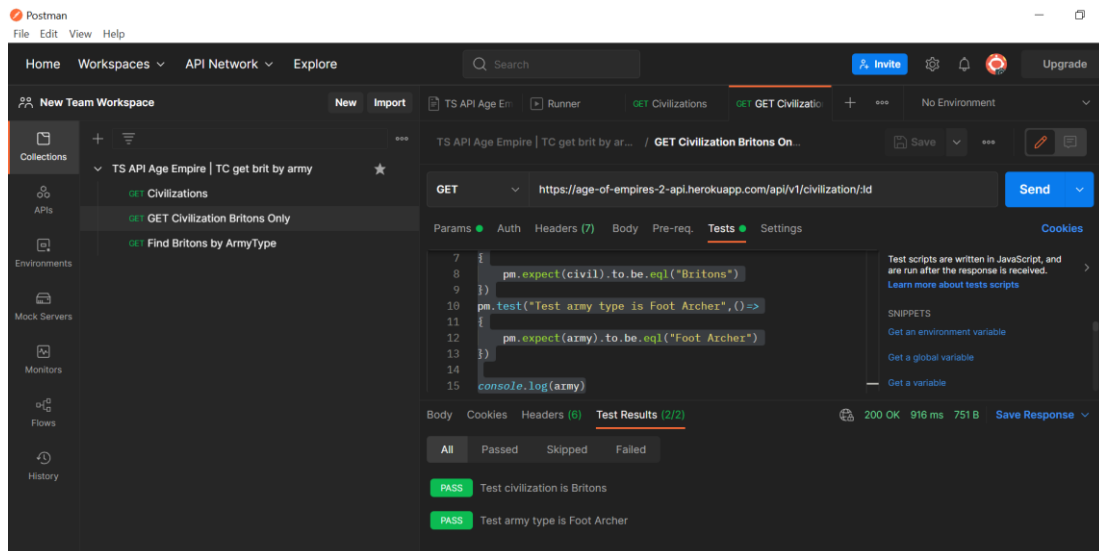
```

### 3. Buscando Britons pero desde el Armytype ("foot archer")

```

4. const body = pm.response.json()
5.
6. const civils = body.civilizations
7. const britons = civils.find(i => i.army_type === "Foot Archer")
8. const name = britons.name
9. const army = britons.army_type
10.
11. pm.test("Test civilization is Britons", () =>
12. {
13.     pm.expect(name).to.be.eql("Britons")
14. })
15. pm.test("Test army type is Foot Archer", () =>
16. {
17.     pm.expect(army).to.be.eql("Foot Archer")
18. })
19.
20. console.log(army)
21. const body = pm.response.json()
22.
23. const civils = body.civilizations
24. const britons = civils.find(i => i.army_type === "Foot Archer")
25. const name = britons.name
26. const army = britons.army_type
27.
28. pm.test("Test civilization is Britons", () =>
29. {
30.     pm.expect(name).to.be.eql("Britons")
31. })
32. pm.test("Test army type is Foot Archer", () =>
33. {
34.     pm.expect(army).to.be.eql("Foot Archer")
35. })
36.
37. 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 length. 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.

