



Laboratorio No. 5

Programación de Plataformas Móviles Sección 10

José Rodrigo Marchena Molina - 22398

Repositorio:

https://github.com/MarchMol/lab5_PM

Video demomstrativo:

<https://drive.google.com/file/d/1sqw8UvEZbzSZe3RkYF6kEP43Mgx4Pjrk/view?usp=sharing>

API utilizada:

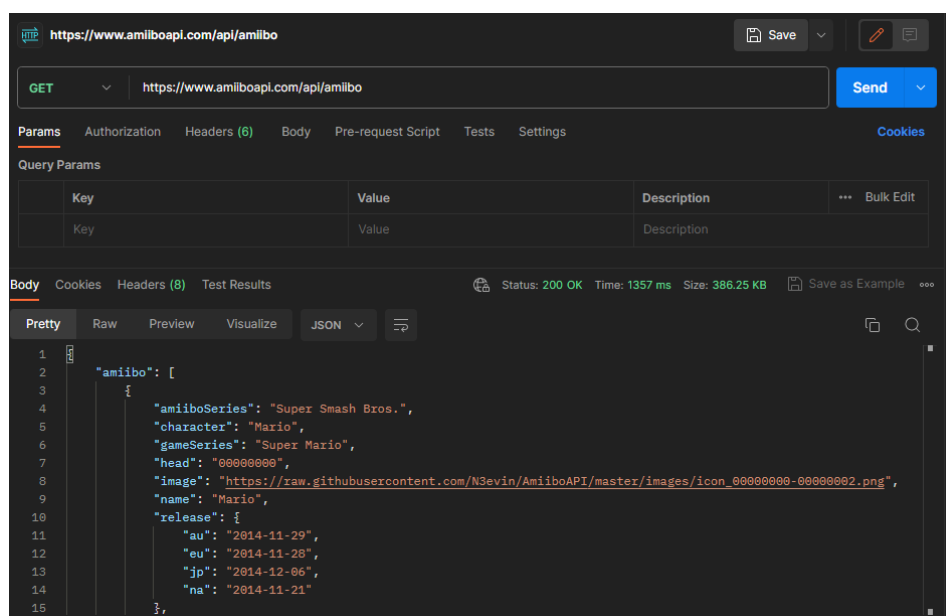
<https://amiiboapi.com/>

Investigación:

La API utilizada es una RESTful API que almacena información sobre las figuras Amiibo creadas por la compañía de videojuegos Nintendo. La interfaz está dedicada para propósitos educativos, de manera que no permite verbos que alteren la información presente, solo GET.

Entre otras características, esta API no requiere de ninguna autorización ni key, pues está abierta para el uso público y su única función es devolver información. En la documentación del API, también se provee información y ejemplos sobre los distintos tipos de llamadas como *amiibo/* y *gameseries/* Las cuales serán utilizadas para este laboratorio.

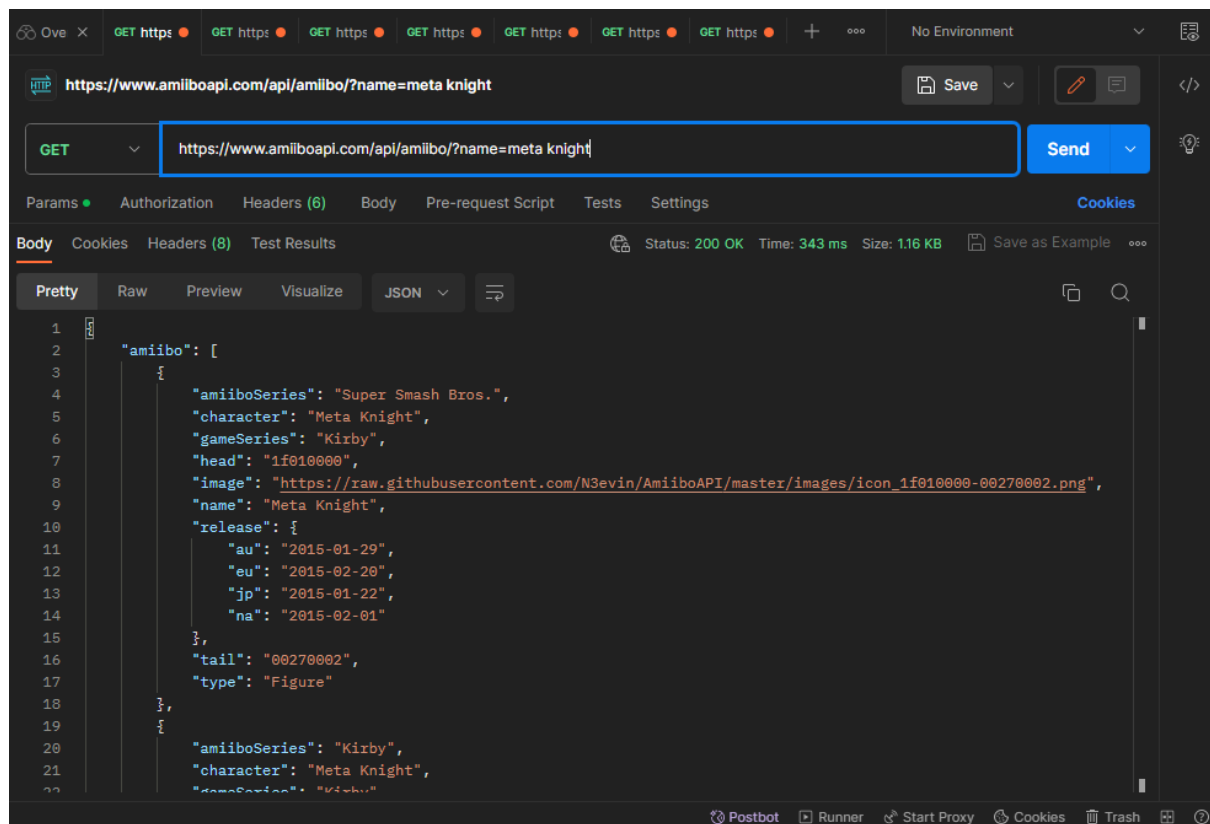
Se decidió investigar primero sobre la llamada *amiibo/* la cual regresa información de todos los amiibos presentes en la base de datos. A continuación se puede observar que pasa si hacemos la llamada sin parámetros.



```
1 GET https://www.amiiboapi.com/api/amiibo
2
3 {"amiibo": [
4   {
5     "amiiboSeries": "Super Smash Bros.",
6     "character": "Mario",
7     "gameSeries": "Super Mario",
8     "head": "00000000",
9     "image": "https://raw.githubusercontent.com/N3evin/AmiiboAPI/master/images/icon_00000000-00000002.png",
10    "name": "Mario",
11    "release": {
12      "au": "2014-11-29",
13      "eu": "2014-11-28",
14      "jp": "2014-12-06",
15      "na": "2014-11-21"
16    }
17  }
18 ]}
```

Luego de esto, se decidió investigar cómo funciona la llamada con parámetros. Esta estaba escrita con sintaxis completa dentro de la documentación, sin embargo, también es posible utilizar un Query que nos permita hacer la búsqueda solo con el parámetro utilizando Retrofit. Se decidió implementar la segunda opción para el programa, ya que

sería más flexible y sencilla su implementación. A continuación, se muestra esta llamada con parámetros desde Postman.



La API puede regresar varios registros asociados a un mismo nombre, por lo que también se deberá tomar esto en cuenta dentro del programa. Este método devuelve un array llamado amiibo que contiene toda la información dentro del json, la cual sera convertida en una data class dentro de Kotlin.

Como segundo método, se decidió utilizar la llamada *gameseries/* la cual lo único que hace es devolver todas las franquicias o series de videojuegos que tienen amiibos asociados a ellos. A continuación se puede ver la llamada desde Postman, y como devuelve un array que tiene objetos con solo 2 atributos, lo que facilitará la implementación de este.

Overview GET https GET https GET https GET https GET https GET https GET https + ... No Environment

https://www.amiiboapi.com/api/gameseries/ Save

GET https://www.amiiboapi.com/api/gameseries/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 315 ms Size: 7.49 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 2 "amiibo": [  
3     {  
4         "key": "0x000",  
5         "name": "Super Mario"  
6     },  
7     {  
8         "key": "0x001",  
9         "name": "Super Mario"  
10    },  
11    {  
12        "key": "0x002",  
13        "name": "Super Mario"  
14    },  
15    {  
16        "key": "0x008",  
17        "name": "Yoshi's Woolly World"  
18    },  
19    {  
20        "key": "0x00c",  
21        "name": "Donkey Kong"  
22    }  
23 ]
```

Postbot Runner Start Proxy Cookies Trash