



Alejandro Brugarolas Sánchez-Lidón

Curso 2º DAW

EFA El Campico

alejandrobregarolas@elcampico.org

PRÁCTICA ANGULAR | TOUR OF MARVEL

Realizar el proyecto propuesto

Objetivos

1. Conocer la forma de trabajar con el routing Angular.
2. Saber interpretar la documentación de una API REST.
3. Saber realizar y obtener los resultados de peticiones HTTP desde Angular.

Puesta en marcha

Proyecto Angular

Para esta práctica, se partirá del ejemplo provisto por Angular en el tutorial Tour Of Heroes.

El mismo se puede consultar desde <https://angular.io/tutorial> o descargar desde <https://angular.io/generated/zips/toh-pt6/toh-pt6.zip>.

Una vez descargado, para instalar todas las dependencias, hay que ejecutar en la carpeta raíz el comando ``npm install`` (Si diese un error, posiblemente sea por un fallo de la última versión de npm, se puede probar con el comando ``npm install --save --legacy-peer-deps``).

A continuación, será necesario actualizar el fichero ``app\app.module.ts`` para retirar la clase que intercepta las peticiones HTTP que se utilizan en el tutorial:

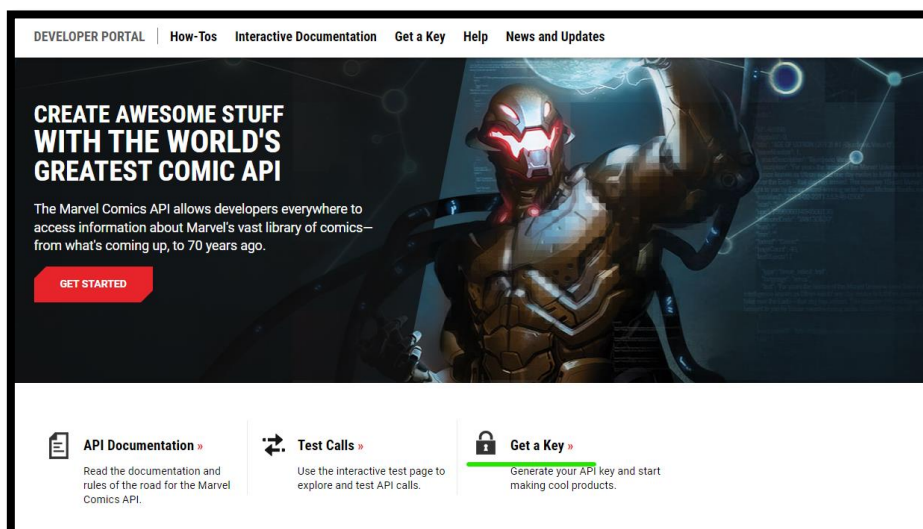
Borrar este trozo de código:

```
// The HttpClientInMemoryWebApiModule module intercepts HTTP requests
// and returns simulated server responses.
// Remove it when a real server is ready to receive requests.
HttpClientInMemoryWebApiModule.forRoot(
  InMemoryDataService, { dataEncapsulation: false }
)
```

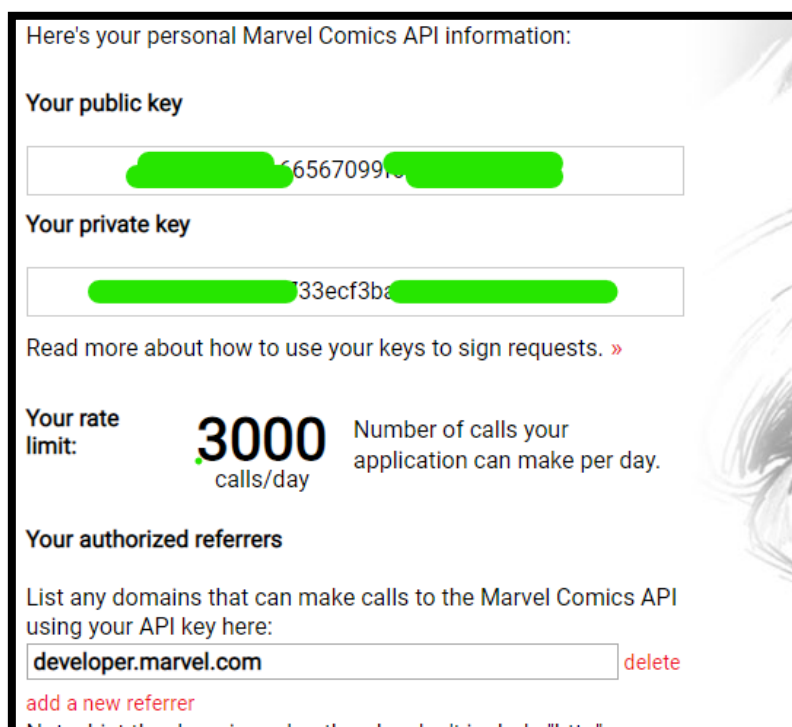
API Marvel

Para esta práctica se utilizará la API pública de personajes Marvel. Para ello, será necesario obtener una API Key desde la web <https://developer.marvel.com>.

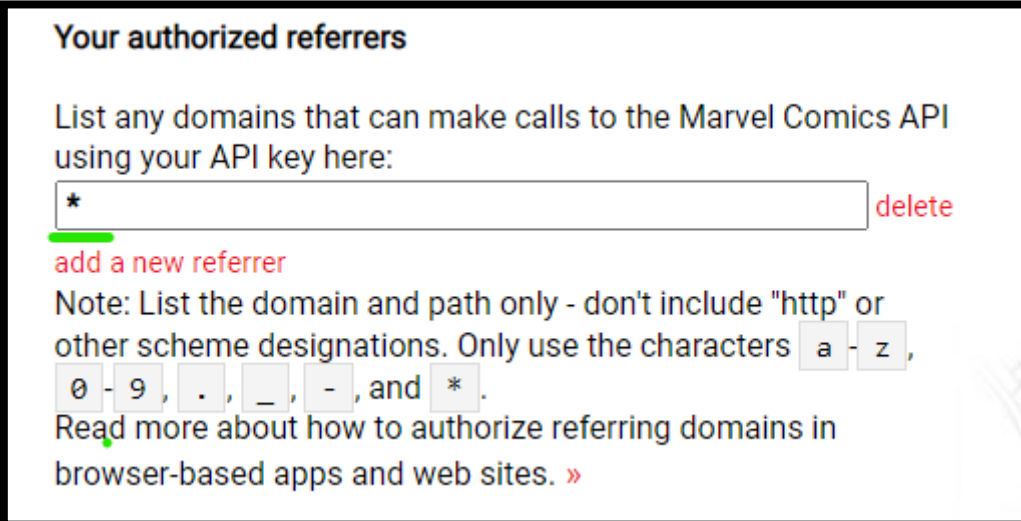
Una vez aquí, el alumno se registrará y pulsará sobre "Get a Key":



Al pulsar, se mostrará una clave pública y una privada con el límite de peticiones diarias para esa clave:



Con el fin de poder utilizar la API key desde cualquier web, será necesario modificar el apartado 'Your authorized referrers' para añadir la wildcard '*', debe quedar de la siguiente manera (Al hacer la modificación, no olvidéis pulsar sobre el botón 'Update' que se encuentra en la parte de abajo de la página):



Your authorized referrers

List any domains that can make calls to the Marvel Comics API using your API key here:

* delete

add a new referrer

Note: List the domain and path only - don't include "http" or other scheme designations. Only use the characters a - z , 0 - 9 , . , _ , - , and * .

Read more about how to authorize referring domains in browser-based apps and web sites. »


Por último, para realizar cualquier llamada a la API, será necesario enviar tres parámetros de la siguiente manera:

<http://gateway.marvel.com/v1/public/comics?ts=patata&apikey=1234&hash=ffd275c5130566a2916217b101f26150>

- El parámetro ``ts`` será cualquier string aleatorio, por ejemplo 'patata'.
- El parámetro ``apikey`` es la clave pública obtenida en el apartado anterior.
- El parámetro ``hash`` es un código md5 generado de la siguiente forma:

`md5(`ts+clavePrivada+clavePublica`)` sin incluir los símbolos +.

Por ejemplo: Para un ts = patata, una clave pública = 1234 y una clave privada = 456, el hash sería el siguiente:



MD5 Encryption

Enter a word here to get its MD5 hash :

patata4561234

Crypt

No credit required in this tool

The MD5 hash for patata4561234 is : dde78078846f6e64fd2e569c9c4096b

(Se pueden hacer cifrados en md5 online en la siguiente web: <https://www.md5online.org/md5-encrypt.html>).

Por lo que la llamada para obtener personajes sería:

<http://gateway.marvel.com/v1/public/comics?ts=1&apikey=1234&hash=ffd275c5130566a2916217b101f26150>.

Por último, si todos los pasos anteriores se han realizado correctamente, al utilizar una url de la api en el navegador con los parámetros correctos, deberemos poder obtener resultados:



```
1 // 20220205144029
2 // https://gateway.marvel.com/v1/public/characters?apikey=9c2b0e532a66567099f6a5ac9a120699&ts=patata&hash=ad291ac4ac4220584d31130af46ea7ba
3
4 {
5   "code": 200,
6   "status": "Ok",
7   "copyright": "© 2022 MARVEL",
8   "attributionText": "Data provided by Marvel. © 2022 MARVEL",
9   "attributionHTML": "<a href='\"http://marvel.com/\"'>Data provided by Marvel. © 2022 MARVEL</a>",
10  "etag": "57ee19981cb01ed081f2db8f41562405ce108e58",
11  "data": {
12    "offset": 0,
13    "limit": 20,
```

Todo lo aquí descrito se encuentra en la documentación oficial (en inglés) en el siguiente enlace: <https://developer.marvel.com/documentation/authorization>.

Enunciado

A partir del último punto del tutorial tour-of-heroes, el objetivo de la práctica será conectarlo con la API de Marvel implementando las siguientes modificaciones.

Dashboard

(2ptos) En el Dashboard ahora aparecerán personajes de Marvel (nombre y foto) (https://developer.marvel.com/docs#!/public/getCreatorCollection_get_0).

(1pto) Los personajes mostrados en el dashboard serán aleatorios con cada recarga de la página.

(2ptos) Se podrá filtrar por nombre, siendo necesario que el filtro se haga como petición a la API.

Listado:

(1pto) El listado mostrará personajes también obtenidos de la API

(2pto) La lista deberá paginarse (en el lado del servidor) para que únicamente muestre 20 en pantalla al mismo tiempo.

Se mostrará id y nombre en cada elemento de la lista

(https://developer.marvel.com/docs#!/public/getCreatorCollection_get_0)

Detalle:

(2ptos) En el detalle se tendrá que obtener el personaje con la petición concreta de cada personaje (https://developer.marvel.com/docs#!/public/getCharacterIndividual_get_1).

Aquí se mostrará: Nombre, Foto, Id y algún otro dato que el alumno desee.

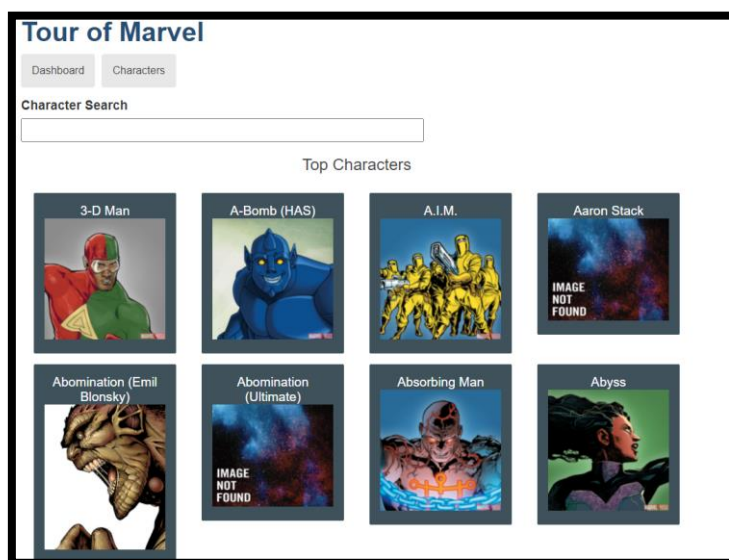
Requisitos

1. Todas las llamadas a la API se realizarán desde servicios.
2. Será necesario utilizar Observables.
3. Será necesario trabajar con el routing de Angular, de forma que el cambio de páginas no suponga una recarga de la página ni el desarrollo se haga todo en un mismo componente con directivas ngIf.
4. Será necesario implementar un mínimo de una interfaz.

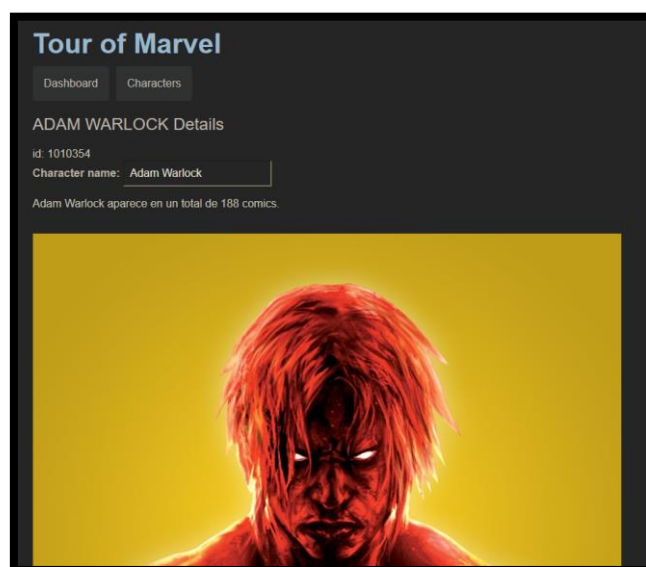
Ejemplo del resultado esperado

La práctica, desarrollada por el profesor, se puede probar en <https://tom.brugui.dev>.

Dashboard



Detalle



Búsqueda

Tour of Marvel

[Dashboard](#)[Characters](#)

Character Search

iron man

Iron Man
Iron Man (Iron Man 3 - The Official Game)
Iron Man (LEGO Marvel Super Heroes)
Iron Man (Marvel Heroes)
Iron Man (Marvel War of Heroes)

Listado

Tour of Marvel By Brugui

[Dashboard](#)[Characters](#)

Characters

1011334	3-D Man
1017100	A-Bomb (HAS)
1009144	A.I.M.
1010699	Aaron Stack
1009146	Abomination (Emil Blonsky)
1016823	Abomination (Ultimate)
1009148	Absorbing Man
1009149	Abyss
1010903	Abyss (Age of Apocalypse)
1011266	Adam Destine
1010354	Adam Warlock
1010846	Aegis (Trey Rollins)

Se valorará positivamente

1. El correcto uso de las instrucciones de Typescript.
2. La limpieza del código y su correcta estructuración.

Entrega

La entrega se realizará subiendo el proyecto a GitHub y compartiendo el repositorio al email alejandrobrugarolas@elcampico.org. Posteriormente, se copiará el enlace **de la web del repositorio** en la entrega abierta en Moodle.

Aclaración

Aunque se haya compartido el repositorio por email, será necesario enviarlo a través de la tarea en Moodle para poder calificar la tarea y que el trabajo aparezca como enviado.

La URL de la web del repositorio **no** es la que termina en .git, si no la que aparece en el navegador cuando se accede al repositorio.

