

## API para consumir películas.

La solución planteada fue desarrollada en JavaScript bajo el entorno de desarrollo Node.js versión 8. Para la implementación del server se ha utilizado el framework Express y se ha utilizado el módulo node-fetch para poder realizar las consultas a la API externa The Movie Data Base. Los endpoints implementados para realizar dichas consultas pueden ser consumidos mediante la aplicación PostMan (adjunto a este documento se encuentra *Postman collection* con las consultas precargadas). Se ha utilizado JWT para poder realizar la autenticación de los usuarios. Estos y las películas registradas como favoritas son persistidas en los archivos users.txt y favourites.txt respectivamente.

Para cada respuesta se indica el status de la misma. 200 cuando todo ha salido bien, 400 cuando hay un error del usuario, como por ejemplo introducir la contraseña mal y 500 cuando el error es del sistema, como, por ejemplo, leer el archivo users.txt.

Todas las respuestas son enviadas en formato JSON con excepción del endpoint *Obtener películas favoritas* ya que se obtiene una respuesta en texto plano no pudiéndose obtenerla en dicho formato.

De ser necesario el archivo users.txt ya contiene un usuario persistido con los siguientes datos:

email: "phoebe@gmail.com"

password: "smellycat"

En la siguiente tabla se puede encontrar para cada requerimiento los endpoints a utilizar.

Requerimiento	Endpoint
Registrar usuario	http://localhost:3000/api/users
Autenticar usuario	http://localhost:3000/api/login
Obtener películas	http://localhost:3000/api/movies
Agregar película a favoritos	http://localhost:3000/api/favourites
Obtener películas favoritas	http://localhost:3000/api/favourites

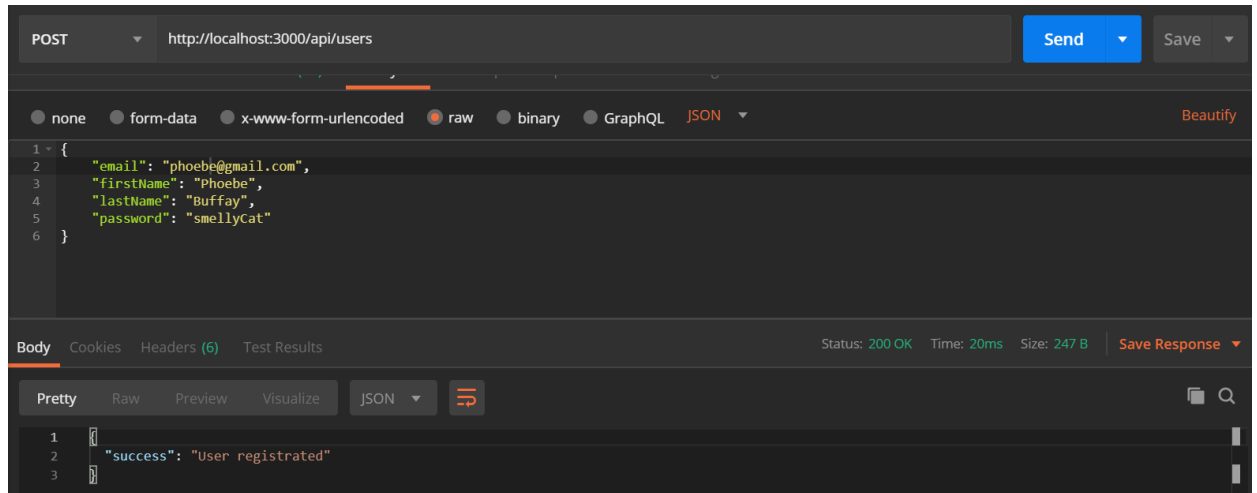
Para las consultas realizadas a la API externa se han optado por la utilización de los siguientes endpoints ya que son los que más se adaptan a los requerimientos planteados:

Requerimiento	Endpoint
Obtener películas	/discover/movie
Obtener películas dada una keyword	/keyword/{keyword_id}/movies

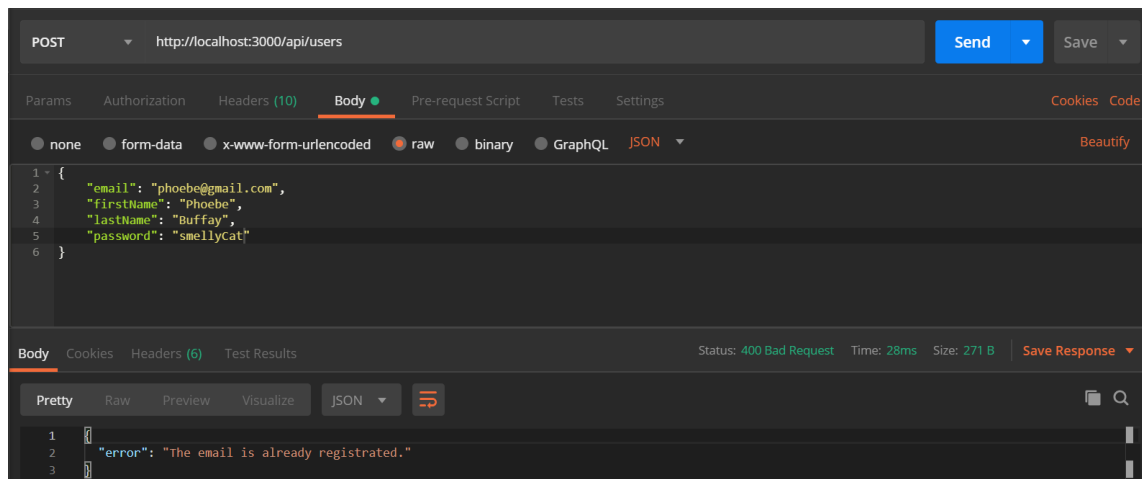
A continuación, se sugiere un flujo para probar cada uno de los requerimientos con sus respectivas respuestas esperadas.

### Registrar usuario

Caso datos correctos:

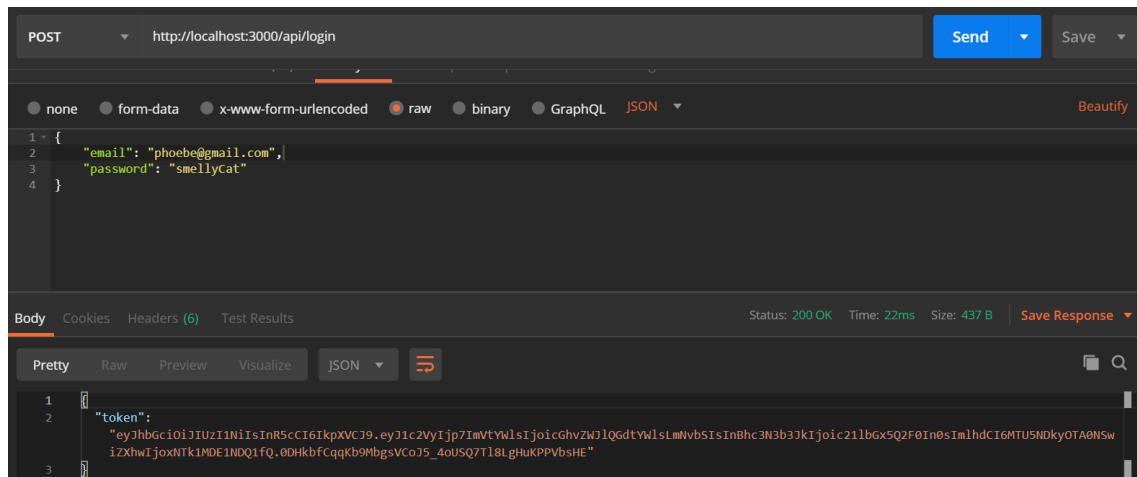


Caso usuario ya registrado:

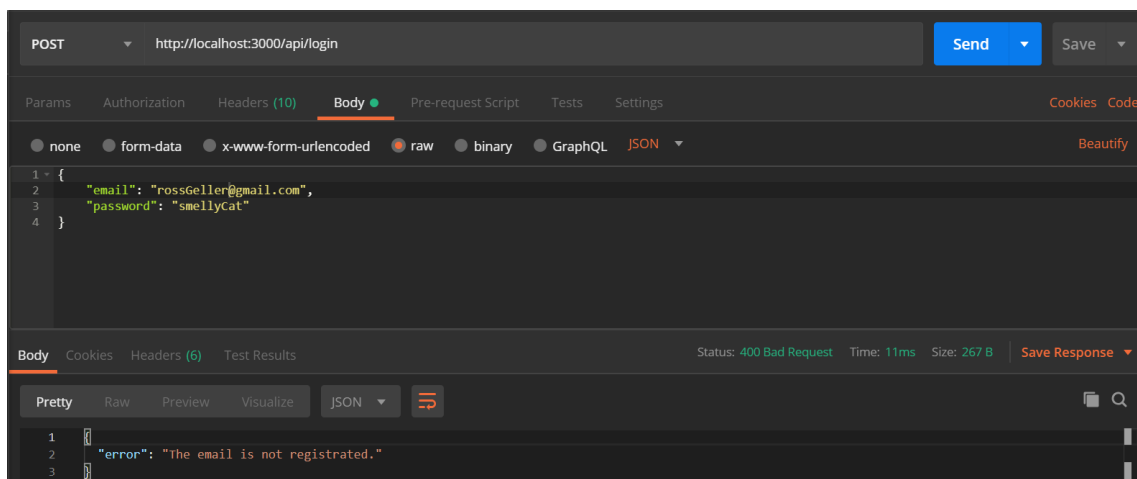


## Autenticar usuario

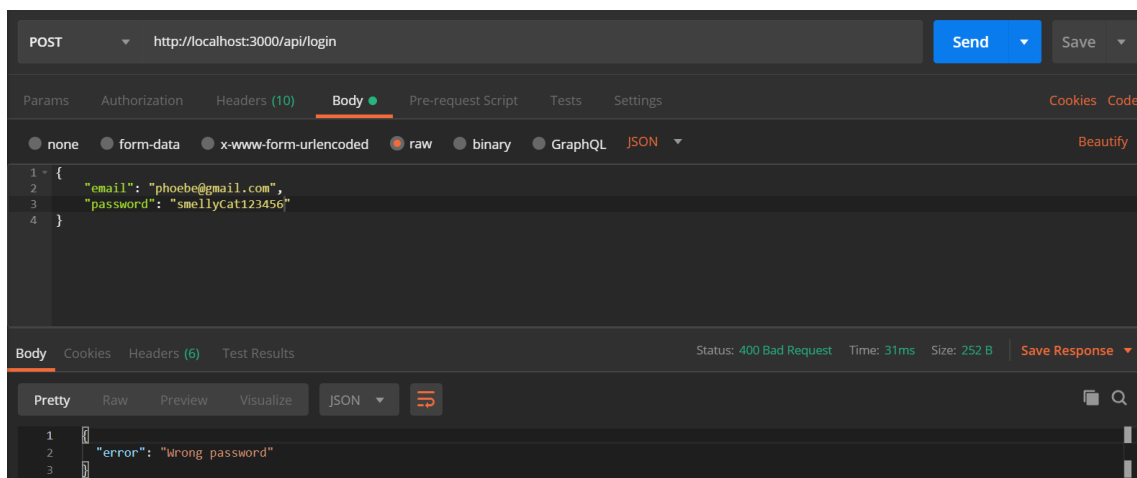
Caso datos correctos:



Caso email no registrado:



Caso contraseña incorrecta:



Obtener películas

Caso usuario no autenticado:

GET

http://localhost:3000/api/movies

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

▼ Headers (2)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
<input type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6IjE2MzQ1MjM0IiwiaWF0IjoxNjM0MjM0MjM0In0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6IjE2MzQ1MjM0IiwiaWF0IjoxNjM0MjM0MjM0In0.				
	Key	Value	Description			

▶ Temporary Headers (8)

Body

Cookies

Headers (6)

Test Results

Status: 400 Bad Request

Time: 51ms

Size: 252 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{  
 "error": "Action denied."  
}

Caso usuario autenticado:

GET

http://localhost:3000/api/movies

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

▼ Headers (2)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6IjE2MzQ1MjM0IiwiaWF0IjoxNjM0MjM0MjM0In0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6IjE2MzQ1MjM0IiwiaWF0IjoxNjM0MjM0MjM0In0.				
	Key	Value	Description			

▶ Temporary Headers (8)

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 73ms

Size: 15.57 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{  
 "popularity": 111.923,  
}

### Agregar película a favoritos

Caso usuario no autenticado:

POST

http://localhost:3000/api/favourites

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

▼ Headers (2)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
<input type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2V5Ij...				
	Key	Value	Description			

▶ Temporary Headers (8)

Body

Cookies

Headers (6)

Test Results

Status: 500 Internal Server Error

Time: 11ms

Size: 280 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{

"error": "Action denied. Unauthorized user"

}

Caso usuario autenticado:

POST

http://localhost:3000/api/favourites

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON

Beautify

1

2

3

4

5

6

7

8

9

10

{

"popularity": 111.923,

"vote\_count": 13729,

"video": false,

"poster\_path": "/udbc1JoHjfjb8Eksd4FDte0kCU.jpg",

"id": 475557,

"adult": false,

"backdrop\_path": "/n6bUvigpRFqSwmPp1m2YADdbR8c.jpg",

"original\_language": "en",

"original\_title": "Joker",

}

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 57ms

Size: 256 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{

"success": "Movie added to favourites"

}

### Obtener películas favoritas

Caso usuario no autenticado:

GET

http://localhost:3000/api/favourites

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

▼ Headers (2)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
<input type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2V5Ij...				
	Key	Value	Description			

▶ Temporary Headers (8)

Body

Cookies

Headers (6)

Test Results

Status: 500 Internal Server Error

Time: 16ms

Size: 280 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{

"error": "Action denied. Unauthorized user"

}

### Caso usuario autenticado:

The screenshot shows the Postman application window. At the top, the method is set to "GET" and the URL is "http://localhost:3000/api/favourites". The "Send" button is highlighted in blue. Below the URL bar, there are tabs for "Params", "Authorization", "Headers (10)", "Body", "Pre-request Script", "Tests", and "Settings". The "Headers" tab is selected, displaying a table of headers:

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9keW8udmVpZS1leSI6ImNpbGUiLCJyb3kiOiJkbGVhbnQifQ.SwmPp1m2YADdbR8c.jpg1				
	Key	Value	Description			

Below the header table, there is a section for "Temporary Headers (8)" with an information icon. At the bottom of the interface, the "Body" tab is selected, showing the response status as "Status: 200 OK", time as "Time: 70ms", size as "Size: 1.54 KB", and a "Save Response" button. The response body is displayed in JSON format:

```
{\n  \"popularity\":111.923,\n  \"vote_count\":13729,\n  \"video\":false,\n  \"poster_path\":\"/udDclJoHfjfb8EKgsd4FDteOkCU.jpg1\",\n  \"id\":475557,\n  \"adult\":false,\n  \"backdrop_path\":\"/n6buvgprFqSwmPp1m2YADdbR8c.jpg1\",\n  \"original_language\":\"en\",\n  \"original_title\":\"Joker\\\",\\n  \"genre_ids\":[80,18,53],\\n  \"title\":\"Joker\\\",\\n  \"vote_average\":8.2,\\n  \"overview\":\"During the 1980s, a failed stand-up comedian is driven insane and turns to a life of crime and chaos in Gotham City while becoming an infamous psychopathic crime figure.\"}
```