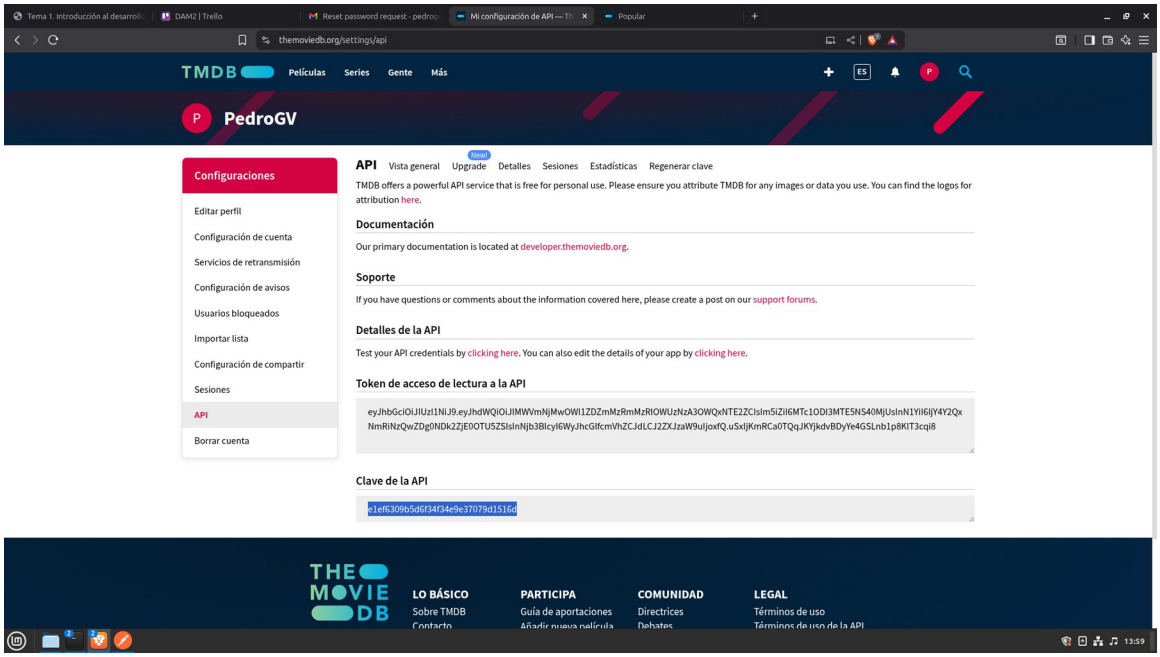
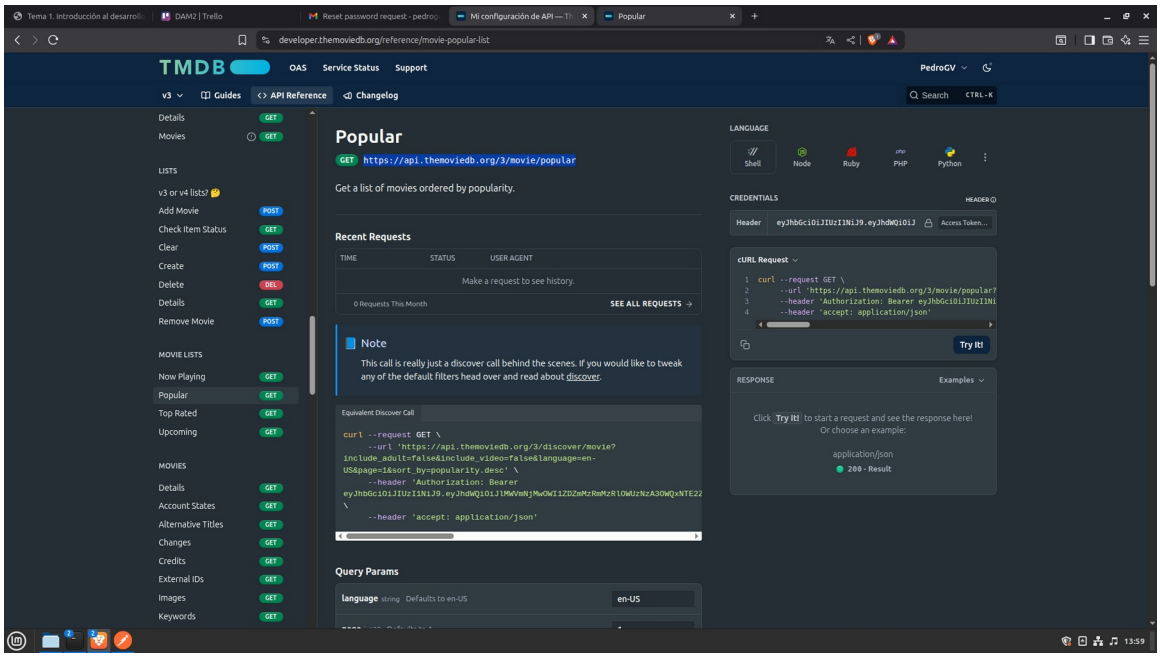
4. Consulta una API de películas con una API Key

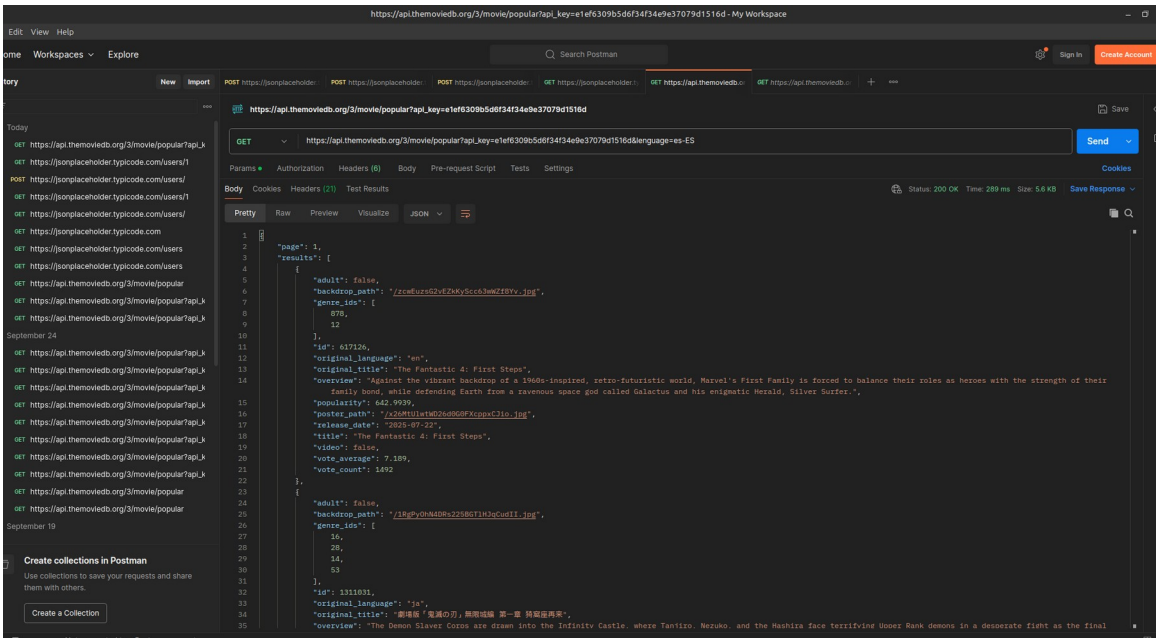
a. Regístrate en la página web de The Movie Database (TMDb). Una vez iniciada la sesión, accede a Ajustes en tu perfil y después a la sección API del menú lateral izquierdo para obtener tu API Key.



b. Consulta la referencia de la API y usa Postman para realizar una solicitud HTTP GET que obtenga la lista de películas populares.

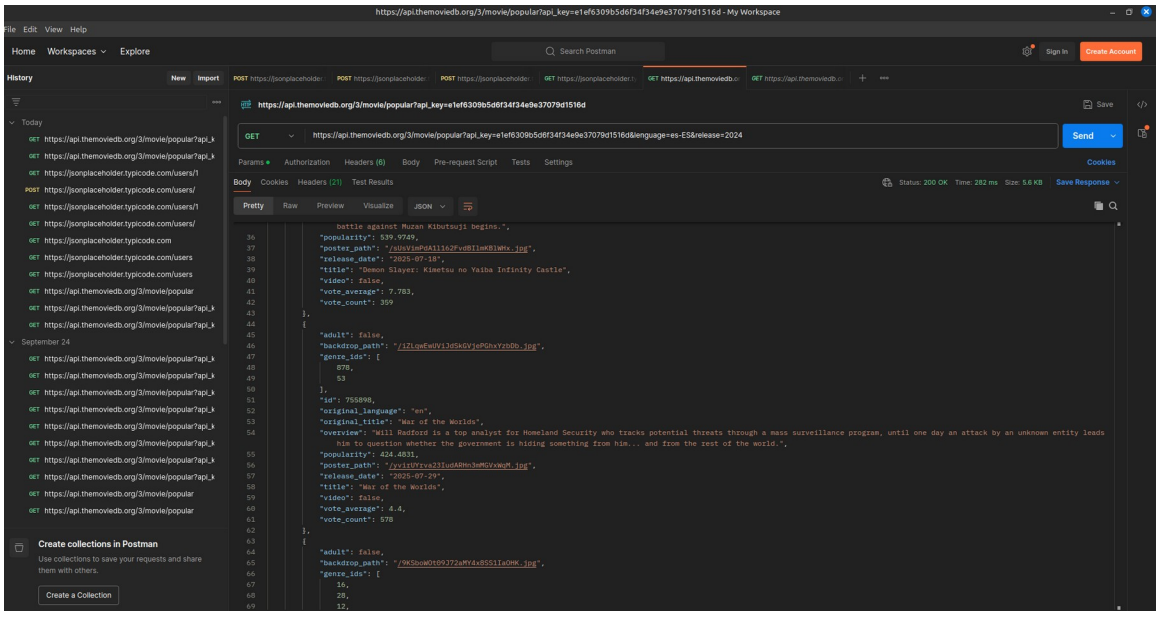


GET https://api.themoviedb.org/3/movie/popular?api_key=TU_API_KEY&language=es-ES&page=1



c. Busca en Postman todas las películas del año 2024.

GET https://api.themoviedb.org/3/discover/movie?api_key=TU_API_KEY&language=es-ES&primary_release_year=2024



d. Busca las películas de la saga de Star Wars.

```
GET https://api.themoviedb.org/3/search/collection?
api_key=TU_API_KEY&language=es-ES&query=Star%20Wars
```

e. Busca las películas en las que haya actuado Cillian Murphy.

```
GET https://api.themoviedb.org/3/search/person?
api_key=TU_API_KEY&language=es-ES&query=Cillian%20Murphy
```

5. Crea una API de pruebas mediante json-server

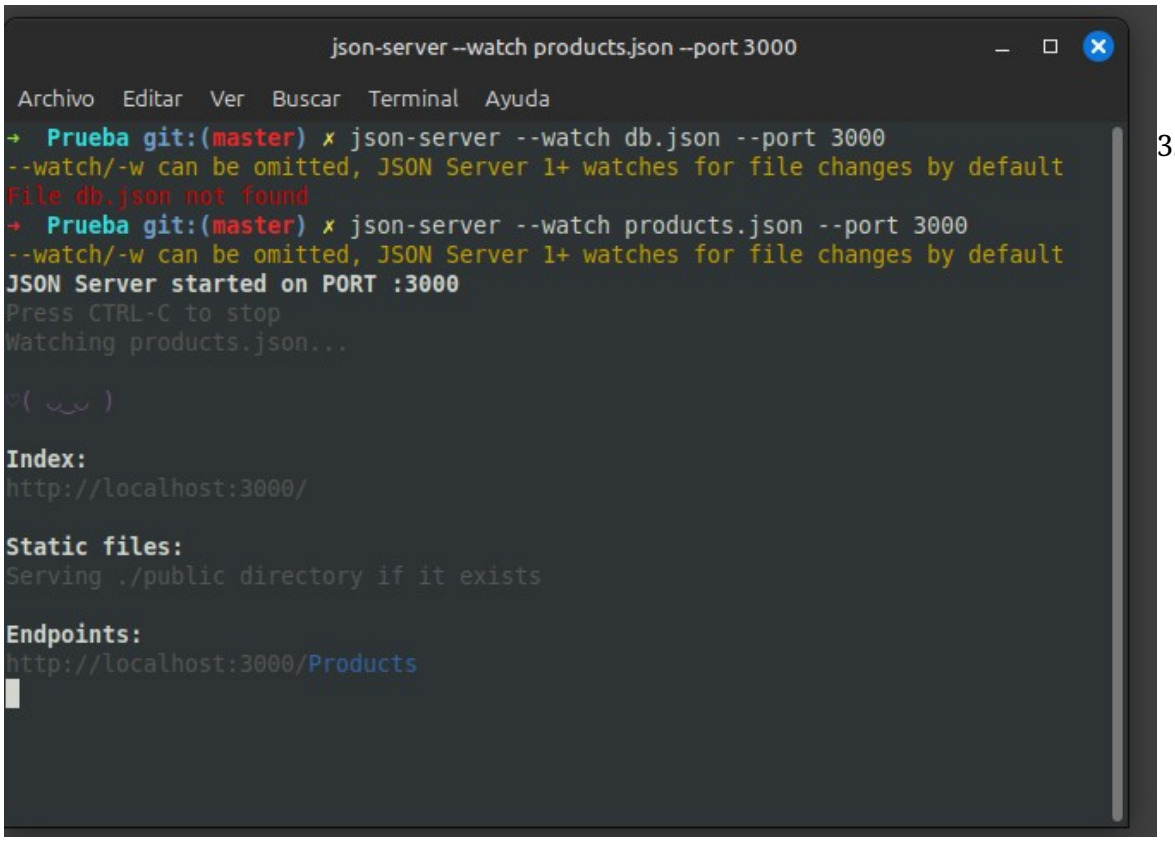
Para crear una API de pruebas con json-server:

1. Crea un fichero JSON con los datos de productos de una tienda de informática (por ejemplo, db.json).

```
{
  "Products": [
    {
      "id": "1",
      "name": "Monitor de 24 pulgadas",
      "pvp": 80.51,
      "description": "Monitor led"
    },
    {
      "id": "2",
      "name": "Monitor de 24 pulgadas",
      "pvp": 80.51,
      "description": "Monitor led"
    },
    {
      "id": "3",
      "name": "Monitor de 24 pulgadas",
      "pvp": 80.51,
      "description": "Monitor led"
    }
  ]
}
```

2. Ejecuta el servidor con el comando:

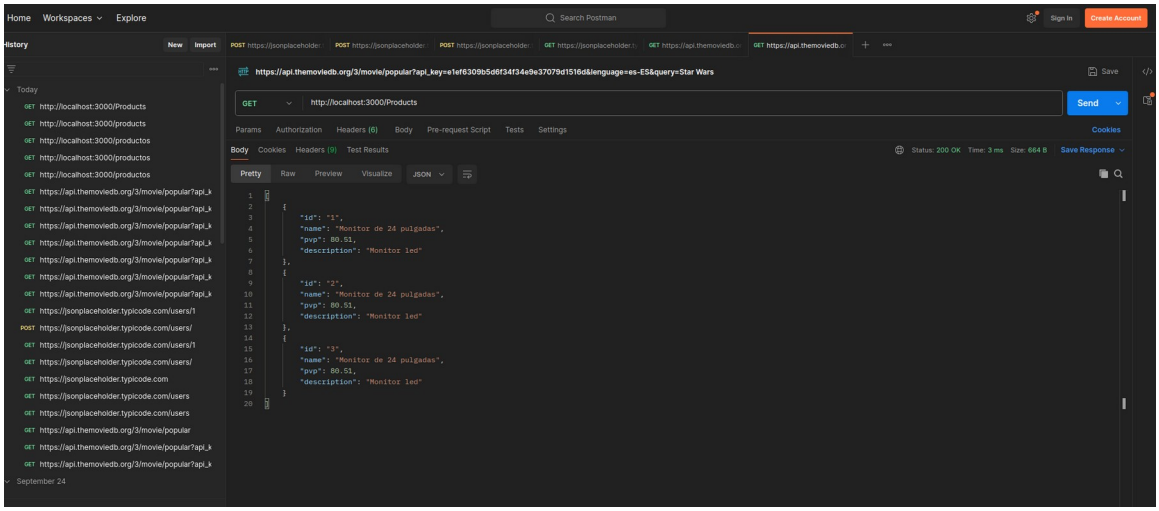
```
json-server --watch db.json --port 3000
```



3. Usa Postman para realizar las siguientes solicitudes:

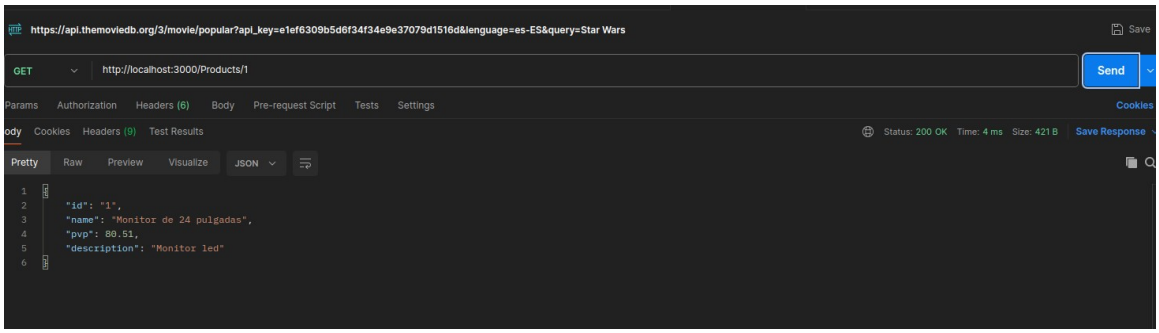
a. Obtener todos los productos:

```
GET http://localhost:3000/Products
```



b. Obtener un producto por su id:

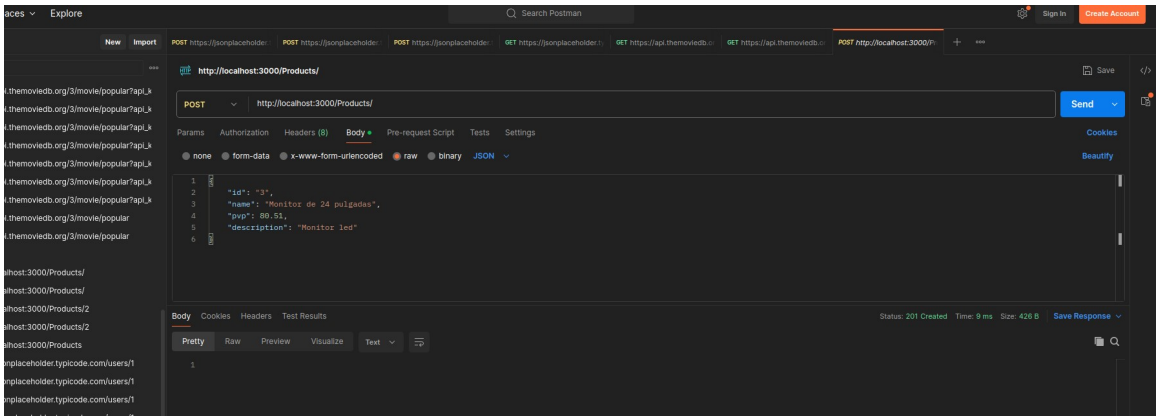
```
GET http://localhost:3000/Products/1
```



c. Crear un nuevo producto:

```
POST http://localhost:3000/Products
Content-Type: application/json

{
  "nombre": "Impresora",
  "precio": 80
}
```



d. Actualizar un producto existente con el método PUT:

```
PUT http://localhost:3000/Products/1
Content-Type: application/json

{
  "id": 1,
  "nombre": "Teclado mecánico",
  "precio": 50
}
```

e. Actualizar un producto existente con el método PATCH:

```
PATCH http://localhost:3000/Products/2
Content-Type: application/json

{
  "precio": 18
}
```

f. Eliminar un producto:

```
DELETE http://localhost:3000/Products/3
```