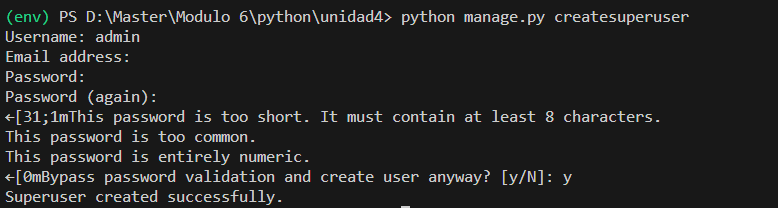
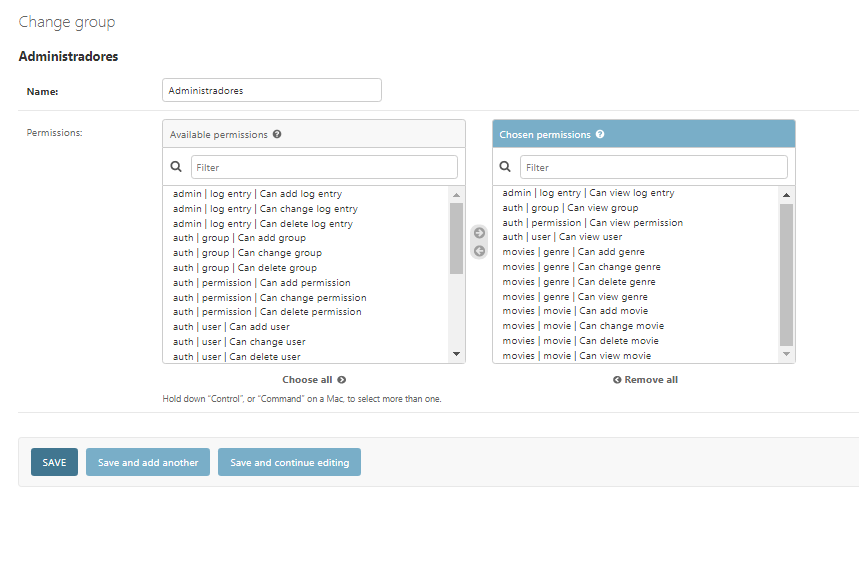
# Gestión de usuarios

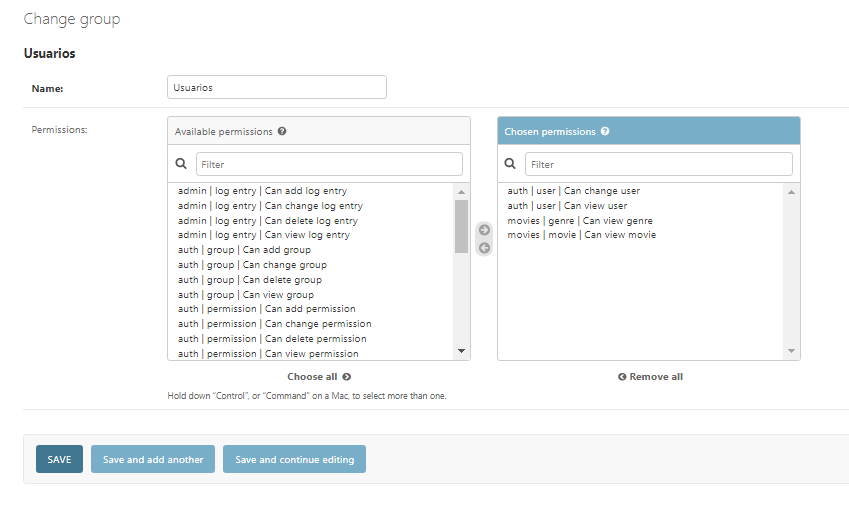
## Creamos un superusuario para acceder al administrador de Django

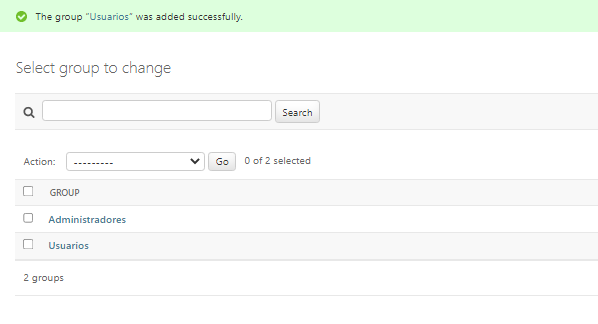


## Creamos un usuario: “Maria”

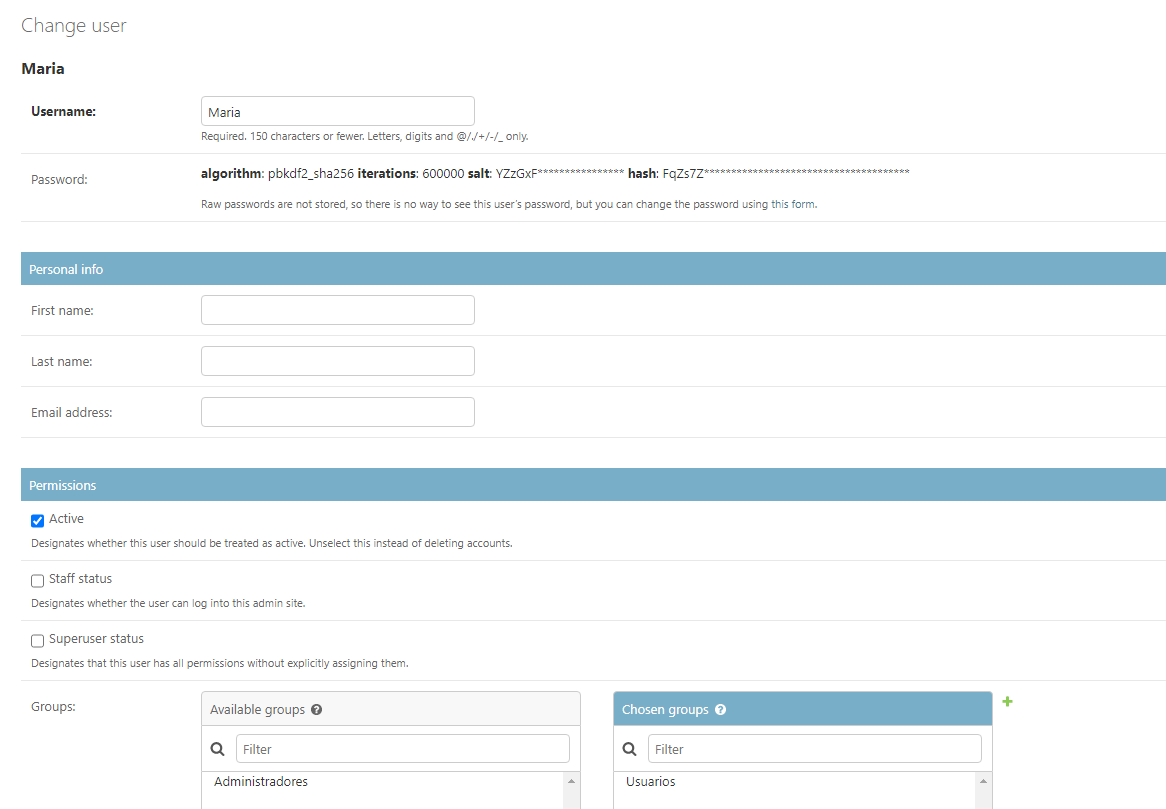
## Creamos dos grupos: “Administradores” y “Usuarios”



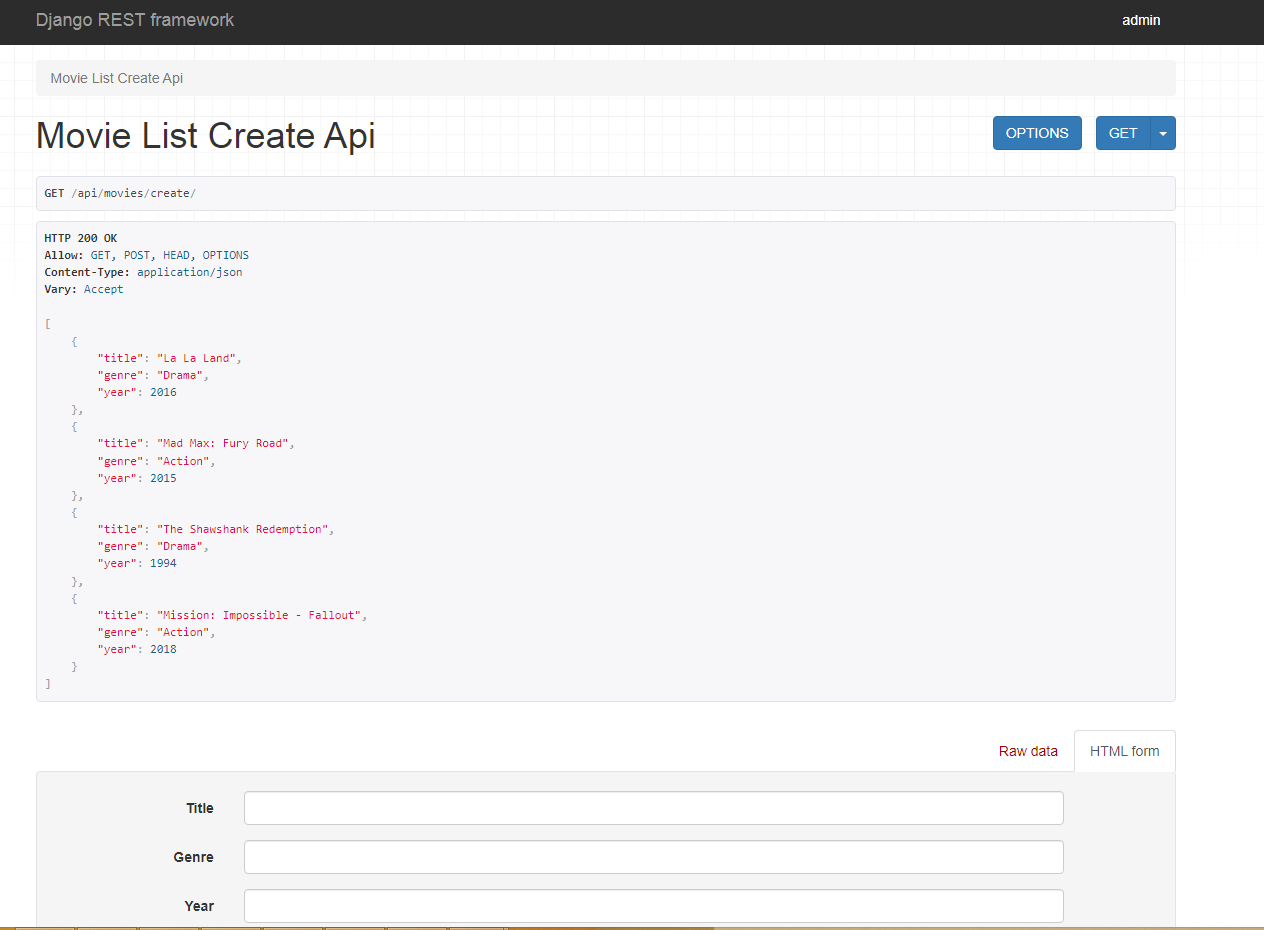


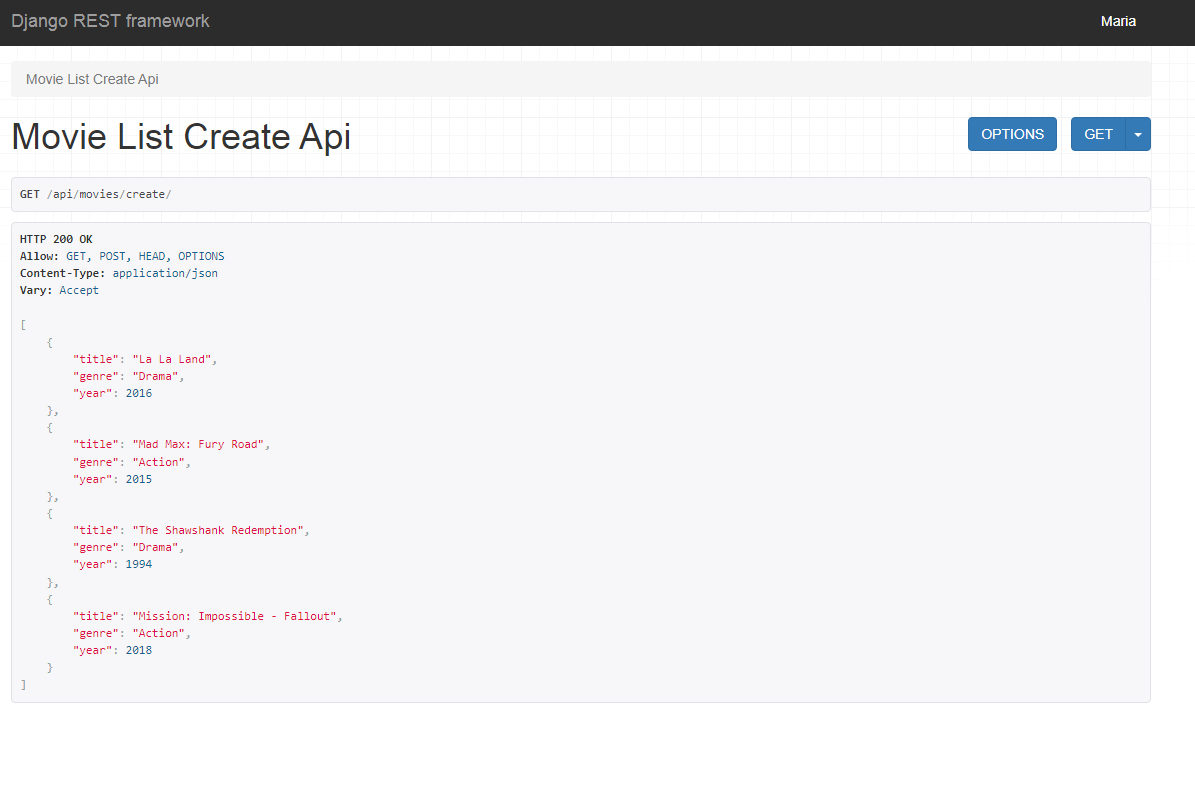


## Asignamos el grupo “Usuarios” al usuario Maria, “Administradores” al usuario admin

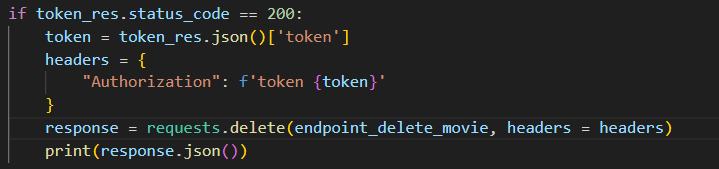


## Comprobamos que al acceder al url de create/list movies como administrador, tenemos la opción de añadir datos, al acceder como Maria,podemos solamente listar datos

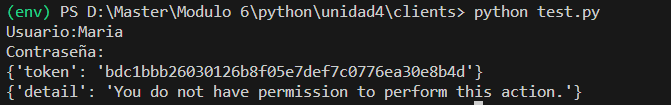


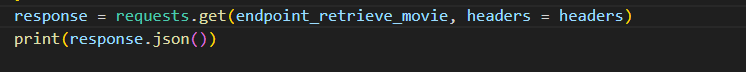


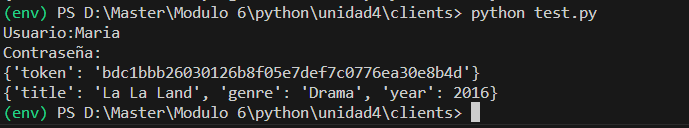
## Añadimos asigración de los tokens



## Hacemos comprobaciones de acceso a diferentes endpoints como usuario Maria

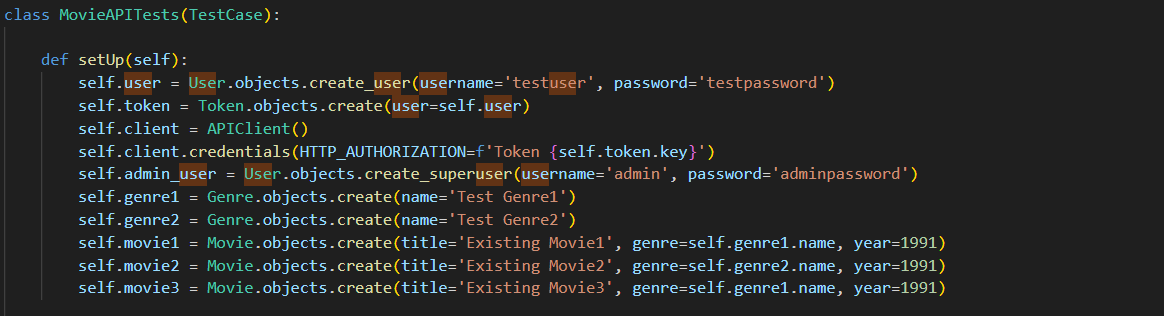




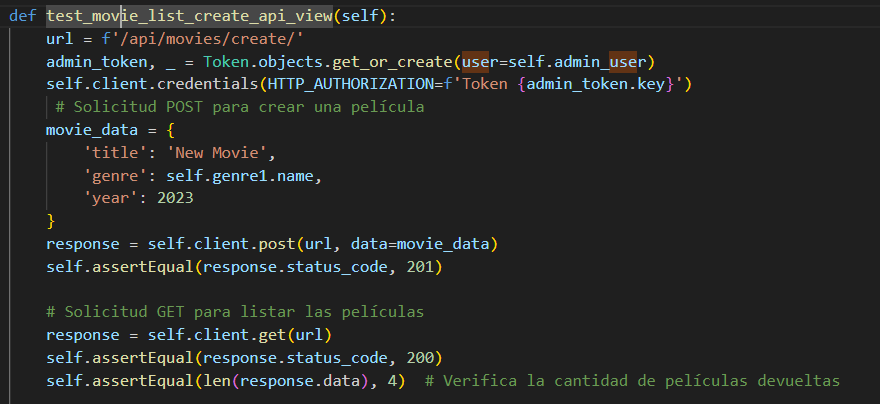


# Implementar los tests unitarios necesarios para comprobar que la API funciona

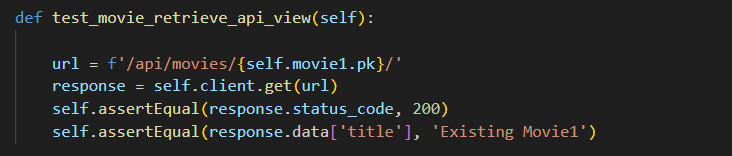
# Implementamos 3 tests unitarios. Como al algunos enpoints se puede acceder solamente con los permisos de administrador, configuramos los usuarios dentro de los tests



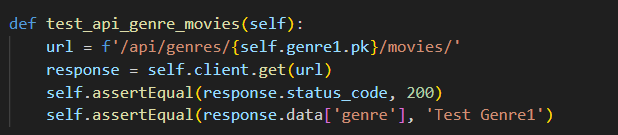
## Verificar el funcionamiento de la vista MovieListCreateAPIView (listar y crear películas)



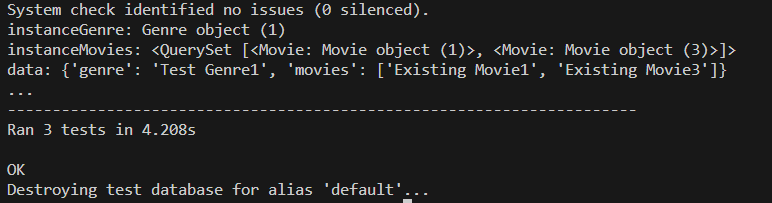
## Verificar MovieRetrieveAPIView que que obtiene los detalles de una película específica



## Verificar api\_genre\_movies, que devuelve las películas de un género específico



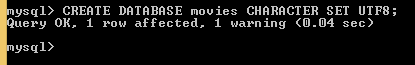
## Ejecutamos los tests



# Implementar al menos 1 viewset

# Utilizar MySQL como BBD

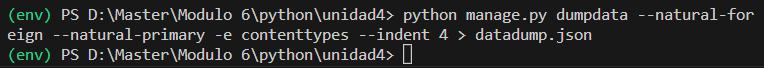
## Creamos una base de datos







## Hacemos el dump de data del proyecto



## Comprobamos el contenido del archivo datadump.json

[

{

    "model": "movies.genre",

    "pk": 1,

    "fields": {

        "name": "Action"

    }

},

{

    "model": "movies.genre",

    "pk": 2,

    "fields": {

        "name": "Drama"

    }

},

{

    "model": "movies.movie",

    "pk": 2,

    "fields": {

        "title": "La La Land",

        "genre": "Drama",

        "year": 2016

    }

},

{

    "model": "movies.movie",

    "pk": 3,

    "fields": {

        "title": "Mad Max: Fury Road",

        "genre": "Action",

        "year": 2015

    }

},

{

    "model": "movies.movie",

    "pk": 4,

    "fields": {

        "title": "The Shawshank Redemption",

        "genre": "Drama",

        "year": 1994

    }

},

{

    "model": "movies.movie",

    "pk": 5,

    "fields": {

        "title": "Mission: Impossible - Fallout",

        "genre": "Action",

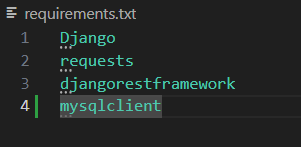
        "year": 2018

    }

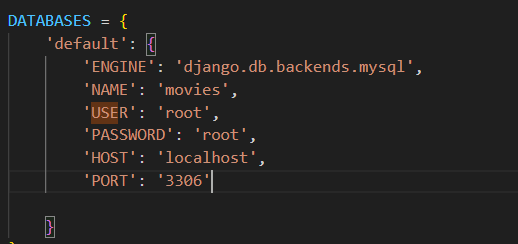
}

]

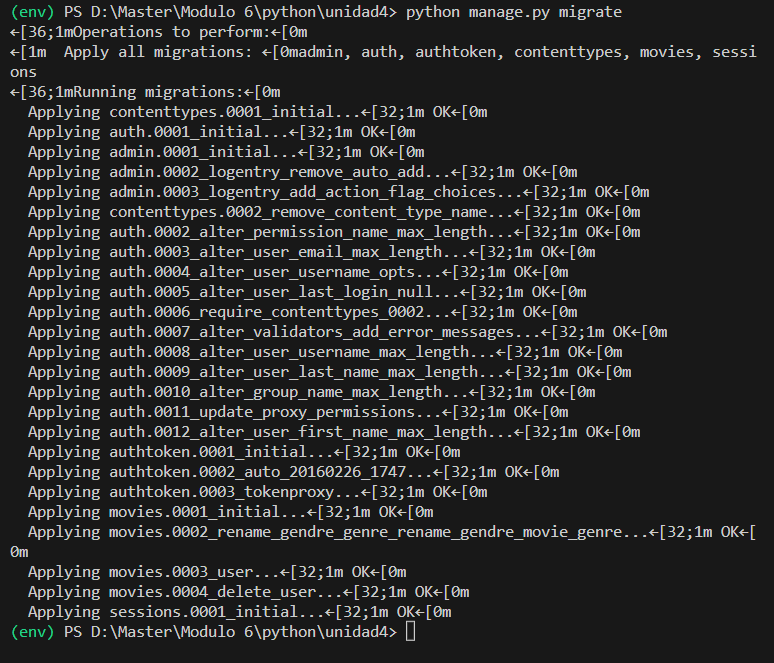
## Instalamos mysqlclient



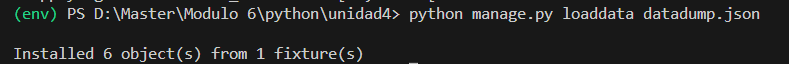
## Configuramos el archivo de settings



## Ejecutamos el comando python manage.py migrate



## Hacemos la trasferencia de los datos



# Observaciones:

Al hacer la migración de los datos a mysql se pierden los usuarios, grupos y permisos.

Gracias por tu comentario de lo de añadir frontend. De echo, el tema de mi TMF tambíen es desarrollo de una aplicación de peliculas, entonces espero poder aprovechar de algunos partes del código y convertirlo en algo más o menos digno ☺