

INFORME PARA TRABAJO PRÁCTICO DE API DE “RICK Y MORTY” UNGS COMISIÓN 06 GRUPO 6

Nicolás Altamirano, Ignacio Juárez,
Cardozo Leonardo

Turno: 13:00 a 17:00

Materia: Introducción a la
programación

Introducción

En el trabajo presente se encontró una api incompleta cuyas funcionalidades principales no se hallaban llevándose a cabo. Las funciones de los archivos .py estaban sin terminar (junto con un template) por cuya razón lo que se buscó realizar fue completarlas para que la función básica de la visualización de elementos que ya se encontraban dentro del programa, estén en correcto estado.

Desarrollo

Se focalizó el análisis del problema en tres archivos principales, views.py, services.py y home.html. El planteamiento estuvo direccionado a qué era lo que posibilitaba que las tarjetas mostraran información, color dependiendo del estado del personaje ficticio en la serie e imagen. Por lo tanto, la investigación hecha fue en base a recursos que nos ayudaran a saber qué funciones eran las adecuadas para cada situación.

Primero en principal, se armó un repositorio remoto para que los tres integrantes del grupo clonaran el programa en sus computadoras por separado. Se decidió que cada uno por su parte se encargue de cada uno de los archivos.

En el views.py se llamó a la función getAllImages del archivo service que devuelve una lista con la información de todos los personajes en formato de card (desarrollada en services.py). Además que se implementó la funcionalidad de la función home() llamando a la función getAllFavourites de services.py. Junto con la implementación de getAllFavouritesByUser, llamando a la función service.getAllFavourites() dentro de favourite_list.

En el services.py se desarrolló en profundidad la función getAllImages y getAllFavourites. saveFavourite y deleteFavourite fueron completadas con la finalidad de usarlas después de forma provisoria (es decir, no se focalizó todo el trabajo en esas dos últimas). En getAllImages se decidió usar un for para ir agregando a la lista images los elementos que arrojaba json_collection, convirtiéndolos en cards previamente. En getAllFavourites usamos un for para agregar a la lista de favoritos todos los elementos de la lista de favourite_list que pasó a valer todo lo que el usuario devolvía al interactuar con la página.

En el home.html se utilizó clases del Bootstrap para establecer los colores del borde, border success para los personajes vivos, border-danger y border-warning para personajes con estado desconocido, incluyendo un condicional para mostrar cada caso con cada color correspondiente al estado del personaje.

Por lo tanto, de opcionales se realizó el inicio de sesión solo importando funciones y organizarlas adecuadamente junto a la selección de favoritos.

Problemas surgidos

En un principio costó entender que debíamos solo de completar las funciones e implementar otros recursos son modificar mucho el código, con lo que tuvimos que recurrir a realizar otro repositorio de nuevo para poder hacer el trabajo más prolijo una vez sabiendo qué es lo que debíamos hacer en específico, ya que con un par de aclaraciones del docente entendimos qué debíamos de modificar.

Conclusión

En resumen, se investigó la documentación presentada en el repositorio para proceder a leer los archivos de python con la finalidad de saber qué funciones importar en cada caso dependiendo de las anotaciones que aparecían. Se entendió que aquellas cumplen la finalidad de poder usarse como recurso principal para lograr hacer funcionar cuestiones como la visualización, la organización de información y el almacenamiento de datos particulares en, por ejemplo, forma de cards para ser plasmadas en la página

Datos a aclarar

La funcionalidad de los favoritos da error al querer guardarlos pero aún así se conservan en la lista de favoritos de la página bajo el nombre del usuario "admin".