Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2023.

API PARA EL PROCESAMIENTO DE DATOS DE LA PLATAFORMA CHAPINCHAT Y UN FRONTEND PARA TESTEO

202109114 – Sebastian Alejandro Vásquez Cartagena

Resumen

Se elaboró una API cuya utilidad es el procesamiento de datos dentro de archivos XML, para dicha API se utilizó el framework escrito en python Flask. Dicha API cuenta con capacidades para el procesamiento de información que requieren los servicios de la plataforma ChapinChat, estos son el guardado de usuarios, mensajes y el guardado y creación de los perfiles de los usuarios, estos perfiles cuentan con palabras clave, que, al ser solicitado, se pueden hacer cálculos sobre los mensajes que cada usuario de la plataforma ha creado, y de esta forma obtener una estadística que asigna a cada mensaje del usuario la probabilidad de que su mensaje pertenezca a cada uno de los perfiles. De igual forma es posible obtener un peso que dados todos los mensajes que los usuarios han cargado, se le asigne que tanto el usuario habla sobre un perfil en especifico.
También para el fácil testeo de los desarrolladores de ChapinChat se elaboró un frontend para cargar archivos XML y ver de forma más visual las capacidades de la plataforma.

Palabras clave
API Python Red Social Frontend Backend

Abstract

An API was developed to process data from XML file, the API used the framework writed in python called Flask, this API is capable of the processing the information that the services of the plataform ChapinChat required, this services are the storing of users, messages and the storing and creation of the user profiles. This profiles have keywords, that, at the time of being requested, can be used to make calculations with every individual message of every user in the plataform had created, and, with this method is possible to get an statistic that assigns to every message of every user the probability of being from any of the stored profiles. Likewise is also possible of obtaining a weight of all the messages that a user had loaded, and the user is assigned with how much them talk about a specific profile.
Also for the ease of testing for the developers at ChapinChat a front end was created for loading XML files and view in a more visual way the capability of the plataform.

Keywords
API python Social Network Frontend Backend

Introducción

El proyecto que has descrito es una aplicación web que consta de una API y un frontend. La API está hecha en Python con Flask y se utiliza para brindar estadísticas a una red social ficticia llamada ChapinChat. El frontend, por otro lado, está hecho en Django y consta de varias aplicaciones para diferentes funcionalidades.

Desarrollo del tema

En cuanto a la API, esta tiene una base de datos hecha de un archivo XML que almacena la información recibida por los endpoints y es utilizada por cada uno de los endpoints. La API ofrece seis endpoints diferentes, cada uno con su método HTTP correspondiente: POST, GET y DELETE. Los endpoints se encargan de tareas específicas, como recibir perfiles de usuario, mensajes y devolver información sobre ellos. Además, la API también se encarga de calcular el peso de cada perfil de usuario y de resetear la base de datos.

En cuanto al frontend, se divide en varias aplicaciones. La primera de ellas es llamada "Components" y se encarga de mostrar una template que permite cargar ya sea un archivo XML con perfiles o mensajes, y enviarlos a la API y mostrar el XML formateado en la pantalla. La segunda aplicación es "Peticiones" y consta de tres templates, una para buscar por las estadísticas por usuario con distintas opciones de filtrado, lo mismo para obtener los pesos, ambos muestran un pdf con el resultado, y una para cargar un mensaje de prueba y enviarlo a la API. Por último, hay una aplicación llamada "Ayuda" que muestra información sobre el proyecto y su documentación.

En cuanto a la estructura de la aplicación, esta sigue un patrón de diseño Modelo-Vista-Controlador (MVC). El modelo es la base de datos XML que se utiliza en la API. La vista se refiere a las templates y páginas web que se muestran al usuario en el frontend. Por último, el controlador es la lógica que se encarga de procesar las solicitudes y respuestas de la API.

En cuanto a las tecnologías utilizadas, Flask y Django son dos frameworks populares de Python para la creación de aplicaciones web. Flask es conocido por su simplicidad y su capacidad para crear aplicaciones web rápidamente, mientras que Django es más completo y ofrece una gran cantidad de herramientas para el desarrollo de aplicaciones complejas. Además, el uso de una base de datos XML es interesante ya que ofrece una alternativa a las bases de datos relacionales o NoSQL más tradicionales.

Conclusiones

El proyecto que has presentado es una aplicación web interesante y compleja que utiliza tecnologías de vanguardia como Flask y Django para ofrecer una API completa y un frontend atractivo y fácil de usar. El uso de una base de datos XML es

interesante y demuestra una alternativa viable a las bases de datos más tradicionales. En general, el proyecto parece estar bien estructurado y bien pensado, lo que sugiere que se ha puesto mucho esfuerzo en su diseño y desarrollo.

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

Python Software Foundation. (2023). The Python Standard Library. Retrieved from https://docs.python.org/3/library/index.html

Flask Documentation. (2023). Flask Documentation. Retrieved from https://flask.palletsprojects.com/en/2.2.x/

Django Software Foundation. (2023). Django Documentation. Retrieved from https://docs.djangoproject.com/en/4.2/