
PROYECTO 3 – FLASK DJANGO PYTHON

201612174 – Alberto Gabriel Reyes Ning

Resumen

Escribe un programa utilizando HTML+CSS+JS, DJANGO, FLASK y PYTHON que utilice un REST API. El programa consiste en una página web creado utilizando con HTML+CSS+JS y desplegado por DJANGO. FLASK se utiliza para crear el API, así la pagina web se puede comunicar con PYTHON.

Tiene operaciones de tres tipos que utiliza el GET/POST. La primera operación se manda los datos a FLASK utilizando un POST. El segundo y tercer operación se crea graficas utilizando datos de la entrada utilizando un GET. Tiene una operación adicional que limpia el programa de información y lo regresa al estado inicial.

Esta aplicación consiste en mandar datos utilizando un API y manipularlo en Python para mostrarlo en HTML. El programa fue escrito en Python 3.8.1 en Visual Studio Code utilizando Regex para leer los archivos de entrada. Se utiliza ElementTree para crear los XML de salida y guardar las informaciones en un archivo. Se utiliza xmltodict para cambiar el XML de salida a JSON para que se puede utilizar en el API como se funciona mejor con JSON.

Palabras clave

Full Stack, FLASK, DJANGO, API, JAVASCRIPT

Abstract

Write a program utilizing HTML+CSS+JS, DJANGO, FLASK and PYTHON that utilizes a REST API. The program consists of a webpage created with HTML+CSS+JS and deployed using DJANGO. Flask is utilized in the creation of the API as it lets the web page communicate with PYTHON.

There are three types of operations that utilize GET/POST. The first operation sends data to FLASK utilizing the POST method. The second and third operation creates graphs utilizing data from the XML file utilizing the GET method. There is an additional operation that cleans the program of information returning it to the initial state.

This program consists of sending data utilizing an API and manipulating the information in PYTHON so it can be displayed in HTML. This program was written in PYTHON 3.8.1 in Visual Studio Code utilizing Regex to read the entrance file.

ElementTree is used to create the XML files and save the information in a file. Xmltodict is used to change the XML file into a JSON object so that it can be readily accessed by the flask API.

Keywords

Full Stack, FLASK, DJANGO, API, JAVASCRIPT

Introducción

Crear un programa el cual se debe ser de página web con FLASK API y fácil de utilizar. EL programa debe ser de capaz de cargar un archivo XML en el HTML y procesarlo en PYTHON por el api de FLASK.

Dicha aplicación se recibe un archivo XML que contiene información sobre varios eventos donde reportan códigos de errores. Este archivo este procesado y guardado en un nuevo archivo XML que contiene las estadísticas de todos los errores reportados por fecha. En el lado del pagina web, se puede llamar esta información utilizando el GET método del FLASK API.

La aplicación se entregará al inicio del mes de mayo y se realizará en Visual Studio Code 1.54.1 en lenguajes Python 3.8.1, Django 3.2.2 y FLASK 1.1.2.

Desarrollo del tema

A. Funciones del Sistema

1. Cargar Archivo
2. Peticiones
 - 2.1 Consultar Datos
 - 2.2 Filtrar información por fecha y usuario que reporta
 - 2.3 Filtrar por fecha y código de error
3. Ayuda
 - 3.1 Datos del Estudiante
 - 3.2 Documentación

B. Explicación del Programa

El programa al iniciar se muestra a página web al usuario donde el usuario se puede cargar un archivo XML. Si el usuario no carga un archivo, las operaciones del programa no funcionaria.

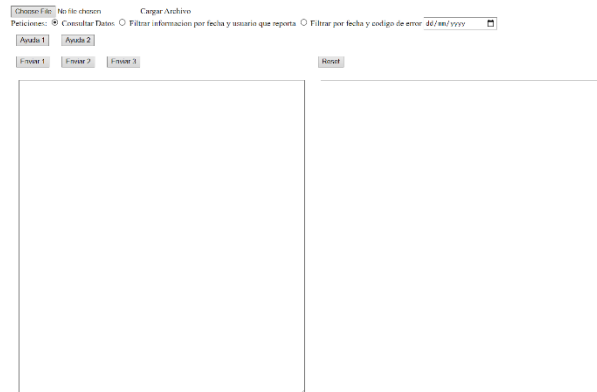


Figura 1. Página Principal

Fuente: elaboración propia

Después de cargar un archivo XML, el programa se mostrará él información en el primer área de texto.

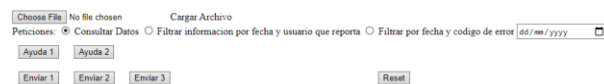


Figura 2. Operaciones en el HTML

Fuente: elaboración propia

Al presionar el botón de “Ayuda 1”, se mostrará información del estudiante en la segunda área de texto.

Al presionar el botón de “Ayuda 2”, se abre en otro pestana la documentación del Proyecto.

Al presionar el botón Enviar 1, se manda la información en la primera área de texto al FLASK API para procesar y guardarlo en memoria.

Al presionar el botón Enviar 2, se pide información del FLASK API según la fecha en la entrada de fecha, y muestra la cantidad de usuarios que reportaron en esa fecha.

Al presionar el botón Enviar 3, se pide información del FLASK API según la fecha en la entrada de fecha, y muestra la cantidad de errores que reportaron en esa fecha.

Al presionar el botón “Reset”, se reinicia el programa y la regresa al estado inicial.

C. Atributos del Sistema

- Aplicación desarrollada en Visual Studio Code
- Cargar Archivos XML y leerlo con Regex
- Crear Archivos XML utilizando ElementTree
- Cambiar Archivos XML a JSON utilizando xmldict
- Programación Orientado a Objetos
- Utilización de Listas Dinámicas (Listas Simples)

D. TDA Utilizado

Lista Simple

Una lista simple es una lista linear donde los elementos no están guardados en ubicaciones de memoria contiguas. Estas listas son dinámicas debido que se guarda los elementos en memoria utilizando apuntes donde cada nodo se apunta al siguiente nodo o a None. Listas simples son utilizados por su eficiencia en agregando y borrando información cuando él información se agrega/borra al inicio/fin de la lista.

Conclusiones

El programa podría estar escrito utilizando solo FLASK para desplegar los HTML y crear el API. En este problema si funciona utilizar DJANGO y FLASK juntos, pero para problemas mas complejos, seria mejor utilizar solo un framework.

Referencias bibliográficas

Howson, S. (2018, 6 marzo). *Python XML with ElementTree: Beginner's Guide*.
www.datacamp.com.
<https://www.datacamp.com/community/tutorials/python-xml-elementtree>

Graphical User Interfaces with Tk — Python 3.9.2 documentation. (s. f.).
<https://docs.python.org>. Recuperado 21 de marzo de 2021, de
<https://docs.python.org/3/library/tk.html>

Quickstart — Flask Documentation (1.1.x). (s. f.).
Flask. Recuperado 6 de mayo de 2021, de
<https://flask.palletsprojects.com/en/1.1.x/quickstart/>