
PROYECTO 2-CHET

201602834 – Anderson Danilo García Alvizures

Resumen

La modernización para el manejo de datos es constante por lo cual una de las opciones son aplicaciones web, el buen manejo de Back-end y Front-end proporciona una optimización para el manejo de datos y la presentación de los mismos. Desarrollado en lenguaje Python, utilizando dos frameworks

Las dos herramientas utilizadas son Flask para el Back-end y Django para el Front-end, donde se busca optimizar archivos csv convirtiéndolos a XML y también mostrar a través de estos mostrar las gráficas necesarias para una mejor interpretación. La verificación de los csv es fundamental para evitar que existan datos mal colocados en otros campos para ello se valida por medio de expresiones regulares.

Palabras clave

Frameworks
Flask
Django
API

Abstract

The modernization for data management is constant, for which one of the options is web applications, the good management of Back-end and Front-end provides an optimization for data management and its presentation. Developed in Python language, using two frameworks

The two tools used are Flask for the Back-end and Django for the Front-end, where it is sought to optimize csv files by converting them to XML and also to show through these the necessary Graphics for a better interpretation. The verification of the csv is essential to avoid misplaced data in other fields, for this it is validated by means of regular expressions.

Keywords

Frameworks
Flask
Django
API

Introducción

Por medio de dos frameworks escritos en Python (Flask, Django), donde se optimiza archivos csv verificándolos por medio de expresiones regulares en cada uno de los campos del archivo y para visualizar todo en un archivo XML y de esa forma no tener archivos distintos para el manejo de la cartera de clientes y ventas, por medio de Django se optimiza la visualización en una página web amigable al usuario para la carga de los archivos csv y la visualización en XML.

Desarrollo del tema

El envío y recepción de datos se realiza por medio de Flask, creando una API en el cual se recibe un archivo XML editado en Django el cual por medios del método Get, y reenvía los datos necesarios por medio del método Post, para que en el front se pueda realizar las gráficas.

La verificación de los archivos se realiza en Django, donde por medio de expresiones regulares para validar que los campos tengan los datos adecuados y así poder crear el archivo XML y enviarlo la API para el procesamiento y devolución de datos

Las herramientas de desarrollo fueron las siguientes

a. Flask

El marco Flask se volvió tremendamente popular como una alternativa a los proyectos de Django con su estructura y dependencias monolíticas. El éxito de Flask generó mucho trabajo adicional en tickets de emisión y solicitudes de extracción.

Inicialización: las aplicaciones de matraces deben crear una instancia de aplicación. El servidor web pasa todas las solicitudes que recibe de los clientes a los objetos para su manejo utilizando un protocolo para WSG de flask.

b. Django

Siendo de código abierto, escrito en python fomenta el desarrollo rápido y diseño de una aplicación desde el concepto hasta la funcionalidad, permitiendo crear desde sistemas de gestión de contenido hasta redes sociales y plataformas informáticas científicas. Cumpliendo así con los estrictos requerimientos de seguridad web.

c. API

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

d. Funcionamiento

Los archivos csv se cargan en la parte de front end para que Django realice el procesamiento de los mismos y se devuelva un archivo XML con los datos de los cuatro archivos csv, en caso de que tengan errores solo se muestre un pequeño listado de errores para su corrección en los archivos csv.

Las peticiones muestran una mejor visualización de la cartera de clientes, donde se muestran gráficas de mejores clientes, juegos más vendidos, cumpleaños y listados de juegos.

Conclusiones

En el desarrollo de la aplicación web se comprobó la simplificación del manejo de datos por medio de Flask y la optimización en Django para la recepción de datos de del servidor para luego crear y mostrar los datos en un archivo XML.

Por medio de diferentes frameworks para el HTML para una mejor visualización de los gráficos y estilo de la página.

La API por medio de Flask permite crear una comunicación entre el back end y el front end esto evita que se usen terceras aplicaciones para eso.

Referencias bibliográficas

<https://www.fullstackpython.com/flask.html>

<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

Extensión: de cuatro a siete páginas como máximo

Adicionalmente, se pueden agregar apéndices con modelos, tablas, etc. Que complementan el contenido del trabajo.