
PROYECTO 3

201906588 – Byron Estuardo Solís González

Resumen

Se desarrolló una aplicación capaz de leer el contenido de un archivo con extensión XML, analizar y catalogar su contenido, además de poder generar un archivo de salida con un reporte de los eventos registrados en el archivo de entrada. La aplicación también es capaz de obtener cierta información del reporte de una forma más específica.

Para su creación se utilizó el lenguaje de programación Python y la herramienta basada en este Django. El desarrollo se dividió en dos etapas, la creación de una API REST a través de Python y el desarrollo del Frontend de la aplicación a través de Django. Para la primera se hizo uso de expresiones regulares para poder obtener información específica de las cadenas de texto proporcionadas por la entrada y programación orientada a objetos para organizar los datos obtenidos; Para la comunicación entre estos dos componentes se utilizó el protocolo HTTP.

Como resultado se obtuvo una aplicación sencilla que facilita la búsqueda, extracción y catalogación de la información de un archivo de extensión XML.

Palabras clave

HTTP Django API XML Regex

Abstract

An application was developed capable of reading the content of a file with an XML extension, analyzing and cataloging its content, as well as being able to generate an output file with a report of the events recorded in the input file. The application is also capable of obtaining certain information from the report in a more specific way.

For its creation, the Python programming language and the tool based on this Django were used. The development was divided into two stages, the creation of a REST API through Python and the development of the Frontend of the application through Django. For the first, regular expressions were used to obtain specific information from the text strings provided by the input and object-oriented programming to organize the data obtained; The HTTP protocol was used for communication between these two components.

As a result, a simple application was obtained that facilitates the search, extraction and cataloging of information from an XML file extension.

Keywords

HTTP Django API XML Regex

Introducción

El leer, analizar y catalogar grandes cantidades de información de forma manual es poco rentable, por ello el desarrollar aplicaciones que automaticen estas acciones es de gran valor, tanto para la efectividad como para la eficiencia en el manejo de la información.

Para el desarrollo de la aplicación se partió de esto, buscando dar una alternativa sencilla que utilizara como materia prima del análisis archivos de extensión XML. Para que el consumo de esta aplicación como servicio fuera posible, y a un futuro un servicio por internet, se optó por dividir el desarrollo de la aplicación en Backend y Frontend.

Para el Backend se creó una API REST que utiliza el protocolo HTTP para su comunicación con otras aplicaciones, permitiendo que el servicio que presta sea consumido por otras aplicaciones y no solamente por el Frontend de esta.

Desarrollo del tema

Funcionalidades:

1. **Cargar archivo:** Permite ingresar un archivo con extensión XML para cargarlo a la aplicación y a partir de este generar las siguientes funcionalidades
2. **Creación de estadísticas:** Toma como base los datos del archivo leído anteriormente, analiza su contenido y genera un reporte de los eventos registrados por fecha.
3. **Filtrar información por fecha y usuario que reporta:** Al seleccionar esta opción el usuario puede elegir la fecha por la cual se requiere filtrar y se le devuelven los correos de los usuarios que reportaron errores en esa fecha

4. **Filtrar por fecha y código de error:** Al seleccionar esta opción el usuario puede ingresar un código de error y se le devuelve el total de mensajes que contienen ese código por cada fecha en donde se hizo el reporte de dicho error.

Lectura de archivos XML:

Para la extracción de la información se utilizó la librería de Python llamada ElementTree, para el análisis de la info obtenida se hizo uso de la división de las cadenas de entrada utilizando como referencia las palabras propias de la estructura definida para el archivo de entrada, separando así la información de los distintos campos, para luego extraer de estos la información a través de expresiones regulares.

- a. **Expresiones regulares:** Para la generación de estas se utilizó la librería de Python llamada Re, que permite crear y personalizar expresiones regulares adaptables a los requerimientos del proyecto. A continuación se presentan las expresiones utilizadas en la aplicación:
 - i. Fecha: [0-3][0-9]/[0-1][0-9]/[0-9][0-9][0-9][0-9]
 - ii. Correo: [\w]+@[\w]*[. \w]+
 - iii. Código de error: [0-9]+
 - iv. Descripción: Para esta opción de optó por extraer todas las palabras por separado y luego unir las en una cadena
 - Palabra : [0-9]+

Desarrollo de la Api

Para el desarrollo de esta se hizo a través de Python, utilizando para su comunicación el protocolo HTTP

Conclusiones

La utilización de expresiones regulares para el análisis de información es de gran utilidad, ya que reducen el tiempo que se le dedica a esta tarea además de aumentar la eficiencia de esta.

El desarrollar toda la lógica de una aplicación en una API REST mejora la el orden del código, además de tener todas las funciones agrupadas en un mismo servicio

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.