

Entrega 4

Desarrollo de una Web API utilizando MongoDB y Flask

Fecha de Entrega: 17 de noviembre de 2017

Esta entrega busca conectar la aplicación que han desarrollado hasta ahora con una API desarrollada por ustedes. Esta API deberá ser desarrollada con el *framework* Flask, y su base de datos almacenará información utilizando MongoDB. En concreto, en esta entrega deberán:

- Importar datos a MongoDB.
- Extender un código base de una aplicación en Flask para hacer las consultas deseadas y retornar los JSON respectivos.
- Hacer búsqueda por texto en MongoDB.
- Visualizar resultados de las consultas.
- Montar todo el trabajo en un servidor.
- Consumir los JSON en la aplicación web en PHP.

Detalles Académicos

Para esta entrega cada jefe de grupo recibirá los datos de un servidor ubicado en el DCC. En este servidor deben hacer el traspaso a un entorno de producción de todo su trabajo. Se recomienda que para trabajar, comiencen la aplicación en su computador personal y la ejecuten en *localhost*. Una vez que comprueben su funcionamiento la envíen al entorno de producción. Además, existe una aplicación base. Los grupos pueden extender esta aplicación según sus necesidades.

Introducción

Justo cuando las visitas a su aplicación estaban bajando drásticamente, se produjo una filtración de mensajes enviados por muchas celebridades del mundo de la música. Usted quiso aprovechar el momento para que su aplicación sirva de interfaz para navegar por estos mensajes.

Descripción de tareas a realizar

Desarrollo de una Web API

En primera instancia, debe realizar una Web API que exponga en formato JSON los mensajes. La API debería contar con al menos las siguientes rutas para trabajar sobre los mensajes:

- Una ruta que al recibir el `id` de un mensaje, obtenga toda la información asociada a ese mensaje.
- Una ruta que al recibir el `id` de un artista, obtenga toda la información de ese artista, junto con los mensajes emitidos por él.
- Una ruta que al recibir el `id` de dos artistas, obtenga todos los mensajes intercambiados entre esos artistas.

Además la API debería estar preparada para hacer filtros por fecha y ubicación de mensajes.

Búsqueda por texto: la API debe poder permitir obtener mensajes mediante búsqueda por texto. Se debe poder consultar por:

- Agregar una o más frases que sí o sí deben estar en el mensaje.
- Agregar una o más palabras que deseablemente deben estar, pero no son necesarias.
- Agregar un conjunto de palabras que no pueden estar en el mensaje.

Esta API debe ser subida a producción¹ en el servidor destinado para eso, con el fin de ser consumida por la aplicación en PHP. Esta API no debe estar en el servidor `bases.ing.puc.cl`, sino que se debe subir al servidor designado para esto.

¹Para hacer deploy, se sugiere revisar el siguiente tutorial <https://www.digitalocean.com/community/tutorials/how-to-deploy-a-flask-application-on-an-ubuntu-vps>.

Consumir la API desde la aplicación PHP

Una vez construida la API, debe ser posible consumirla desde la aplicación en PHP. Debe haber uno o más formularios que permitan:

- Buscar un mensaje o un usuario por `id`.
- Desplegar todos los mensajes por un usuario.
- Desplegar todos los mensajes intercambiados por dos usuarios².
- Buscar mensajes por texto utilizando las funcionalidades de la búsqueda por texto señaladas anteriormente.

Además se debe poder ingresar un usuario, y un rango entre dos fechas. Para cada mensaje emitido por el usuario entre esas dos fechas, se debe visualizar en un mapa los lugares en que ha estado ese usuario (según la geolocalización de los mensajes). Para esto puede usar cualquier librería que permita manejar mapas en entornos web. Un ejemplo es **Leaflet**³.

Importante: Toda esta parte del trabajo se realiza en `bases.ing.puc.cl`.

- Para facilitar la corrección se solicita indicar a los ayudantes en un **readme** toda la información necesaria para comprender de manera más rápida cualquier aspecto de la entrega. Además el código de su aplicación debe ser subido en una carpeta comprimida a `bases.ing.puc.cl`. **Este debe estar en su carpeta de grupo de las entregas anteriores**. Además, los detalles de cómo manejar su API deben estar documentados en una página en `bases.ing.puc.cl/grupoXX/entrega4/api.html`.

²Para buscar usuarios en este punto y el anterior, en primera instancia se puede hacer por `id`, sin embargo se debería contar con un menú que permita buscar por nombre de artista.

³<http://leafletjs.com/>