



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
2º semestre 2016

Actividad 15

Web Services + Regex

Instrucciones

Como con toda relación en la vida, durante este largo y duro semestre se han formado relaciones de amor y odio entre alumnos y ayudantes. En esto no hay nada de malo, salvo que Bahvrakis, descubrió que era odiado por TODOS los alumnos. Sumido en su odio, utilizó su valioso tiempo en la creación de una serie de obstáculos para acceder a esta información, las que solo podrían ser superadas por un gran programador. Esto evitaría que los alumnos puedan utilizar estas relaciones a su favor. Es así como tú decides poner tu mayor esfuerzo en averiguar cuales son estas relaciones y de esta manera saber con que ayudante tienes más afinidad para solicitar una corrección al final del ramo (en caso de ser necesario.¹).

Los obstáculos son en una serie de etapas para proteger la clave de acceso a la base de datos. Cada usuario tiene una única clave y para acceder a ella debe realizar lo siguiente:

- Paso 1: Realizar una request tipo GET a la página <http://ac15-aaossa.rhcloud.com/inicio>. Esto retornará un string y una nueva URL.
- Paso 2: Utilizar Regex para concatenar strings que cumplan condición 1 definida en la sección **Condiciones** (Ver página siguiente).
- Paso 3: Realizar una request tipo POST a la URL http://ac15-aaossa.rhcloud.com/nueva_url, entregando el string resultante del paso 2.
- Paso 4: Repetir pasos (1, 2, 3) 3 veces con las nuevas URL obtenidas y condiciones respectivas.
- Paso 5: En el último ciclo de repetición de pasos, obtendrás la clave de la base de datos.

Para realizar las instrucciones anteriores, revisa el archivo `API.md`, que describe detalladamente los parámetros y los retornos de cada *request*.

¹Sinceramente esperamos que esto no sea necesario, pero siempre termina pasando

Condiciones

Las condiciones que debes verificar en cada etapa son las siguientes:

1. Solo se deben aceptar correos electrónicos de dirección @gmail.com, @gmail.arg, @gmail.cn, @gmail.eu, @gmail.cl, donde el nombre de la dirección (nombre@correo.cl) debe tener SOLO UN '.', ningún '-' y debe estar compuesto solamente por letras (sin mayúsculas ni dígitos).
2. Solo se aceptan palabras similares a *bastián*, que inicien con b/B y terminen con n/N. Pueden faltar letras entremedio, pero no existir letras no correspondientes a la palabra. Ej: baaaaassstian, BAS-TiiiiAAnnnnnnN, Basian.
3. La palabra debe iniciar con 'el+' seguido de la palabra 'terrible', 'odiado' o 'amado?' sin mayúsculas. Luego puede contener o no la palabra 'jefe', 'boss'(en mayúsculas o minúsculas). Por ultimo, viene 'x1000' donde puede existir una cantidad indefinida de 0, donde incluso puede no haber un 0. Ej: 'el+terriblex1000000', 'el+odiadoBOsSx1'.
4. La palabra *tarealograda* debe estar contenida dentro del string (en minúsculas y exacta).

Direcciones

Una vez obtenida tu clave, tendrás acceso a la base de datos. Para extraer datos tienes a tu disposición los recursos descritos a continuación:

- <http://ac15-aaossa.rhcloud.com/ayudantes>: retorna diccionario de *nombre_github: id* para todos los ayudantes.
- <http://ac15-aaossa.rhcloud.com/alumnos>: retorna diccionario de *nombre_github: id* para todos los alumnos.
- http://ac15-aaossa.rhcloud.com/ayudantes/<id_ayudante_x>: retorna lista de todas las relaciones del *ayudante_x*. El elemento y de la lista corresponderá a la relación del ayudante x con el alumno y.
- http://ac15-aaossa.rhcloud.com/alumnos/<id_alumno_x>: retorna lista de todas las relaciones del *alumno_x*. El elemento y de la lista corresponderá a la relación del alumno x con el ayudante y.

Para acceder a estos datos, tendrás que hacer un request de tipo POST en la dirección correspondiente, de la manera especificada en *API.md*.

Así, teniendo libre acceso a la base de datos, puedes obtener los ids de los ayudantes y alumnos para luego hacer un request tipo POST a */ayudantes/id_ayudante* y recibir la lista de sentimiento del ayudante de id *id_ayudante*. Cada índice de la lista de sentimientos del ayudante representa al alumno de *id_alumno* igual al índice correspondiente a la lista. Cada relación se verá representada por un 0 en caso de relación neutra, un 1 en caso de relación de amor y un 2 en caso de relación de odio. Las listas de sentimientos de los alumnos funcionan de forma análoga a la de ayudantes.

Consultas

Con los datos disponibles, decides averiguar la siguiente información:

- Número de alumnos amados por cada ayudante que te ama. Debes mostrar en consola: nombre_ayudante, número de alumnos. Finalmente, debes imprimir: "Debo ir a recorrer con nombre_ayudante", donde el nombre de ayudante corresponde al ayudante que te ama y que a su vez ame a menos personas.
- Ayudante más Odiado (No incluir a Bahvrakis que los odia a todos).
- Ayudante más tierno (el ayudante que más ama).

Restricciones

Está estrictamente **prohibido** generar una matriz de todas las relaciones y almacenarla en memoria. Los datos utilizados deben ser requeridos al servidor solo para ser utilizados inmediatamente.

Notas

- La base de datos es totalmente ficticia y no representa la realidad (menos lo de Bahvrakis)
- Puedes encontrar más detalles de cada dirección de la API, sus parámetros y contenidos retornados en el archivo `API.md`.

To - DO

- (2.00 pts) Correcta implementación de métodos GET y POST.
- (2.00 pts) Realizar filtrado de listas de palabras utilizando Regex.
- (2.00 pts) Correcto resultado de cada consulta.

Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC15
- **Hora:** 16:55