

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

IC-4700 Lenguajes de Programación

Tarea Programada 2: Aplicación de Recetas

Prof. Andréi Fuentes L.

Estudiantes:

Susana Cob García 2017136954

Alejandro Tapia Álvarez 2017116952

07/06/19

Tabla de contenido

Descripción del programa.....	3
Diseño del programa	3
Restricciones en la aplicación	5
Referencias Bibliográficas	5

I. Descripción del programa

Este proyecto consiste en crear una aplicación móvil que se comunice con una API y la API se comunice con una base de conocimientos escrita en Prolog. La aplicación móvil puede realizar consultas a la API de datos almacenados en la base de conocimientos o puede insertar datos en la base de conocimientos (siempre a través de la API).

La aplicación debe ser dirigida a usuarios Android y debe tener las siguientes funcionalidades:

- Crear cuenta
- Iniciar/Cerrar Sesión
- Ver recetas en una lista
- Mostrar la información de una receta (nombre, tipo, ingredientes, instrucciones y fotos).
- Agregar nueva receta (nombre, tipo, ingredientes, instrucciones y fotos).

La API debe ser escrita en Ruby o Python y debe correr sobre AWS o Heroku (en nuestro caso, utilizamos Python y Heroku). Se deben manejar tokens como método de autenticación. La API debe tener los siguientes endpoints:

- Crear cuenta
- Iniciar Sesión
- Obtener lista de recetas
- Obtener información de una receta
- Agregar una receta
- Buscar recetas por nombre, por tipo o por ingrediente.

La base de conocimientos debe ser implementada en Prolog y esta almacena las recetas. Los usuarios se almacenan con SQL (PostgreSQL).

II. Diseño del programa

Para crear el API al que la aplicación se va a conectar, utilizamos la librería de Python "Flask". Esta es una librería que facilita la creación de aplicaciones con múltiples "rutas" para ejecutar diferentes procedimientos dependiendo de la ruta y que los que llamen a esas rutas puedan obtener resultados en formato JSON.

Tokens

Para la creación de tokens se usa la librería JWT, la cual permite cifrar un diccionario con datos como el usuario, la fecha actual y el tiempo que el token se mantendrá activo. Esta librería también lo guarda hasta que se termine el periodo de actividad y permite convertir ese cifrado a texto en UTF-8 para que las aplicaciones lo puedan usar.

Cifrado de contraseñas

Para el cifrado de contraseñas, se utiliza la librería “cryptography”, que permite cifrar texto con una llave dada al principio del código.

Corrimiento del API

El API corre en Heroku, lo cual facilitó el uso de una base de datos y la configuración sencilla de los permisos para AWS, no obstante, trajo ciertos problemas con respecto al uso de Prolog.

Base de datos de usuarios

Para la base de datos, Heroku facilita el uso de bases de datos, implementada con PostgreSQL, e incluso hace la conexión entre el API y la base de datos relativamente sencilla. Para hacer las llamadas a la base de datos se usa la librería “psycopg2”, la cual es usada en la página de Heroku para ejemplos de conexión a bases de datos. Esta base de datos se llama cada vez que se registra un usuario para guardar sus datos o cada vez que se ejecuta un Log In, para comprobar que el usuario exista y que la contraseña corresponda con el usuario dado.

Base de conocimientos

La base de conocimientos se guarda en el servicio S3 de AWS, y la obtiene la aplicación por medio de la librería “boto3”, una librería oficial de Amazon para conectarse a S3. A la hora de ejecutar procedimientos en Prolog, nos dimos cuenta de que Heroku no permitía la instalación de un lenguaje de programación que no fuera el que se está usando para la aplicación, por lo que comenzamos a utilizar los servicios de Azure para correr una máquina virtual en línea, y conectarnos a ella a través de la librería “paramiko” para correr métodos escritos en Python y realizar consultas en Prolog. Estas consultas se realizan con la librería “pyswip”.

Paramiko

Paramiko funciona conectándose al servidor dado por medio de la dirección de IP público del servidor, un usuario y una contraseña, y una vez establecida la conexión, llama a la terminal del servidor. Por esta razón, resulta casi imposible enviar comandos con caracteres reservados de una terminal de Linux. Para el “!” que se encuentra en la base de conocimientos, se sustituye por su valor en hexadecimal \x21 y se pueda realizarse la operación sin problemas. Esto también implica que en las recetas no pueden ir muchos de los caracteres reservados en terminal. También se elimina cualquier espacio, ya que, a la hora de llamar a Python en la terminal, cada espacio que exista lo tomará como un argumento nuevo.

Pyswip

Pyswip es una librería que permite crear su propia base de conocimientos dentro de Python, y realizar consultas con SWI-Prolog. Algo a tomar en cuenta con esta librería es que los hechos (o cualquier línea que se inserte en general) se deben insertar uno por uno sin puntos al final ni saltos de línea, por lo que antes de insertar la base de conocimientos a Pyswip se

realiza un “split” del string que recibe, para que se divida por cada punto y salto de línea, y cada elemento de la lista creada sea una línea a insertar.

Android Studio

Respecto a Android Studio, no hubo muchos problemas, ya que hemos trabajado en esta plataforma anteriormente. Para comunicar la aplicación con la API se utiliza OKHTTP.

III. Restricciones en la aplicación

A la hora de usar la aplicación se deben tomar en cuenta los siguientes detalles:

- No se permiten caracteres especiales para cualquier input del usuario (es decir, cualquier carácter diferente a letras, números o espacios), excepto el @ a la hora de registrarse/iniciar sesión con el correo.
- Se deben tomar en cuenta los espacios y mayúsculas a la hora de agregar/buscar una receta. Por ejemplo: Si la receta se registró como ‘Arroz con Pollo ’, la receta será guardada en la base de conocimientos como ‘arroz_con_Pollo_’. Por lo tanto, si, más tarde, el usuario desea buscar esa receta, debe buscarla como ‘arroz con Pollo ’ o ‘Arroz con Pollo ’ (con el espacio al final y las mayúsculas en letras diferentes a la primera). Las mayúsculas no se cambian (excepto la primera letra, por Prolog) para mantenerlas a como el usuario decidió registrarlas.
- La primera consulta que realiza la aplicación a la API es posible que dure un poco más de lo esperado (por lo que, las consultas tienen un Time Out de 40 segundos, a pesar de que, aproximadamente, lo que dura la primera consulta son unos 15 segundos).
- El nombre de las recetas, el tipo y los ingredientes (alguno de los criterios de búsqueda) no pueden empezar con números.
- Los ingredientes y las instrucciones están en un área de texto, por lo que se separan con ‘\n’ (el ‘enter’).
- Tome en cuenta las repeticiones de ingredientes en una misma receta, la aplicación no las valida.

IV. Referencias Bibliográficas

1. <https://stackoverflow.com/questions/15380457/how-to-get-selected-row-object-from-table-in-android>
2. <https://stackoverflow.com/questions/18207470/adding-table-rows-dynamically-in-android>
3. <https://stackoverflow.com/questions/4854492/setting-width-to-wrap-content-for-textview-through-code>
4. <https://stackoverflow.com/questions/28007921/overlaying-the-action-bar-not-working>

5. <https://stackoverflow.com/questions/4854492/setting-width-to-wrap-content-for-textview-through-code>
6. <https://stackoverflow.com/questions/11348703/select-a-table-row-within-a-table-layout-dynamically>
7. <https://viralpatel.net/blogs/pick-image-from-gallery-android-app/>
8. <https://github.com/krunalpatel3/Simple-Camera-Save-Image-app>
9. <https://stackoverflow.com/questions/11686692/android-fit-image-to-fill-whole-layout>
10. <https://www.viralandroid.com/2015/11/load-image-from-url-internet-in-android.html>
11. <https://es.stackoverflow.com/questions/13521/error-android-os-networkonmainthreadexception-en-usar-httpurlconnection-de-andro>
12. https://www.tutorialspoint.com/android/android_json_parser.htm
13. <https://www.designstring.com/upload-images-to-the-aws-from-android/>
14. <https://stackoverflow.com/questions/12585492/how-to-test-if-a-json-object-is-null-or-doesnt-exist>
15. <https://stackoverflow.com/questions/9671546/asynctask-android-example>
16. <https://trinitytuts.com/get-and-post-request-using-okhttp-in-android-application/>
17. <https://stackoverflow.com/questions/3913592/start-an-activity-with-a-parameter>
18. <https://github.com/Dchengg/Progra2>
19. *OKHTTP Documentation*. <https://square.github.io/okhttp/>
20. <https://stackoverflow.com/questions/6858051/how-to-clear-the-previous-contents-of-the-table-layout/6858109>
21. <https://stackoverflow.com/questions/4841952/convert-arraylistmycustomclass-to-jsonarray>
22. *How do I handle HTTP errors like 401, 403, 503 etc.* <https://github.com/ribot/ribot-app-android/issues/4>
23. <https://stackoverflow.com/questions/21962502/convert-from-jsonarray-to-arraylistcustomobject-android/21980398>
24. <https://stackoverflow.com/questions/29456534/handling-authentication-in-okhttp>
25. <https://stackoverflow.com/questions/3904579/how-to-capitalise-the-first-letter-of-a-string-in-java>
26. <http://www.java2s.com/Code/JavaAPI/java.lang/ThreadJoinUsingJoinToWaitForThreadToFinish.htm>
27. <https://stackoverflow.com/questions/36455656/java-net-sockettimeoutexception-timeout>
28. <https://stackoverflow.com/questions/4052840/most-efficient-way-to-make-the-first-character-of-a-string-lower-case>
29. <https://stackoverflow.com/questions/13386107/how-to-remove-single-character-from-a-string/13386289>
30. <https://www.quora.com/How-can-a-string-be-checked-for-the-presence-of-special-characters-like-and-in-Java>
31. <https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask>

32. *Authenticating a Flask API Using JSON Web Tokens.*
<https://www.youtube.com/watch?v=J5bIPtEbS0Q&t=858s>
33. <https://devcenter.heroku.com/categories/postgres-data-transfer-preservation>
34. <https://devcenter.heroku.com/articles/heroku-postgresql#connecting-in-python>
35. <https://stackoverflow.com/questions/15237366/how-to-execute-a-sql-script-on-heroku>
36. <https://www.mssqltips.com/sqlservertip/5173/encrypting-passwords-for-use-with-python-and-sql-server/>
37. https://www.linuxtopia.org/online_books/database_guides/Practical_PostgreSQL_database/PostgreSQL_x20238_004.htm
38. <https://carto.com/help/working-with-data/sql-stored-procedures/#functions-that-doesnt-return-any-row>
39. https://www.w3schools.com/js/js_json_arrays.asp
40. <https://stackoverflow.com/questions/10434599/how-to-get-data-received-in-flask-request>
41. <https://pypi.org/project/pyswip/>
42. <https://www.pythoniza.me/paramiko-implementacion-python-de-ssh/>
43. <https://stackoverflow.com/questions/11551043/paramiko-ssh-connect-what-arguments-to-send>
44. <https://realpython.com/python-boto3-aws-s3/#downloading-a-file>
45. <https://stackoverflow.com/questions/40336918/how-to-write-a-file-or-data-to-an-s3-object-using-boto3>
46. <https://devcenter.heroku.com/articles/s3>
47. <https://devcenter.heroku.com/categories/postgres-basics>
48. <https://codehandbook.org/working-with-json-in-python-flask/>
49. <https://pynative.com/python-postgresql-select-data-from-table/>
50. <https://devcenter.heroku.com/articles/getting-started-with-python>
51. <https://docs.microsoft.com/en-us/azure/virtual-machines/linux/quick-create-portal>