



DOCUMENTACIÓN DE DISEÑO DEL WEB-BOT

Proyecto I



Ingeniería Informática

Servicios Web

1C2017-166001G1

Diana Cecilia Camacho Mendoza

Profesor:

Maikol Arguedas Valerio

Marzo 2017

Contenido

HERRAMIENTAS	3
POSTMAN	3
FLASK.....	3
JSON.....	3
PYMONGO	4
LOG	4
RAM	4
PYTHON	5
FUNCIONALIDADES.....	5
INFORMACIÓN GENERAL.....	5
FUNCIONES POR DEFECTO	5
API GOOGLE CALENDAR	5
APRENDER	5
MOSTRAR LOG Y RAM.....	6
OLVIDAR	6
BIBLIOGRAFÍA.....	7

Ilustraciones y Tablas

Ilustración 1: Diagrama funcionamiento	2
Tabla 1: CRUD en PyMongo.....	4

Web-Bot posee API REST y atiende peticiones de un cliente (Postman). Con esta aplicación es posible ejecutar diversas tareas por medio del envío de código. Estas tareas quedan registradas de forma que luego se pueden consultar.

A continuación, se presenta un diagrama que explica de forma general, el funcionamiento de este Web-Bot.



Ilustración 1: Diagrama funcionamiento

HERRAMIENTAS

POSTMAN

Para esta aplicación se utiliza como cliente a Postman, el cual es un cliente HTTP. Con él se envían las peticiones a la aplicación Web-Bot. Dependiendo las funciones, hay rutas específicas para hacer uso de las funcionalidades.

FLASK

Flask es un microframework de Python, su implementación resulta sencilla, y existe bastante documentación.

En este proyecto se utilizan dos métodos de HTTP: GET y POST con peticiones en formato JSON. A pesar de que ambos realizan request y response, se deben asignar tareas que correspondan a sus funciones respectivas.

GET: Con este método se **obtienen datos del servidor**, para ello se envía una petición (*request*) y se recibe un *response* con los datos.

POST: Es para cuando se desea **enviar datos al servidor** y que éste los procese. Mediante el request es posible enviar parámetros al servidor, una vez procesados, se devuelve una respuesta con el resultado.

JSON

Para los request con formato JSON se debe importar el paquete JSON y así obtener todas las funcionalidades del mismo.

- Para obtener datos recibidos en formato JSON se usa `request.json['clave']`
- Para codificar (de tipos nativos de Python a JSON) se hace uso de `json.dumps(variable)`
- Para decodificar se hace uso de `json.loads(variable)`

PYMONGO

Es una base de datos NoSQL, en donde la base de datos no contiene tablas, sino, colecciones, no hay filas, pues son reemplazadas por documentos y la llave primaria es sustituida por un "Object_Id". En el caso de este proyecto, MongoDB resulta sencilla de implementar y amigable con las funcionalidades.

La funcionalidad de esta base de datos en el Web-Bot corresponde a el almacenamiento de un historial y el almacenamiento de los recursos para el funcionamiento de API.

LOG

El Log corresponde a un tipo de historial que contiene todas las acciones que se realizan en la aplicación. Se insertan los datos en una colección de la base de datos (acción, fecha, usuario), y se muestran cuando el cliente lo solicita.

RAM

La RAM corresponde a otra colección que contiene una lista con los códigos a ejecutar. Este código se recibe mediante la acción de "Aprender", una vez almacenado en esta colección, se ejecuta. Cuando se desea que el Web-Bot olvide acciones, se eliminar el código de la colección RAM.

Métodos Utilizados	
CREATE	INSERT
READ	FIND
UPDATE	UPDATE
DELETE	REMOVE

Tabla 1: CRUD en PyMongo

PYTHON

Para el desarrollo de esta aplicación, se utiliza el lenguaje de programación Python. Es multiparadigma (puede ser orientado a objetos e interpretativo). Es un lenguaje limpio, flexible, fácil de aprender y muy compatible, por lo que resulta bastante útil para el desarrollo de aplicaciones web.

FUNCIONALIDADES

INFORMACIÓN GENERAL

Contiene datos como el nombre del Web-Bot, fecha de creación, un identificador (mediante un hash generado a partir de la fecha), y el creador. El nombre puede ser asignado por el usuario.

FUNCIONES POR DEFECTO

Este Web-Bot por defecto sabe realizar operaciones básicas, como sumar y multiplicar “n” cantidad de números, y puede realizar restas y divisiones entre dos dígitos. Para cada una de estas operaciones existe un método encargado de resolverlas.

API GOOGLE CALENDAR

Haciendo uso de un API de Google, es posible acceder al calendario ligado a una cuenta Gmail. Primeramente, esta API solicita un permiso para acceder a los datos, y luego brinda los próximos 10 eventos.

APRENDER

Para esta funcionalidad se recibe un código. Este código llega en formato JSON desde Postman (junto con el nombre de la acción que realiza ese código), y es

procesado con Flask. Posteriormente, se guarda en una colección y se ejecuta. Como respuesta se envía un mensaje confirmando la acción aprendida.

Una vez aprendida la acción, se ingresa en Postman la ruta para acceder a esa nueva funcionalidad, con los respectivos parámetros y demás configuraciones dependiendo de la acción.

MOSTRAR LOG Y RAM

Con ayuda de la función “Find” se realiza la búsqueda de todo lo que se ha guardado en las colecciones (en el log de operaciones y en la RAM). Estas son escritas en un archivo de texto, y luego leídas para ser enviadas con respuesta al cliente.

Lo que está en el LOG no desaparece de ninguna forma, únicamente se puede eliminar lo que está almacenado en la RAM, con el fin de olvidar cómo se realiza alguna acción.

OLVIDAR

Haciendo uso de la función “remove” de PyMongo, se borra de la colección código que se aprendió previamente. Una vez eliminado, se puede decir que el Web-Bot ha olvidado como realizar dicha acción.

BIBLIOGRAFÍA

- Ardissone, J. (09 de febrero de 2011). *Métodos GET vs POST del HTTP*. Obtenido de [Blog]: <https://blog.micayael.com/2011/02/09/metodos-get-vs-post-del-http/>
- Flask. (s.f.). Obtenido de <http://flask.pocoo.org/>
- Google. (s.f.). *Google Calendar API*. Obtenido de <https://developers.google.com/google-apps/calendar/quickstart/python>
- Moya, R. (23 de marzo de 2015). *Python MongoDB Driver (pymongo)*. Obtenido de Jarroba[Blog]: <https://jarroba.com/python-mongodb-driver-pymongo-con-ejemplos/>
- Pérez Esteso, M. (s.f.). *Cómo utilizar JSON en Python*. Obtenido de Geeky Theory [Blog]: <https://geekytheory.com/como-utilizar-json-en-python/>
- Python. (s.f.). Obtenido de <https://www.python.org/>