WebHook (Chatbot-SmartCDMX)

Introducción

Nuestro Webhook permitirá obtener información sobre eventos que suceden en la plataforma de (Facebook-Messenger) a medida de que vallan sucediendo. Para ello configuramos una URL, la cual siempre se encuentra a la escucha de recibir información sobre un evento producido en el chatbot de Calidad del aire, el cual producirá una respuesta de acción cuando tenga lugar para entregar la respuesta.

Para el desarrollo del WebHook proponemos desarrollarlo en un proyecto de CodeIgniter ya que es un potente framework de PHP, diseñado para desarrolladores que necesitan un conjunto de herramientas simple y elegante para crear aplicaciones completas.

Diseño

Crearemos un nuevo controlador extra al controlador principal del proyecto de CodeIgniter, llamado BOT.php el cual va a extender del controlador nativo de CodeIgniter, CI_Controller, para poder implementar funciones relacionadas con los roles de usuario.

```
class Bot extends CI_Controller{

        public function __construc(){
                parent::__construct();
        }


}
```

Luego de crear nuestro controlador base el cual va a ser utilizado por nuestro WebHook, vamos a crear las variables y funciones necesarias para poder dar funcionalidad a nuestro chatbot.

Variables:

•private $token_facebook;
El token de facebook es una cadena opaca que identifica a nuestra aplicación o página, y nuestro WebHook ocupara para realizar llamadas a la API Graph.

•private $token_acceso;
El token de acceso es una cadena el cual nos servirá para poder conectar nuestra aplicación a nuestro WebHook, este token es configurado desde la aplicación de Facebook-Messenger y ningún otro cliente podrá conectarse a nuestro WebHook sin dicho token.

•private $url;
Esta url es la principal herramienta que permite a las aplicaciones en este caso nuestro chatbot leer y escribir en la gráfica social de Facebook. Todos nuestros SDK y productos interactúan de algún modo con la API Graph y las demás API son extensiones de esta. Por esta razón, es fundamental declarar esta url.

Funciones:

•response – Escuchar eventos y mandar respuesta.
-Verificar si el token de acceso coincide con el del WebHook. Existen dos tipos de opciones, si coincide sigue el flujo, sino no continua más.
-Obtener evento de un mensaje recibido en nuestro chatbot. A continuación un ejemplo de un mensaje mandado por el usuario final de facebook:


```
{
   "object": "page",
   "entry": [
      {
         "id": "177383299689103",
         "time": 1520529590905,
         "messaging": [
            {
               "sender": {
                  "id": "1166392116797450"
               },
               "recipient": {
                  "id": "177383299689103"
               },
               "timestamp": 1520529590602,
               "message": {
                  "mid": "mid.$cAAD7p-Am48JoN-WRSliBqBIxY6CB",
                  "seq": 183831,
                  "text": "Que la calidad del aire en la Magdalena contreras"
               }
            }
         ]
      }
   ]
```

}
-Obtener el json que se produjo cuando el usuario final envió un mensaje al chatbot de Calidad del aire, de dicho json se puede obtener el siguiente parametro:

Campo	Tipo	Descripción
sender	int	Identificador del cliente

-Evaluar el evento:
• message
•postback

conectar_Witai($mensaje)

Parámetros:
$mensaje – es el mensaje que el usuario envía a través de facebook al webhook

Inicia sesión cURL para poder enviar el mensaje del usuario a la aplicación en **Wit.ai**, manteniendo el formato esperado

```
curl \
 -H 'Authorization: Bearer VWOBMOKUVFWO5X3TMTDBBRW4FBHA6PEI' \
 'https://api.wit.ai/message?v=13/03/2018&q=cual%20es%20la%20calidad%20del%20aire'
```

Para el envió del curl se requiere de la autorización (Bearer) la cual es el token de acceso de la aplicación de **Wit** y la fecha actual del día en que se envía el mensaje y obtiene la respuesta del curl en la variable $response

evaluar_respuesta_wit($sender,$respuesta)

Parámetros:
$sender - Identidifador del cliente
$respuesta – Json que se obtiene de conectar_Witai( ) como respuesta de **Wit**

Recupera de $respuesta los intent que **Wit** encontró en el mensaje del usuario para verificar el caso en que el Webhook debe buscar; después obtiene los entities del mismo Json esto con el fin de tener mayor información y ser más exactos al obtener la información de la API

Ejemplo de Json de respuesta de **Wit**

```
{
    "_text":"cual es la calidad del aire en iztapalapa",
    "entities":{
        "e_calidadaire":[{
            "confidence":1,
            "value":"c\u00e1lida del aire",
            "type":"value"
        }],
        "location":[{
```

```
            "suggested":true,
            "confidence":0.93645,
            "value":"iztapalapa",
            "type":"value"
        }],
        "intent":[{
            "confidence":0.99958561908572,
            "value":"I_CalidadAire"
        }]
    },
    "msg_id":"0FCY34G1vvVRaoKjh"
}
```

Si en el caso de que no exista una ubicación en el mensaje se enviara CDMX como ubicación predefinida. Y en el caso de que $respuesta tenga algún error se enviara un texto de error

respuesta_mensaje($sender, $calidad)

Parámetros:

$sender - Identidifador del cliente

$calidad – información obtenida por la api como respuesta al mensaje del usuario en un Json

Recupera la respuesta establecida por la api de Calidad del Aire guardada en $mensaje_texto. En dicha variable se estructura un Json donde se concatenan los parámetros de la función

```
$mensaje_texto = '{
                  "messasing_type": "RESPONSE",
                    "recipient":{
                        "id":"'.$sender.'"
                    },
                    "message":{
                        "text": "'.$calidad.'",
                    }
                  }';
```

mandar_respuesta($mensajeData,$url)

Parámetros:

$mensajeData - Json estructurado como respuesta de evaluar_respuesta_wit( )

$url – variable para conexión con API Graph

Inicia sesión cURL para poder enviar la respuesta $mensajeData a facebook a travez de un post. Es indispensable usar $url dentro de la conexión curl (CURLOPT_URL) pues lleva el acceso a la gráfica social de Facebook y su respectivo token, sin este no se lograria realizar la conexión

mostrar_accion($accion,$url)

Parámetros:

$accion - Json estructurado para dar comienzo al bot
$url - variable para conexión con API Graph

Inicia sesión cURL para realizar una accion la cual la primer parte lleva un boton de inicio para poder comenzar una conversacion con el bot

Ejemplo de la variable $accion

```
{
    "recipient":{
        "id":"'.$sender.'"
    },
    "sender_action":"typing_on"
}
```

Una vez realizada esta accion se vuelve a llamar esta misma funcion paro ahora $accion le asignamos otro Json el cual re-estructura con un mensaje de vienvenida

Ejemplo de la variable $accion

```
{
    "messasing_type": "RESPONSE",
    "recipient":{
        "id":"'.$sender.'"
    },
    "message":{
        "text": "Hola, en que te puedo ayudar? :)",
    }
}
```