Bot de whatsapp para FAQ basico

¿Que es el bot framework SDK?

Es una colección de librerias, herramientas y servicios que le permiten crear, implementar, probar y desplegar robots inteligentes. Incluye un SDK modular y extensible para construir bots y conectarse a servicios de IA. Con este framework, los desarolladores pueden crear bots que utilicen lenguaje natural, responder preguntas, entre otras cosas.

Estos bots son generalmente implementados como aplicaciones web, alojados en Azure y utilizandos APIs para enviar y recibir mensajes.

I) Metas para este bot:

https://learn.microsoft.com/en-us/azure/bot-service/bot-service-design-principles?view=azure-bot-service-4.0

1) Preparar la API de whatsapp

Whatsapp provee su propia llave API si usted se registra en <https://developers.facebook.com/docs/whatsapp/cloud-api> . Va a obtener la llave API para poder utilizar whatsapp en pruebas, y tambien necesita usted adjuntar un numero de celular QUE NO ESTE ACTIVADO CON WHATSAPP. No necesita estar en plan, solo compre uno en telcel movistar etc. y utilice ese.

2) frameworks

Aqui empezaremos con botbuilder:

<https://www.npmjs.com/package/botbuilder>

Este framework requiere node.js como minimo. Yarn, typescript

Empecemos con installar yarn:

https://classic.yarnpkg.com/en/docs/usage

sudo npm install –global yarn

Crear dependencias de yarn

sudo yarn

ahora a construir los binarios

sudo yarn build

3) Entiendiendo cURL:

Whatsapp envia un conjunto de informacion independiente de que tipo de dispositivo, navegador, aplicacion, etc. se utilice:

curl 'https://graph.facebook.com/v22.0/579992435207524/messages' \

-H 'Content-Type: application/json' \

-H 'Authorization: Bearer EAATyUT8goNsBO5BZALbJt4dQ01AyKJoGNRdeNAgSetc5wiRgdw5go7l6IthwQIIeTnJ7hXevKMM05VJKF71KyxoWJI4ZAGizcnsK8MJrsAIcLsLaxa53E51CIL4Jd3XVHcObE7mSpL8UQ2JNdmAMUn9XdZBdDqLt7c8J26X7YJdHxV7CHLbjbAZCNsZBEX1e3UsCgrUHbrGFYFLLZC137HqGZCsPIIZD' \

-d '{

  // MENU PRINCIPAL

  "messaging\_product": "whatsapp",

  "recipient\_type": "individual",

  "to": "523541090470",

  "type": "interactive",

  "interactive": {

    "type": "list",

    "header": {

      "type": "text",

      "text": "Gracias por contactar a la CFE."

    },

    "body": {

      "text": "Haga click en en menu de opciones para elegir su consulta."

    },

    "action": {

      "button": "Menu de opciones",

      "sections": [

        {

          "title": "Menu Principal",

          "rows": [

            {

              // ROW CLEARED BALANCE

              "id": "ROW\_CLEAR\_BAL",

              "title": "1",

              "description": "Para Conocer saldo a favor, responder con 1."

            },

            {

              // ROW OUTSTANDING BALANCE

              "id": "ROW\_OUT\_BAL",

              "title": "2",

              "description": "Para Conocer saldo a pagar, responder con 2."

            },

            {

              // ROW DUE DATE

              "id": "ROW\_DUE\_DATE",

              "title": "3",

              "description":"Para conocer fecha limite; responder al bot con 3."

            },

            {

              // ROW CONTACT SUPPORT

              "id": "ROW\_SUPPORT",

              "title": "4",

              "description": "Si ninguna opcion satisface, contactar soporte ."

            }

          ]

        }

      ]

    }

  }

}'

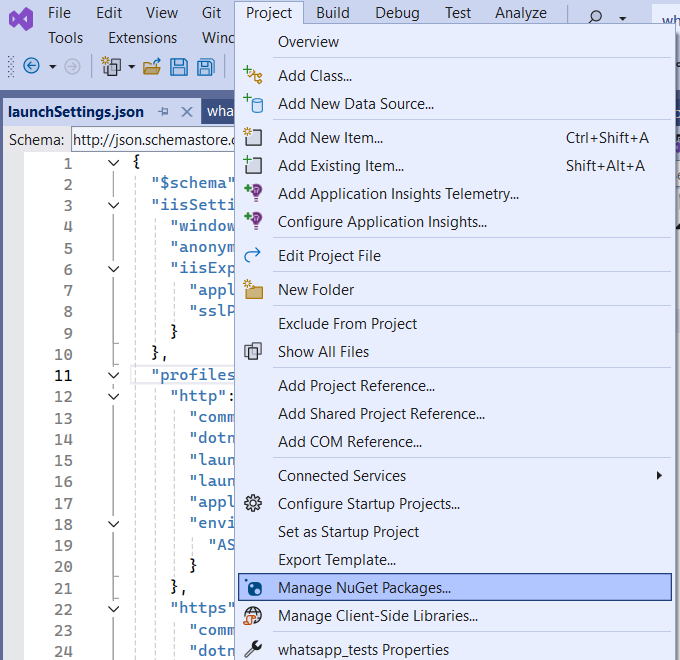
Esto es el mensaje que whatsapp va a enviar al telefono movil una vez el usuario mande un mensaje cualquiera al numero. El identificador, ya sea una frase o un numero, que whatsapp API reciba, y ordene, se llama webhook. Vamos a utilizar ASP.NET puesto que es el servicio mas robusto que hay y se puede enlazar facilmente a diferentes tecnologias segun se requiera.

<https://business.whatsapp.com/blog/how-to-use-webhooks-from-whatsapp-business-api>

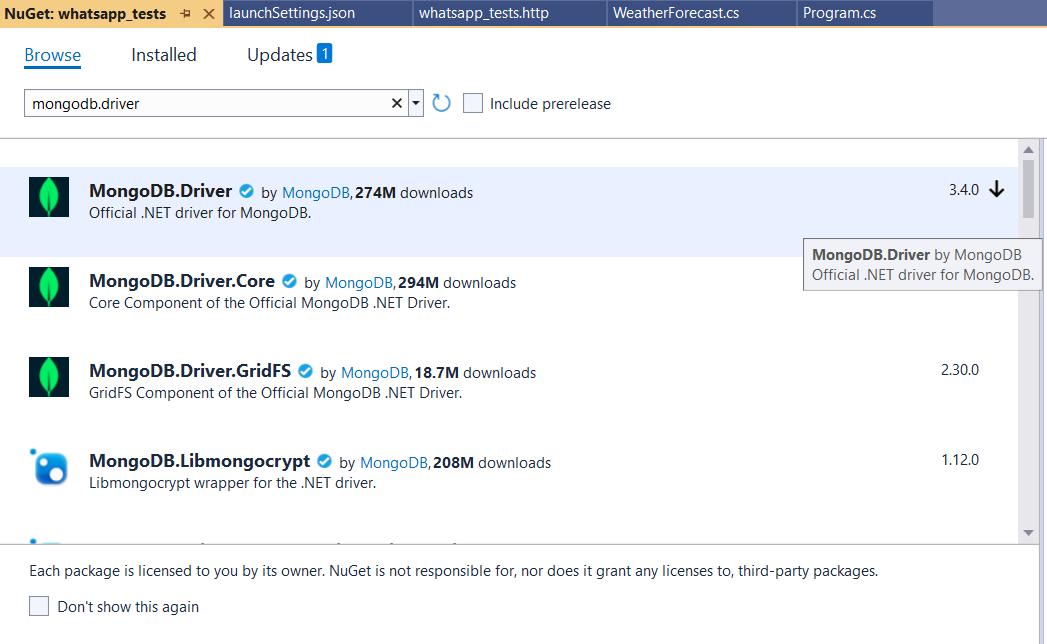
Este tutorial utiliza DynamoDB, que si bien es NoSQL, esta atado a Amazon web services (AWS) que vendra siendo util cuando nuestro proyecto requiera semejante potencia serverless sin latencia, pero MongoDB es por el otro lado, codigo fuente y menos costoso. El punto de aqui es:

Whatsapp API manda mensaje → es digerido por ASP.NET → la informacion de dicho mensaje es mostrada por lo que haya dentro de MongoDB → Esta informacion va hacia el usuario → el usuario responde con comandos basicos (si, no, numero 2, informacion personal) → el sistema obtiene la informacion y decide que hacer con ella.

4) MongoDB: Vamos a crear un cluster llamado “WhatsappBot” y una vez este listo, vamos a:



Ahora seleccionamos el driver:



¿Que es un webhook?

https://www.youtube.com/watch?v=tzRUzuaXV5A&list=PLi9cSkiBrnxa58GtrU8OdNiBErd20I0JY&index=11

Un webhook es una manera sencilla de un servicio para mandar informacion siempre que se ejecute una accion. Usted configura una aplicacion o servidor para recibir mensajes y darle la direccion al servicio. Cuando un evento acciona el servicio, este puede llamar al webhook para mandar el mensaje de su aplicacion.

B) Mensajes:

https://developers.facebook.com/docs/whatsapp/cloud-api/reference/messages

Objeto de usuario:

Contacts

1) whatsapp id (wa\_): numero de whatsapp registrado con un 1 en medio de la LADA y clave regional (por ejemplo, yo soy 523541090470, el wa\_id es 5213541090570).

2) whatsapp client number (input): numero real de whatsapp registrado (en mi caso, 523541090470)

Messages:

1) id: cadena encriptada de informacion utilizando wamid

Objeto del mensaje del servidor:

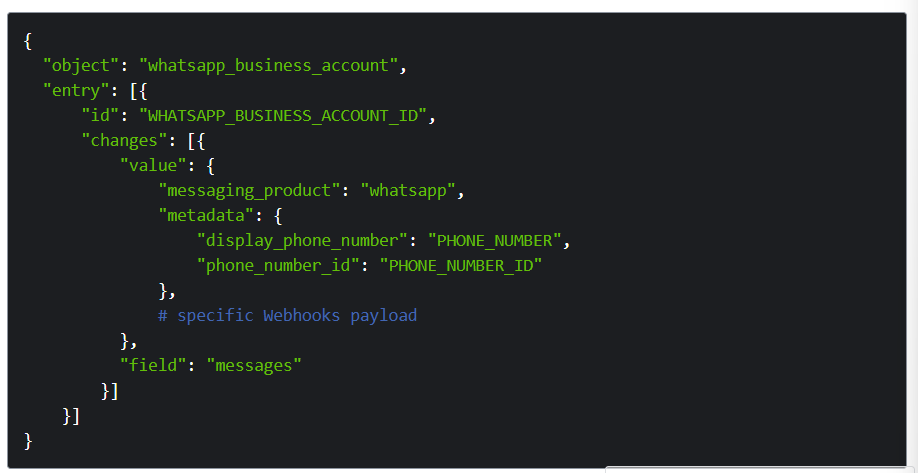
1) messaging\_product: “whatsapp”

2) recipient\_type: “individual”

3) to: “numero de usuario whatsapp

Webhook del usuario:

Todos los webhooks de whatsapp tienen el siguiente formato generico: (https://developers.facebook.com/docs/whatsapp/webhooks/)



Lo que vamos a hacer es tomar esta informacion en crudo, mandado desde el usuario, y deserealizar el JSON y tomar estos valores en un objeto, crear este objeto usuario para la sesion iniciada, y dependiendo de la informacion que tenga, mandar otro paquete JSON al usuario o tomar otra decision:

<https://stackoverflow.com/questions/7895105/deserialize-json-with-c-sharp>

Bibliografia:

<https://techcommunity.microsoft.com/blog/azurecommunicationservicesblog/building-a-whatsapp-ai-bot-for-customer-support/4277880>

<https://learn.microsoft.com/en-us/azure/communication-services/quickstarts/advanced-messaging/whatsapp/get-started?tabs=visual-studio%2Cconnection-string&pivots=programming-language-csharp>

<https://learn.microsoft.com/en-us/javascript/api/botbuilder-dialogs/?view=botbuilder-ts-latest>

https://developers.facebook.com/docs/whatsapp/cloud-api/messages/interactive-list-messages

<https://developers.facebook.com/docs/whatsapp/pricing/>

<https://developers.facebook.com/apps/>

https://developers.facebook.com/tools/explorer/

<https://business.facebook.com/latest/whatsapp_manager/overview?business_id=2757838967620520&asset_id=2384535638585628>

<https://developers.facebook.com/docs/whatsapp/cloud-api>

<https://developers.facebook.com/docs/whatsapp/cloud-api/get-started>

<https://github.com/microsoft/botframework-sdk>

<https://developers.facebook.com/docs/facebook-login/guides/access-tokens>

<https://stackoverflow.com/questions/46805634/facebook-api-access-token-session-has-expired>

<https://devils-heaven.com/facebook-access-tokens/>

<https://www.postman.com/meta/whatsapp-business-platform/collection/3kru5r6/whatsapp-business-management-api>

<https://developers.facebook.com/docs/whatsapp/cloud-api/guides/send-messages>

[https://developers.facebook.com/docs/whatsapp/cloud-api/get-started#configure-webhooks](https://developers.facebook.com/docs/whatsapp/cloud-api/get-started" \l "configure-webhooks)

<https://developers.facebook.com/docs/whatsapp/cloud-api/reference/messages>

<https://medium.com/@chandrashekharsingh25/build-a-restful-web-api-with-net-8-44fc93b36618>

https://business.whatsapp.com/blog/how-to-use-webhooks-from-whatsapp-business-api

<https://learn.microsoft.com/en-us/azure/bot-service/bot-service-design-principles?view=azure-bot-service-4.0>

<https://classic.yarnpkg.com/en/docs/usage>

<https://learn.microsoft.com/fr-fr/aspnet/core/tutorials/min-web-api?view=aspnetcore-9.0&tabs=visual-studio>

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mongo-app?view=aspnetcore-9.0&tabs=visual-studio>

<https://medium.com/@chandrashekharsingh25/build-a-restful-web-api-with-net-8-44fc93b36618>

<https://learn.microsoft.com/en-us/entra/identity-platform/tutorial-web-api-dotnet-core-build-app?tabs=workforce-tenant>

<https://learn.microsoft.com/en-us/entra/identity-platform/howto-call-a-web-api-with-curl?tabs=dotnet6%2Cbash&pivots=api>

<https://learn.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client>

<https://medium.com/bgl-tech/how-to-make-your-first-get-api-call-in-c-net-core-501134ee6e19>

<https://book.cakephp.org/5/fr/development/rest.html>

<https://business.whatsapp.com/blog/how-to-use-webhooks-from-whatsapp-business-api>

<https://learn.microsoft.com/en-us/aspnet/webhooks/>

<https://github.com/conficient/BlazorChatSample>

https://www.youtube.com/watch?v=erSgGcDcV8s