Contenu Afficher

Objet

Ce post décrit un prototype de ChatBot builder qui a été réalisé avec Oracle Application Express (Apex) et qui est destiné à Facebook Messenger. Pour illustration, j'ai créé un bot spécialisé pour un syndic immobilier qui illustre la plupart des possibilités que peut offrir un agent conversationnel de type Facebook Messenger.
On peut également imaginer des bots simplifiés pour des collectivités locales répondant à des question usuelles relatives à la vie locale, ou bien des bots spécialisés pour naviguer dans un catalogue d'objets.

La solution retenue repose sur un modèle de données qui contient les différentes « routes » de réponse ainsi que les modèles de message. Ce genre de ChatBot est à privilégier pour un ensemble fini de réponse. Il n'y a aucune intelligence ni moteur NLP. En revanche, aucune programmation n'est nécessaire: On se contente de remplir les différentes réponses en réaction à des messages.
Ce chatbot tire également parti du moteur de recherche textuel fourni par la Database Oracle et cela offre à l'utilisateur beaucoup de flexibilité dans la formulation des questions (insensibilité aux voyelles accentuées, par exemple, et *stemming* sur les formes conjuguées). Il est possible de rajouter un thésaurus avec des synonymes, ou bien une nomenclature.

Ce prototype inclut la récupération de la géolocalisation issue de Facebook Messenger, la récupération des photos éventuellement envoyées et l'authentification sur un site tiers (*Account Linking*).

Plusieurs Chatbots peuvent être gérés par le même moteur et peuvent être servis par une seule application Facebook.

Prérequis

Soit on utilise une instance APEX en cloud prête à l'emploi soit une installation manuelle on-premise ou en cloud.

- Installation APEX on-premise ou Cloud
    - Apex 19.2.x
    - Database Oracle parmi les variantes suivantes:
        - Standard
        - Enterprise ou
        - Express 18c
          (la variante Express Edition de la database ne peut fonctionner qu'à partir de la version 18c car des appels en HTTPS doivent être passés depuis le code PL/SQL, et il faut donc la présence d'un *wallet* pour enregistrer les certificats de Facebook)
        - Database Service en cloud
- Service APEX en cloud
    - Autonomous Transaction processing (ATP)
    - Always Free Tiers (qui comprend ATP)

Architecture simplifiée

Le serveur Chatbot se présente sous la forme d'un package PL/SQL qui effectue des actions en réponse aux *payloads* JSON envoyés par Facebook Messenger. Les échanges avec Facebook s'effectuent en appels REST via un

module REST (webhook dans la terminologie FB) avec deux *endpoints* :
l'un en GET pour l'enregistrement du chatbot et l'autre en POST pour la réception des messages envoyés par FB Messenger.
Ces échanges se font obligatoirement en SSL. Le rôle du package bot_engine est, in fine, de formater des réponses au format JSON et de maintenir un contexte par utilisateur.

Un *backoffice*, décrit plus bas, permet à un administrateur fonctionnel de créer des chatbots, déclarativement . Cela revient à définir les réponses sous la forme de widgets visuels tels que carrousels, boutons, *quick answers* etc.

Échantillons pour un chatbot de type Syndic immobilier

Description fonctionnelle

Ce chatbot est destiné aux copropriétaires d'une résidence gérée par un syndic immobilier. Le but est de faciliter les demandes d'informations relatives à la copropriété. Ici, il y aura autant de pages Facebook et de Bots que de résidences, mais c'est le même moteur APEX qui traitera les différents chatbots. Dans l'exemple qui suit, une intégration est effectuée avec un service documentaire. C'est celui d'Oracle qui a été utilisé (*Oracle Content & Experience Cloud*).

Lors de l'interaction avec le Bot, il est affiché un menu général (« *call to actions*«) ce qui facilite le démarrage de l'échange avec l'utilisateur qui n'a plus qu'à cliquer sur une des grandes catégories proposées (*Ma Copropriété*, *Mon syndic*, *Signaler un incident*, *Gestion courante*).
La suite du dialogue est majoritairement basée sur l'emploi de boutons et de

réponses rapides.

-
-
-
  Réponses simples
-
  Call to actions
-
  Carrousel
-
  Réponse simple
-
-
-
  Intégration Documents
-
-
-
-
  Account Linking
-
  Account Linking
-
  Déblocage d’une option
-
-
  Intégration Documents
-
  Intégration Documents

- Quick answers
- Quick answers
- Remontée d’une image
-
-
-
- Remontée localisation
-
- Remontée localisation

Backoffice réalisé avec Oracle Apex

Principes de construction

Un modèle relationnel a été conçu à partir de la signature des APIs de Facebook et ce modèle a été implémenté dans une database Oracle.
Pour des raisons de commodités, j’ai utilisé Oracle APEX pour la réalisation d’un backoffice à partir duquel on peut déclarer les différents chatbots et les questions auxquelles ceux-ci pourront répondre.

A ce stade, il ne s’agit que d’un prototype et le modèle de données aurait certainement besoin d’être revisité (vérifier les cardinalités, les règles de gestion …) pour bien correspondre au modèle objet de Facebook Messenger.

-

- Galerie des bots
- Arborescence du dialogue
- Log des images renvoyées par les utilisateurs

Conclusion

Ce prototype de chatbot Builder est un bon point de démarrage pour tester la validité et/ou la pertinence d'un agent conversationnel par rapport à une situation donnée.

Une fois testé, le chatbot peut être déployé en production en utilisant le même type de plateforme Oracle APEX. Comme on sollicite essentiellement le moteur PLSQL et que l'usage du stockage DB est très réduit, la variante DB Express 18c (qui est gratuite) peut très bien faire l'affaire pour une configuration modeste.
Si les contraintes de performances sont élevées, on pourra augmenter la puissance allouée à l'instance DB ou transposer le chatbot (aux prix d'une ré-écriture, il est vrai) sur une plateforme adaptée au traitement asynchrone, comme Node.js par exemple.

Author

- 

  Patrick

  GPM Factory