

*Sujet :*

# **Réalisation d'un chatbot et une application web basé sur une ontologie**

**Réalisé par :**

Ghanem Abdelghani

Ouifak Hafsaa

Ait abderrahmane Latifa

*Le 26/01/2019 à ENSIAS Rabat*

# Résumé

Dans notre projet nous possédons à la réalisation d'un chatbot basé sur une ontologie « GeOntology » comme base de connaissance et pour la gestion du dialogue basé sur les résultats fournis par cette ontologie.

Le chatbot commence après la réception du message par le traitement du langage naturel à l'aide de la plateforme d'intelligence artificielle wit.ai de Facebook, puis possède à chercher les réponses convenables en utilisant l'ontologie localisée dans un serveur fueski installé dans notre machine.

On offre ainsi la possibilité de basculer vers le mode développeur en utilisant une application web qui sert à exécuter directement des requêtes SPARQL.

# Abstract

In our project we have realised a chatbot based on an ontology "GeOntology" as knowledge base and for the dialogue management based on the results provided by this ontology.

The chatbot starts after receiving the message through natural language processing using Facebook's wit.ai artificial intelligence platform, and then has to search for the correct responses using the ontology located in a fueski server installed in our machine.

We thus offer the possibility of switching to the developer mode by using a web application that is used to directly execute SPARQL queries.

# Sommaire

<b>Introduction générale .....</b>	<b>6</b>
<b>Chapitre 1 : .....</b>	<b>7</b>
<b>Etude théorique .....</b>	<b>7</b>
I. Approche de travail .....	8
A. Présentation de projet à réaliser: .....	8
B. Chatbot .....	8
C. Intelligence artificielle .....	8
D. Web sémantique .....	8
E. Ontologie : .....	9
II. Planning de travail .....	9
<b>Chapitre 2 : .....</b>	<b>10</b>
<b>Spécification des besoins et conception.....</b>	<b>10</b>
I. Spécification des besoins .....	11
A. Diagrammes de cas d'utilisation .....	11
II. Conception .....	11
A. Base de données .....	11
B. Diagrammes de séquence .....	12
<b>Chapitre 3 : .....</b>	<b>14</b>
<b>Outils techniques .....</b>	<b>14</b>
I. Outils matériels .....	15
II. Outils logiciels.....	15
A. Plateformes .....	15
B. Services.....	16
C. Langages de structuration .....	19
D. Librairie.....	20
E. Architecture.....	21

F.	Langages de programmation .....	21
G.	Modules NodeJs .....	22
H.	Langage de requêtes .....	23
I.	Langages de commande .....	24
J.	IDE de développement.....	24
K.	Logiciel de modélisation.....	24
L.	Interfaces de programmation (APIs) .....	25
<b>Chapitre 4 : .....</b>		<b>27</b>
<b>Réalisation .....</b>		<b>27</b>
I.	Configuration.....	28
A.	Traitement du langage naturel .....	28
B.	Configuration Webhook .....	28
C.	Application heroku .....	28
D.	Git pour le transfert des fichiers .....	29
E.	Configuration Ngrok .....	30
F.	Configuration Fuseki.....	30
II.	Mode développeur.....	31
A.	Interface page de Query :.....	31
III.	Interface de chatbot.....	32
IV.	Données analytique.....	44
<b>Conclusion générale.....</b>		<b>45</b>
<b>Références.....</b>		<b>46</b>
<b>Glossaire.....</b>		<b>47</b>

# Liste des figures

Figure 1 : Différentes phases de réalisation.....	9
Figure 2 : Diagramme de cas d'utilisation d'un visiteur.....	11
Figure 3 : Base de données des utilisateurs.....	12
Figure 4 : Diagramme de séquence.....	13
Figure 5 : Interface de la plateforme wit.ai.....	15
Figure 6 : Plateforme wit.....	28
Figure 7 : Configuration du webhook.....	28
Figure 8 : Les logs de l'application heroku.....	29
Figure 9 : Git pour le transfert du fichiers.....	29
Figure 10 : Outils Ngrok.....	30
Figure 11 : Sparql over http.....	30
Figure 12 : Application web pour l'exécution des requêtes Sparql.....	31
Figure 13 : Première conversation avec le chatbot.....	32
Figure 14 : Guide du chatbot.....	33
Figure 15 : Menu et sous menu.....	33
Figure 16 : Listes des « islands » et ceux en europe et pas en Italie.....	34
Figure 17 : Liste des « islands » et capital d'un pays.....	36
Figure 18 : Requête sur les « islands ».....	37
Figure 19 : Liste des oceans et auquel oceans appartient un « sea ».....	39
Figure 20 : List des « hills » et location d'un « hill ».....	40
Figure 21 : Liste des « seas ».....	41
Figure 22 : Liste des montagnes et location d'une montagne.....	42
Figure 23 : Détection du non sens.....	43
Figure 24 : Passage au mode développeur.....	44
Figure 25 : Données analytiques du chatbot.....	44

# Introduction générale

L'art de la conversation n'est peut-être déjà plus l'apanage du genre humain. Entre les assistants vocaux comme (Cortana) qui s'invitent sur nos smartphones et les boîtiers connectés qui réagissent au son de la voix (Alexa, qui équipe l'enceinte Echo d'Amazon), les machines ont dû apprendre à nous comprendre afin de répondre intelligemment à nos besoins. Une tendance qui se généralise aux échanges oraux comme écrits (notamment sur les réseaux sociaux), et qui a un nom : chatbots (ou agents conversationnels, chatterbots en anglais). *(Sarah Sermondadaz, 19.12.2016)*

Une étude récente de « business Insider [1] » indiquait que 80% des entreprises veulent utiliser des chatbots d'ici l'an 2020. Une donnée qui ne fait que conforter l'importance d'utiliser cet outil marketing puissant. Mais pourquoi un chatbot Messenger ?

L'application Messenger est la plus populaire, en plus de la large richesse des APIs [2] fournies par Facebook, ainsi qu'elle a profité des derniers progrès de l'intelligence artificielle et le traitement du langage naturel. Ce qui nous intéresse dans notre projet, en se basant sur des ontologies comme base de connaissance dans le service du web sémantique.

Le Web sémantique est une extension du Web standardisée par le World Wide Web Consortium(W3C). Ces standards encouragent l'utilisation de formats de données et de protocoles d'échange normés sur le Web, en s'appuyant sur le modèle Resource Description Framework (RDF), ainsi ils fournissent un modèle qui permet aux données d'être partagées et réutilisées entre plusieurs applications.

Dans le cadre de ce projet nous avons essayé d'exploiter les technologies offertes par la société Facebook en se basant sur une ontologie « GeOntolgy », afin de réaliser un agent conversationnel capable de générer des réponses en langage naturel après l'exécution d'un ensemble de requêtes derrière.

# Chapitre 1 :

## Etude théorique

## I. Approche de travail

### A. Présentation de projet à réaliser:

Le travail porte sur la réalisation d'un chatbot. Ce chatbot est lié à une ontologie « GeOntology » par laquelle il peut générer plusieurs résultats et enrichir la conversation. L'utilisateur peut ainsi passer au mode développeur dont il peut saisir directement ses requêtes sparql.

#### Travail principal :

- Conversation fluide
- Compréhension langage naturel
- Analyse, reformulation et intégration de données récupérées de l'ontologie dans les conversations
- Mode développeur

### B. Chatbot

Un chatbot est un robot logiciel pouvant dialoguer avec un individu ou consommateur par le biais d'un service de conversations automatisées effectuées en grande partie en langage naturel. Le chatbot utilise à l'origine des bibliothèques de questions et réponses, mais les progrès de l'intelligence artificielle lui permettent de plus en plus « d'analyser » et « comprendre » les messages et d'être doté de capacités d'apprentissage liées au machine Learning. *(B.Bathelot)*

### C. Intelligence artificielle

L'intelligence artificielle (IA) est « l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence ». Elle correspond donc à un ensemble de concepts et de technologies plus qu'une discipline autonome constituée.

Yann LeCun, chercheur français en deep learning désormais directeur de la recherche en intelligence artificielle (IA) de Facebook affirme que l'IA est une discipline qui, pour progresser, doit relever un nouveau défi : celui de l'apprentissage non supervisé (ou "auto supervisé"). État des lieux et explications. *(Sarah Sermondadaz)*

### D. Web sémantique

Le Web sémantique, ou toile sémantique<sup>1</sup>, est une extension du Web standardisée par le World Wide Web Consortium (W3C). Ces standards encouragent l'utilisation de formats de données et de protocoles d'échange normés sur le Web, en s'appuyant sur le modèle Resource Description Framework (RDF).

Selon le W3C, « le Web sémantique fournit un modèle qui permet aux données d'être partagées et réutilisées entre plusieurs applications, entreprises et groupes d'utilisateurs ».



L'expression a été inventée par Tim Berners-Lee (inventeur du Web et directeur du W3C), qui supervise le développement des technologies communes du Web sémantique. Il le définit comme « une toile de données qui peuvent être traitées directement et indirectement par des machines pour aider leurs utilisateurs à créer de nouvelles connaissances ». Pour y parvenir, le Web sémantique met en œuvre le Web des données qui consiste à lier et structurer l'information sur Internet pour accéder simplement à la connaissance qu'elle contient déjà.

## E. Ontologie :

En informatique et en science de l'information, une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Plus simplement, on peut aussi dire que l'« ontologie est aux données ce que la grammaire est au langage » **Wikipedia**.

Dans notre projet on a travaillé sur une ontologie « GeOntology », créé par

## II. Planning de travail

Le tableau suivant montre les différentes tâches réalisées pendant ce projet, ensuite la figure 4 illustre la répartition des tâches dans la phase de réalisation.

Tâches		
Nom	Date de début	Date de fin
Cahier de charge	01/01/19	01/01/19
Formation+Conception	02/01/19	08/01/19
Réalisation	10/01/19	24/01/19
facebook app, wit app(API) and intents+ Firebase	10/01/19	11/01/19
heroku app(deploy)+ webhook	12/01/19	13/01/19
API Facebook (Graph)+Ajout d'entités	14/01/19	14/01/19
installation serveur fueski+ ngrok	15/01/19	19/01/19
ontologie+requetes	20/01/19	22/01/19
page web php mode développeur	23/01/19	24/01/19
Test	25/01/19	27/01/19

Figure 1 : Différentes phases de réalisation

# Chapitre 2 :

## Spécification des besoins et conception

# I. Spécification des besoins

## A. Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

Il représente les fonctionnalités d'un système et les interactions entre le système et le monde extérieur.

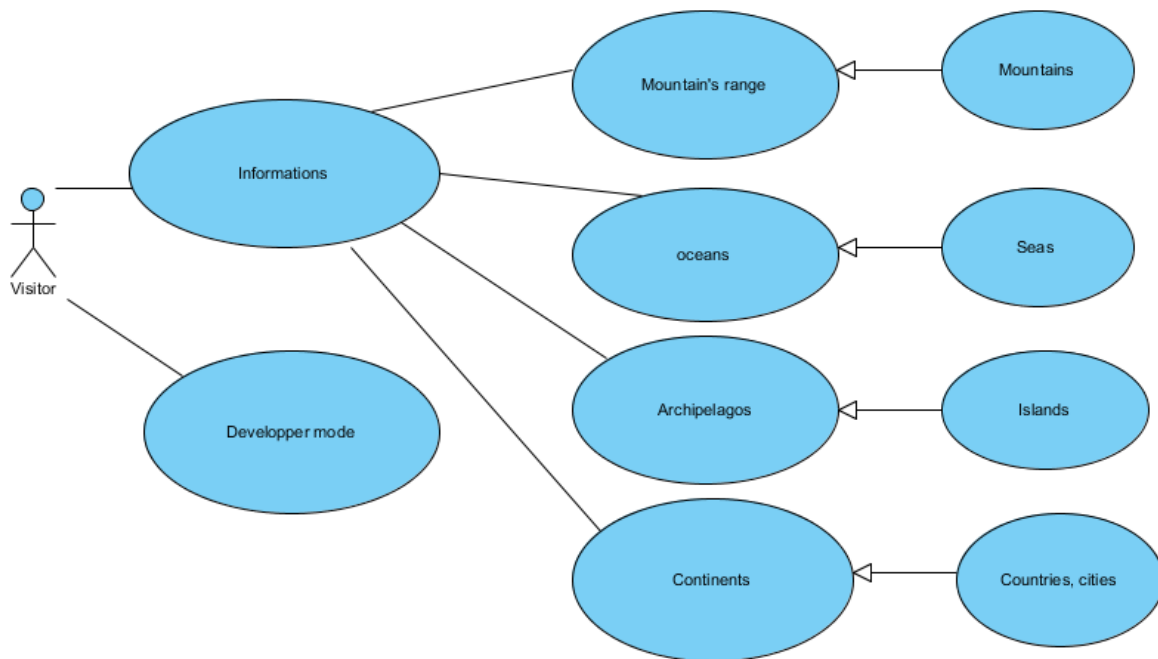


Figure 2 : Diagramme de cas d'utilisation d'un visiteur

## II. Conception

C'est une étape importante du cycle de développement d'un logiciel. Elle se base essentiellement sur la bonne spécification et l'analyse des besoins.

### A. Base de données

Dans la base de données nous sauvegardons la structure « accounts » pour tous les visiteurs de notre application.

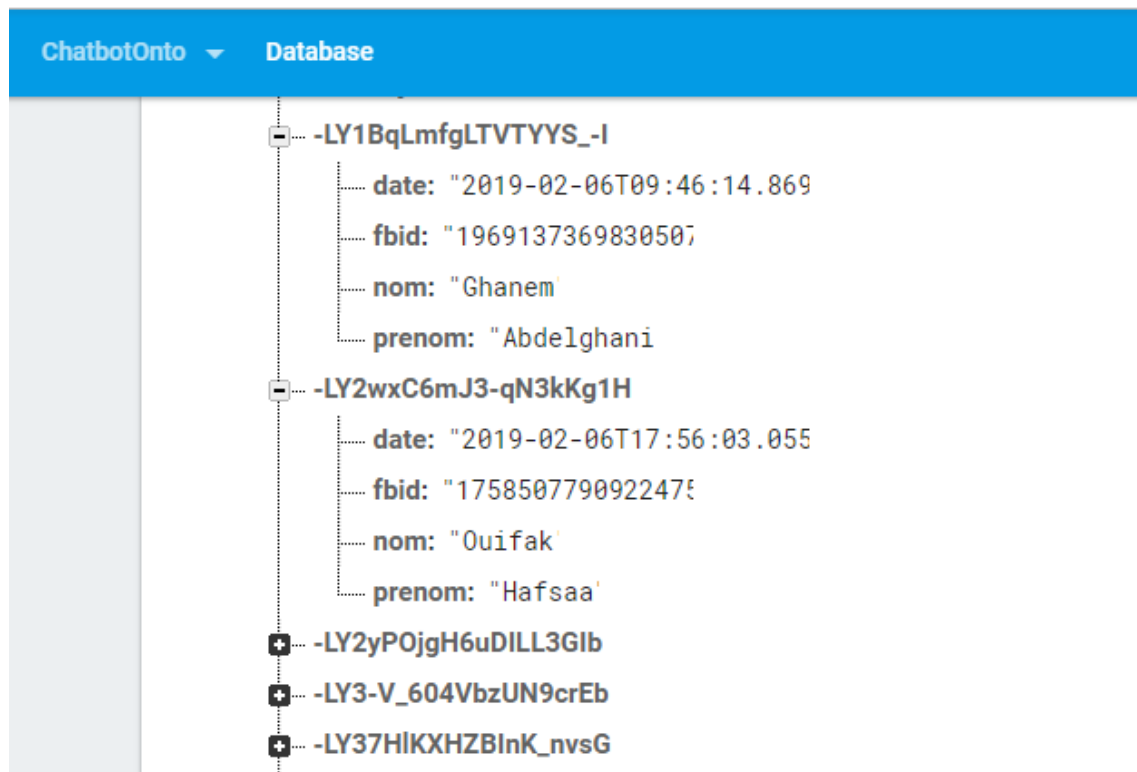


Figure 3 : Base de données des utilisateurs

## B. Diagrammes de séquence

Le diagramme de séquence sert à décrire les scénarios d'exécutions possibles des cas d'utilisation.

Quand un internaute envoie un message, le webhook reçoit un événement et cherche ou crée une nouvelle session, ensuite il envoie le message à la plateforme Wit pour le traitement du texte et l'identification de l'entité convenable, pour choisir l'action prochaine et exécuter les requêtes nécessaires sur l'ontologie, qui rendra par la suite les données à utiliser dans la réponse en format JSON, et enfin la réponse sera envoyée à l'internaute.

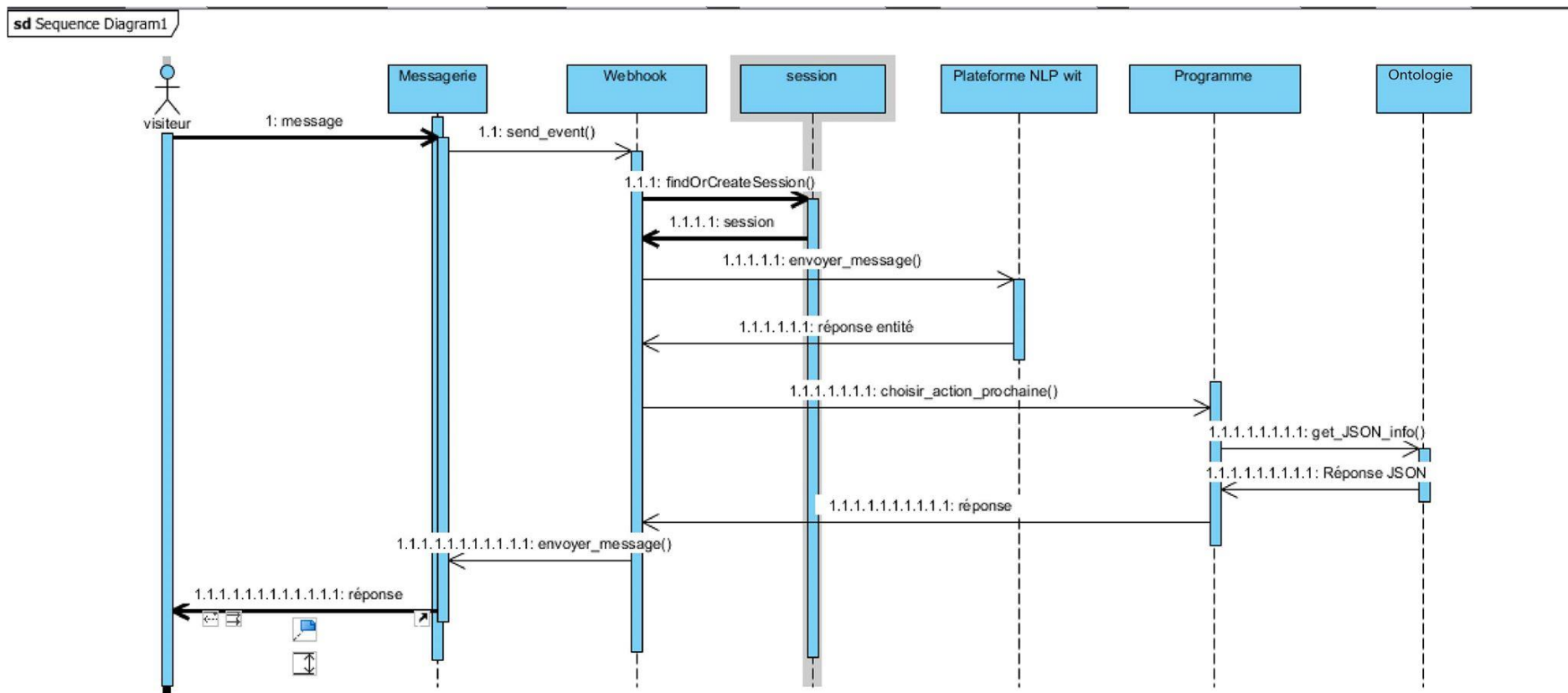


Figure 4 : Diagramme de séquence

# Chapitre 3 :

## Outils techniques

## I. Outils matériels

### Machine de développement :

- **Asus** : Microsoft window 7, 1 Processeur(s) installé(s) : Intel inside i7 , NVIDIA GEFORCE 840 M
- **Toshiba** : Microsoft Windows 10 Entreprise, 1 processeur(s) installé(s) : Intel64 Family 6 Model 58 Stepping 9 GenuineIntel ~1200 MHz, Mémoire physique totale: 3 988 Mo

## II. Outils logiciels

### A. Plateformes

#### 1. Wit.ai

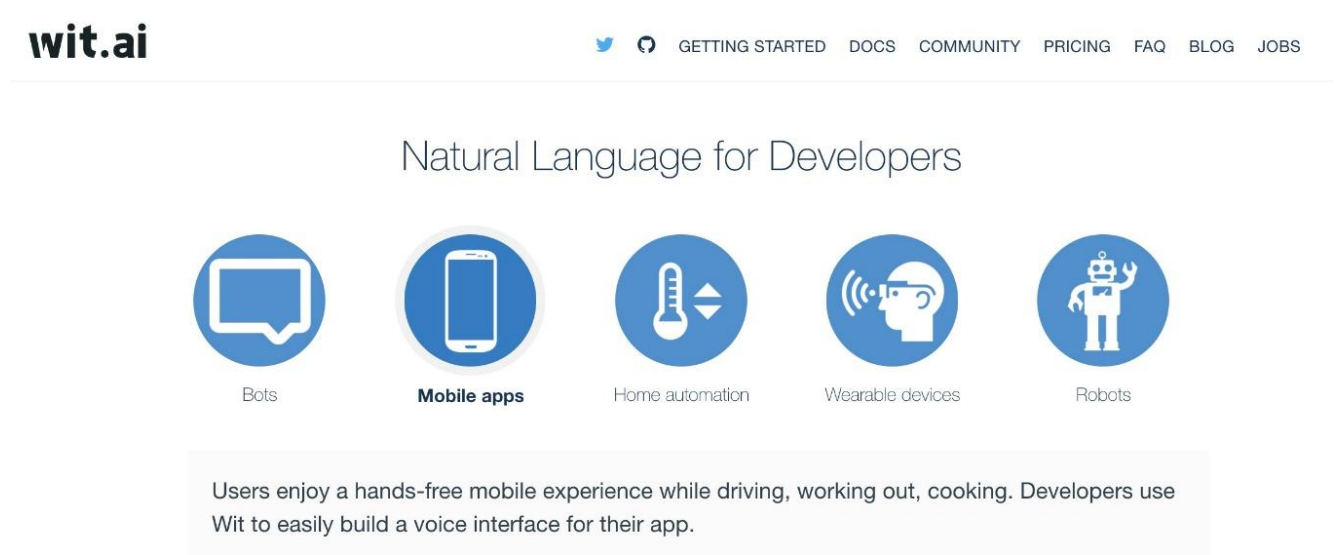


Figure 5 : Interface de la plateforme wit.ai

Wit.ai est une API de service de cloud pour la reconnaissance de la parole et le traitement du langage naturel, à utiliser avec les robots, les applications et les périphériques.

Wit.ai peut aider à :

- ✓ Comprendre le langage naturel: analyser un message (voix ou texte) en données structurées
- ✓ Converse: prédire l'action suivante du robot (Bot Engine)

## 2. NodeJs



Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.

Elle utilise la machine virtuelle V8 [5] connue par sa vitesse.

Parmi les modules natifs de Node.js, on retrouve http [6] qui permet le développement de serveur http. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

Node.js est de plus en plus populaire comme plateforme serveur.

### B. Services

#### 1. Apache Jena Fueski



Apache Jena Fuseki est un serveur SPARQL. Il peut être exécuté en tant que service du système d'exploitation, en tant qu'application Web Java (fichier WAR) et en tant que serveur autonome. Il fournit une sécurité (avec Apache Shiro) et dispose d'une interface utilisateur pour la surveillance et l'administration du serveur.

Il fournit les protocoles SPARQL 1.1 pour la requête et la mise à jour, ainsi que le protocole SPARQL Graph Store.

Fuseki est étroitement intégré à TDB pour fournir une couche de stockage persistante transactionnelle robuste. Il intègre la requête de texte Jena et la requête spatiale Jena. Il peut être utilisé pour fournir le moteur de protocole à d'autres systèmes de requête et de stockage RDF.



## 2. Ngrok



ngrok crée un tunnel de l'internet public vers un port de votre machine locale. Vous pouvez donner cette URL à n'importe qui pour lui permettre d'essayer un site Web que vous développez sans effectuer de déploiement.

Il capture tout le trafic à travers le tunnel. Il affiche des informations sur le trafic HTTP pour votre inspection. Les octets de requête / réponse bruts, les en-têtes et les données de formulaire analysés, la vérification de la syntaxe JSON / XML et plus encore sont inclus. Il peut également rejouer les demandes.

Par défaut, ngrok utilisera ngrok.com en tant que relais tiers. Ce service est fourni gratuitement et sans inscription, mais il est possible d'obtenir des fonctionnalités supplémentaires en s'inscrivant au service (qui est payant à la demande). Cependant, il est possible de configurer et d'utiliser son propre serveur.

## 3. Firebase



**Firebase** est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.

### NoSQL :

En informatique et en bases de données, NoSQL désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles. L'explicitation du terme la plus populaire de l'acronyme est Not only SQL (« pas seulement SQL » en anglais) même si cette interprétation peut être discutée<sup>1</sup>.

La définition exacte de la famille des SGBD NoSQL reste sujette à débat. Le terme se rattache autant à des caractéristiques techniques qu'à une génération historique de SGBD qui a émergé autour des années 2010.

#### 4. Heroku



**Heroku** est un service de cloud computing de type plate-forme en tant que service.

L'architecture Heroku vise à mettre à disposition de l'utilisateur un environnement de déploiement et d'exécution adapté à son besoin.

#### 5. GitHub



**GitHub** est un service en ligne qui permet d'héberger ses repositories de code. GitHub est un outil gratuit pour héberger du code open source, et propose également des plans payants pour les projets de code privés. C'est le numéro 1 mondial et il héberge plus d'une dizaine de millions de repositories !

#### 6. Npm



Npm est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut<sup>4</sup>. Npm fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.js disponibles sur le dépôt npm.

## 7. JSON viewer :

### JSON :

JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple.



JSONView est une extension de Google Chrome pour afficher JSON et JSONP.

## C. Langages de structuration

### 1. HTML 5



HTML5 (HyperText Markup Language 5) : est la dernière révision majeure d'HTML (format de données conçu pour représenter les pages web). Cette version est en développement en 2013. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM :

HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API.

#### Dans notre projet :

Pour la structuration des pages web utilisées .

### 2. CSS 3



Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML

et XML. Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000. CSS3 est le petit nouveau dans la famille feuille de style.

Il offre de nouvelles possibilités intéressantes pour créer un impact avec les dessins, il permet d'utiliser des feuilles de style plus variées pour des effets de style beaucoup plus puissants et variés.

#### **Dans notre projet :**

Pour organiser la vue des pages web.

### **D. Librairie**

#### **1. JQuery**



jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006 par John Resig.

#### **2. Bootstrap 3**



**Bootstrap** est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

#### **Dans notre projet :**

Pour rendre les vue web responsive.

## E. Architecture

### 1. Ajax



L'architecture informatique ajax (acronyme d'asynchronous JavaScript and XML : JavaScript et XML asynchrones) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

Ajax combine JavaScript, les requêtes de type XMLHttpRequest, les manipulations du DOM, ainsi qu'un format de données (XML ou JSON), afin d'améliorer maniabilité et confort d'utilisation des applications internet riches :

DOM et JavaScript permettent de modifier l'information présentée dans le navigateur en respectant sa structure ;

l'objet XMLHttpRequest sert au dialogue asynchrone avec le serveur Web ;

XML, cité dans l'acronyme, était historiquement le moyen privilégié pour structurer les informations transmises entre serveur Web et navigateur, de nos jours le JSON tend à le remplacer pour cet usage.

Dans notre projet, on a utilisé AJAX surtout pour la pagination des pages des résultats.

## F. Langages de programmation

### 1. Javascript/NodeJs

JavaScript avait toujours été utilisé du côté du client, c'est-à-dire du côté du visiteur qui navigue sur notre site. Le navigateur web du visiteur (Firefox, Chrome, IE...) exécute le code JavaScript et effectue des actions sur la page web.

Nous pouvons toujours utiliser du JavaScript côté client pour manipuler la page HTML.

Par contre, Node.js offre un environnement côté serveur qui nous permet aussi d'utiliser le langage JavaScript pour générer des pages web. En gros, il vient en remplacement de langages serveur comme PHP, Java EE, etc. (*Mathieu NEBRA*)

Avec Node.js, nous pouvons créer des applications rapides comme :

- Un serveur de Chat
- Un système d'upload très rapide

### Rapidité de nodeJs :

Node.js est connu par sa rapidité, cela tient principalement à deux choses : le moteur V8 et son fonctionnement non bloquant.



#### 2. EJS

**EJS** est un langage de modèle simple qui permet de générer du code HTML avec JavaScript simple. Aucune religiosité sur la façon d'organiser les choses. Pas de réinvention de l'itération et du contrôle-flux. C'est juste du JavaScript.

### Dans notre projet :

Pour faire appel à une page de contenu HTML dans une application de nodeJs.

#### 3. Ruby



Ruby est un langage de programmation libre. Il est interprété, orienté objet et multi-paradigme. Le langage a été standardisé au Japon en 2011 (JIS X 3017:2011), et en 2012 par l'Organisation internationale de normalisation (ISO 30170:2012).

#### 4. PHP



PHP: Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

## **G. Modules NodeJs**

#### 1. Body parser :

**body-parser** extrait la partie entière du corps d'un flux de requête entrant et l'expose sur req.body.

Le middleware faisait déjà partie d'Express.js mais maintenant nous devons l'installer séparément.

Ce module body-parser analyse les données JSON, buffer, string et URL encodées soumises à l'aide de la requête HTTP POST

## 2. Crypto :

Pour crypter les données sensibles et les sécuriser.

## 3. Express :

Express.js est un framework pour construire des applications web basées sur Node.js. C'est de fait le framework standard pour le développement de serveur en Node.js.

## 4. Node-fetch :

Pour récupérer des données sur http avec le nœud js.

## 5. Request :

La requête est conçue pour être la plus simple possible pour effectuer des appels http. Elle prend en charge HTTPS et suit les redirections par défaut.

## 6. Requestify :

Requestify est un client HTTP super facile à utiliser et extensible pour nodeJS + il supporte le cache.

## 7. Firebase / Firebase admin :

Le package npm JavaScript Firebase contient du code pouvant être exécuté dans le navigateur après la combinaison des modules que nous utilisons avec un regroupeur de packages (par exemple, Browserify, Webpack).

## 8. node-wit :

Interface facile pour l'analyse de langage naturel wit.ai.

# H. Langage de requêtes

## 1. Sparql



SPARQL est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. Son nom est un acronyme récursif qui signifie SPARQL Protocol and RDF Query Language.

## I. Langages de commande

### 1. Bash :



Bash est un shell Unix et un langage de commande écrit par Brian Fox pour le projet GNU en tant que logiciel libre de remplacement pour le shell Bourne. C'est un processeur de commandes qui s'exécute généralement dans une fenêtre de texte dans laquelle l'utilisateur tape des commandes qui provoquent des actions. Bash peut également lire et exécuter des commandes à partir d'un fichier, appelé script shell.

### 2. Batsh/CMD :

Batch est un langage de script, il permet d'exécuter une suite d'instructions ayant un but précis et pouvant servir à l'automatisation de tâches (par exemple, renommer tous les fichiers d'un dossier d'un coup). Cette suite d'instructions est interprétée par l'interpréteur de commande Windows : la console (cmd.exe).

#### Dans notre projet :

Nous utilisons ces Shells pour se connecter à l'hébergeur « heroku », à l'aide du service de github.

Aussi nous utilisons le script du Bash pour configurer le menu persistant du chatbot, le message d'accueil et le bouton démarrer.

## J. IDE de développement

### 1. Atom



**Atom** est un éditeur de texte libre pour OS X, GNU/Linux et Windows développé par GitHub. Il supporte des plug-ins écrits en Node.js et implémente Git Control. La plupart des extensions sont sous licence libre et sont maintenues par la communauté<sup>8</sup>. Atom est basé sur Chromium et Electron, et est écrit en CoffeeScript<sup>9</sup>. Il est aussi utilisé en tant qu'IDE.

## K. Logiciel de modélisation



## 1. Visual Paradigm :



Visual Paradigm (VP-UML) est un outil UML CASE prenant en charge UML 2, SysML et la notation BPMN (Business Process Modeling Notation) du groupe OMG (Object Management Group). En plus de la prise en charge de la modélisation, il fournit des capacités de génération de rapports et d'ingénierie de code, y compris la génération de code. Il peut effectuer une ingénierie inverse des diagrammes à partir du code et fournir une ingénierie d'aller-retour pour différents langages de programmation.

### Dans notre projet :

Nous avons utilisé VisualParadigm pour réaliser les diagrammes de cas d'utilisation, de séquences et d'activités.

## L. Interfaces de programmation (APIs)

### 1. API et API REST

Une interface de programmation applicative (souvent désignée par le terme API pour application programming interface) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

Il existe actuellement deux types d'architecture très utilisées pour les API : Simple Object Access Protocol (SOAP) et Representational State Transfer (REST).

Une API REST se doit d'être sans état ou stateless en anglais. La communication entre le client et le serveur ne doit pas dépendre d'un quelconque contexte provenant du serveur. Ainsi, chaque requête doit contenir l'ensemble des informations nécessaires à son traitement. Cela permet au de traiter indifféremment les requêtes de plusieurs clients via de multiples instances de serveurs.

Dans notre projet nous avons utilisés plusieurs APIs. Ci-dessous la liste de ces APIs :

### 2. Wit

L'api de Wit nous a permis de traiter le texte saisi et détecter l'entité auquel il appartient.

### 3. Firebase

L'api de firebase nous a permis d'accéder aux données collectées en temps réel.

#### 4. Graph

L'api graph de Facebook nous a permis d'envoyer des messages sur la plateforme Messenger de Facebook et collecter les données concernant chaque visiteur.

# Chapitre 4 :

## Réalisation

## I. Configuration

### A. Traitement du langage naturel

On crée une application dans la plateforme wit et on commence à apprendre du langage naturel à notre programme :

Test how your app understands a sentence

You can train your app by adding more examples

The screenshot shows the Wit.ai training interface. At the top, there's a text input field containing the sentence "give me some action movies". Below this, there are two columns of dropdown menus for labeling the intent and entities. The first column has a dropdown labeled "intent" with the selected value "Asking\_for\_movies" and a score of 1.000. The second column has a dropdown labeled "movie\_type" with the selected value "action" and a score of 0.848. Below these, there's a button "Add a new entity" and a "Validate" button. A small bar chart icon is visible on the right side.

Figure 6 : Plateforme wit

### B. Configuration Webhook

Il faut créer une application Facebook et configurer le webhook pour recevoir les messages de notre application.

The screenshot shows the Facebook Webhooks configuration page. At the top, there's a "Webhooks" section with an "Edit events" button. Below this, there's a text box explaining that to receive messages and other events sent by Messenger users, the app should enable webhooks integration. To the right of this text is a green checkmark and the word "Complete". Below this, there's a list of selected events: "messages, messaging\_postbacks, messaging\_optins, message\_deliveries, message\_reads, messaging\_pre\_checkouts, messaging\_checkout\_updates, messaging\_account\_linking, messaging\_referrals". At the bottom, there's a section for selecting a page to subscribe the webhook to, with a "Select a Page" button and the text "Subscribed pages: ChatbotOnto".

Figure 7 : Configuration du webhook

### C. Application heroku

Création d'un compte et un projet dans heroku pour pouvoir déployer l'application en ligne.

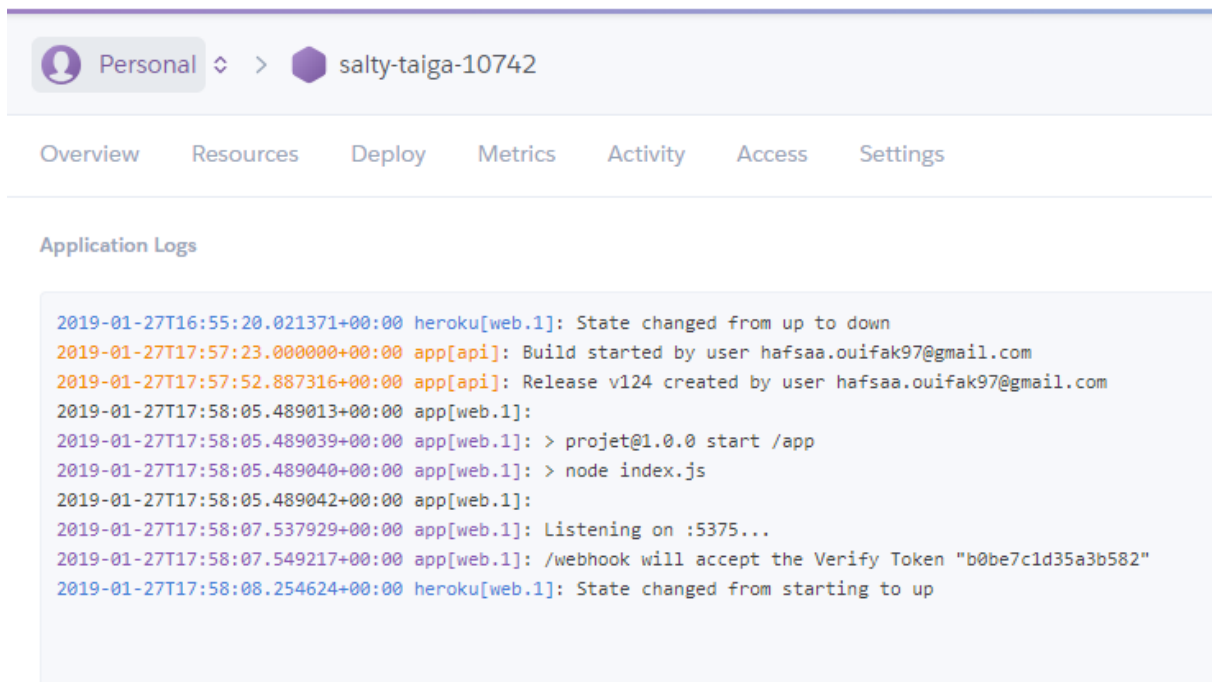


Figure 8 : Les logs de l'application heroku

## D. Git pour le transfert des fichiers

Utilisation de git pour le transfert des fichiers après qu'on se connecte au compte heroku et qu'on configure le proxy :

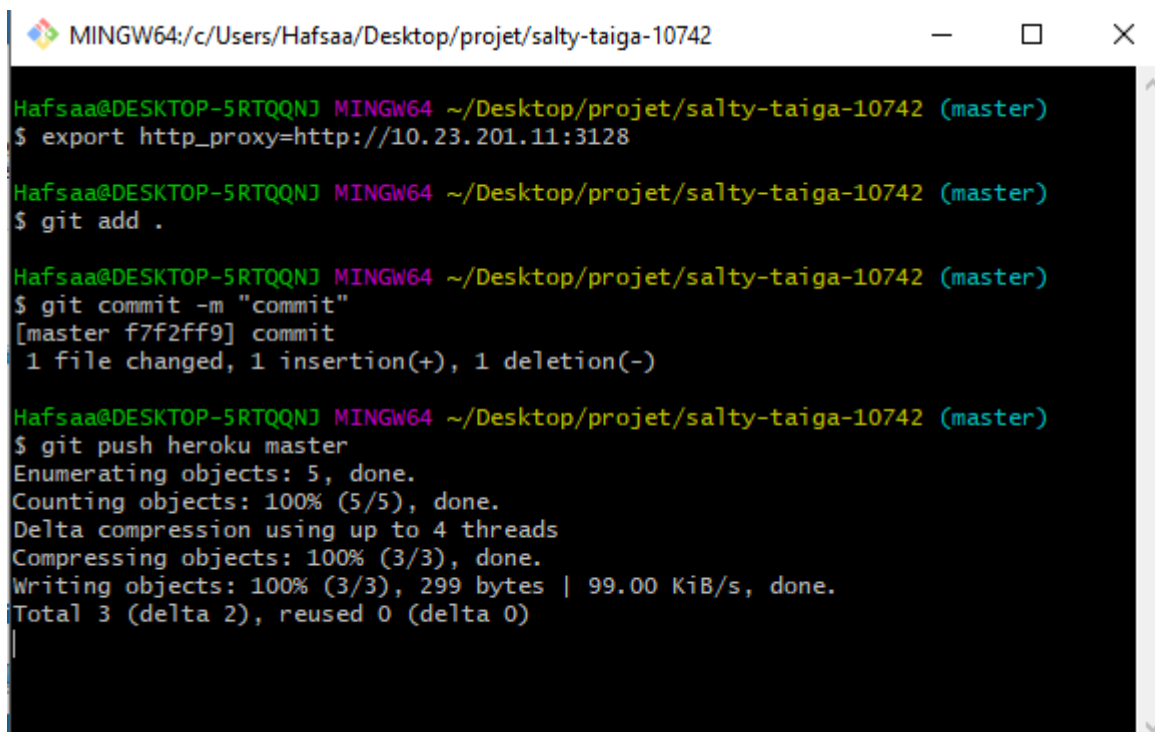


Figure 9 : Git pour le transfert du fichiers

## E. Configuration Ngrok

Ngrok pour lancer notre serveur local en ligne avec une adresse http ou https.

```
Administrateur : C:\ngrok.exe - ngrok http 3030

ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 28 minutes
Version             2.2.8
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://3f68dd2f.ngrok.io -> localhost:3030
Forwarding           https://3f68dd2f.ngrok.io -> localhost:3030

Connections         ttl      opn      rt1      rt5      p50      p90
                   10       1        0.00     0.00     36.65    44.02

HTTP Requests
-----
GET /$/ping          200 OK
```

Figure 10 : Outils Ngrok

## F. Configuration Fuseki

Activer **SOH** (SPARQL Over HTTP) créé en Ruby pour exécuter des requêtes http :

```
Invite de commandes

Microsoft Windows [version 10.0.17134.407]
(c) 2018 Microsoft Corporation. Tous droits réservés.

C:\Users\Hafsaa>cd /

C:\>cd fueski
Le chemin d'accès spécifié est introuvable.

C:\>cd fuseki

C:\fuseki>cd bin

C:\fuseki\bin>set http_proxy=http://10.23.201.11:3128

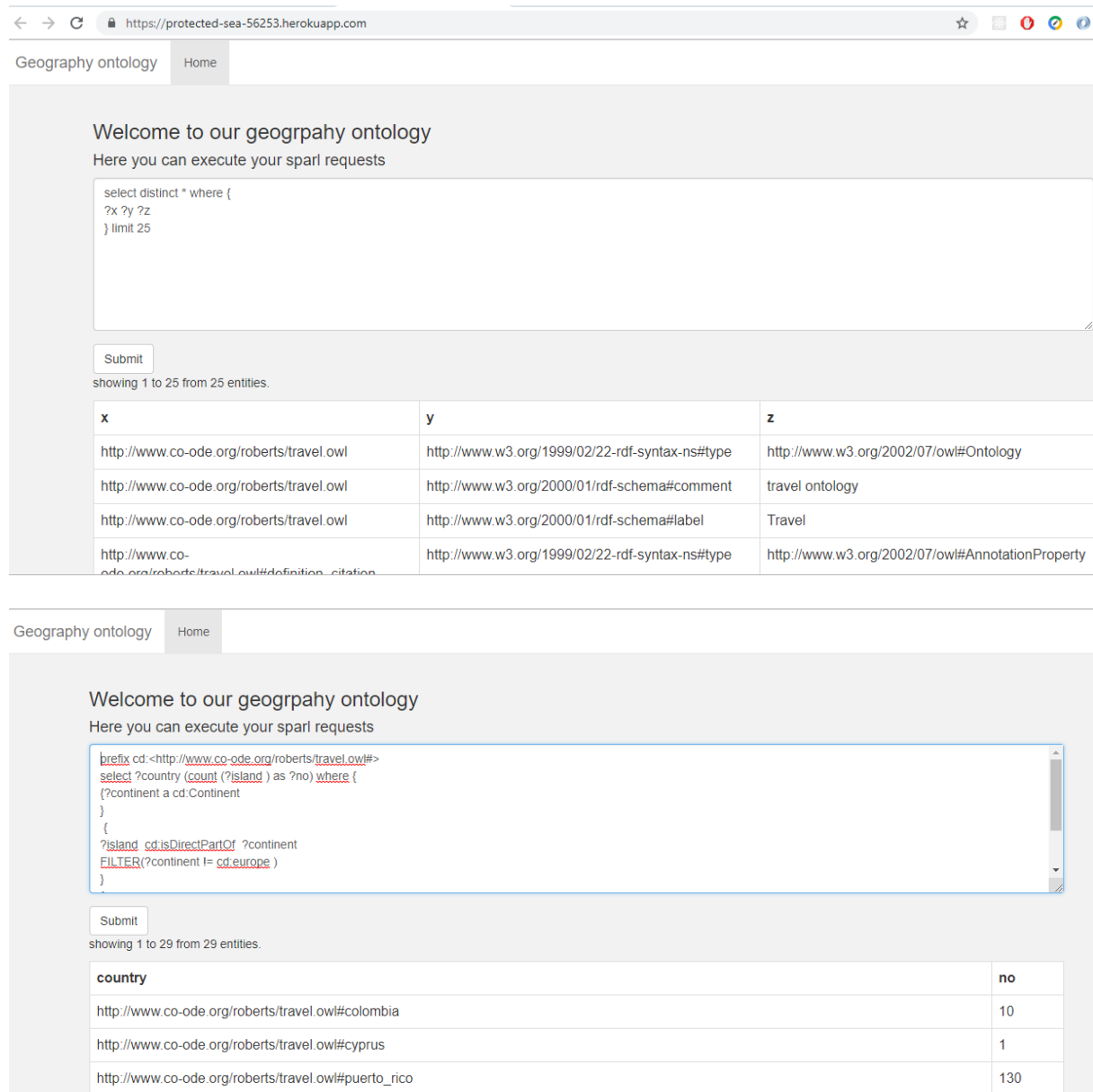
C:\fuseki\bin>ruby s-get http://3f68dd2f.ngrok.io/movies default
@prefix :      <http://www.semanticweb.org/vinu/ontologies/2014/6/untitled-ontology-102#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

Figure 11 : Sparql over http

## II. Mode développeur

### A. Interface page de Query :

Le mode développeur se réalise à l'aide d'une page web en PHP et AJAX comme suit :



Geography ontology Home

Welcome to our geogrpahy ontology  
Here you can execute your sparql requests

```
select distinct * where {  
  ?x ?y ?z  
} limit 25
```

Submit

showing 1 to 25 from 25 entities.

x	y	z
http://www.co-ode.org/roberts/travel.owl	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
http://www.co-ode.org/roberts/travel.owl	http://www.w3.org/2000/01/rdf-schema#comment	travel ontology
http://www.co-ode.org/roberts/travel.owl	http://www.w3.org/2000/01/rdf-schema#label	Travel
http://www.co-ode.org/roberts/travel.owl#definition_citation	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#AnnotationProperty

Geography ontology Home

Welcome to our geogrpahy ontology  
Here you can execute your sparql requests

```
prefix cd:<http://www.co-ode.org/roberts/travel.owl#>  
select ?country (count(?island) as ?no) where {  
  {?continent a cd:Continent  
  }  
  {  
    ?island cd:isDirectPartOf ?continent  
    FILTER(?continent != cd:europa )  
  }  
}
```

Submit

showing 1 to 29 from 29 entities.

country	no
http://www.co-ode.org/roberts/travel.owl#colombia	10
http://www.co-ode.org/roberts/travel.owl#cyprus	1
http://www.co-ode.org/roberts/travel.owl#puerto_rico	130

Figure 12 : Application web pour l'exécution des requêtes Sparql

### III. Interface de chatbot

L'interface du chatbot est celle de toutes les conversations de la plateforme Messenger de Facebook, il suffit à un utilisateur de se connecter à son compte Facebook et contacter l'agent via sa page.

#### A. Première conversation :

Dès que nous démarrons la conversation, le chatbot collecte quelques informations (Nom, genre).

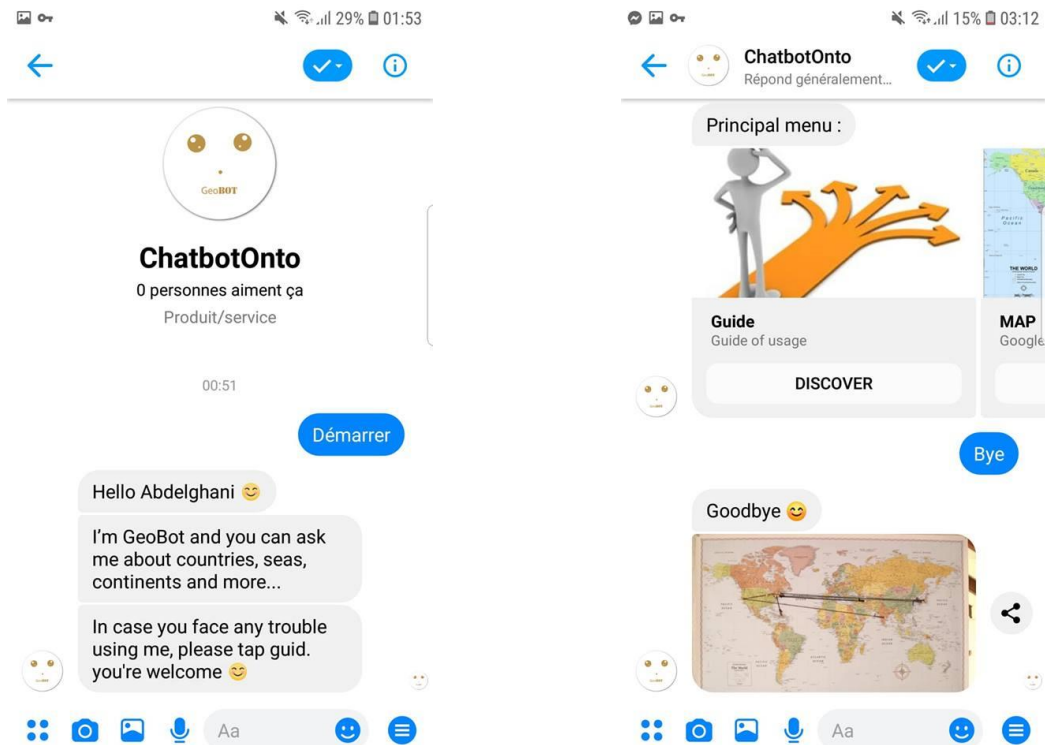


Figure 13 : Première conversation avec le chatbot



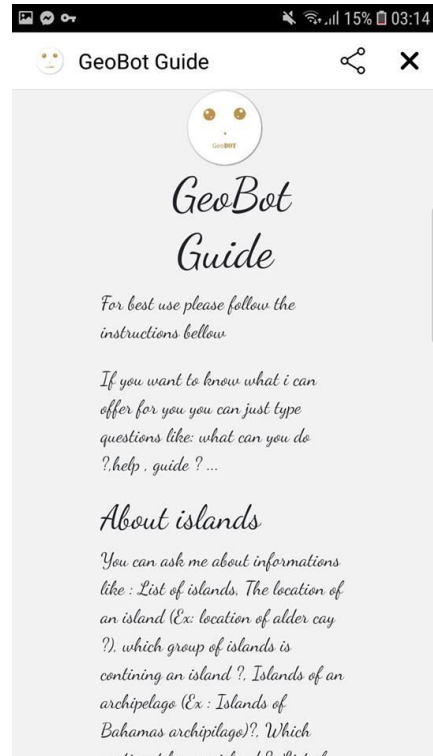
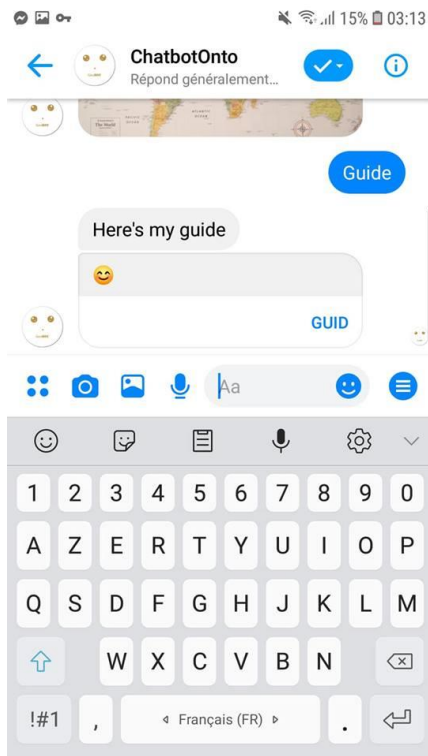


Figure 14 : Guide du chatbot

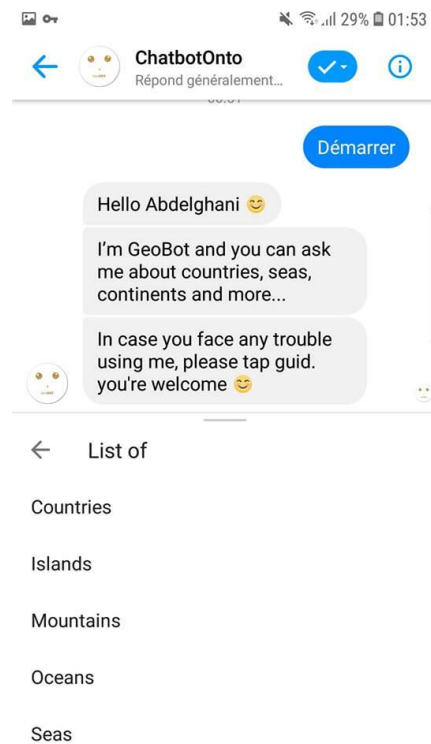


Figure 15 : Menu et sous menu

## B. A propos des « islands » :

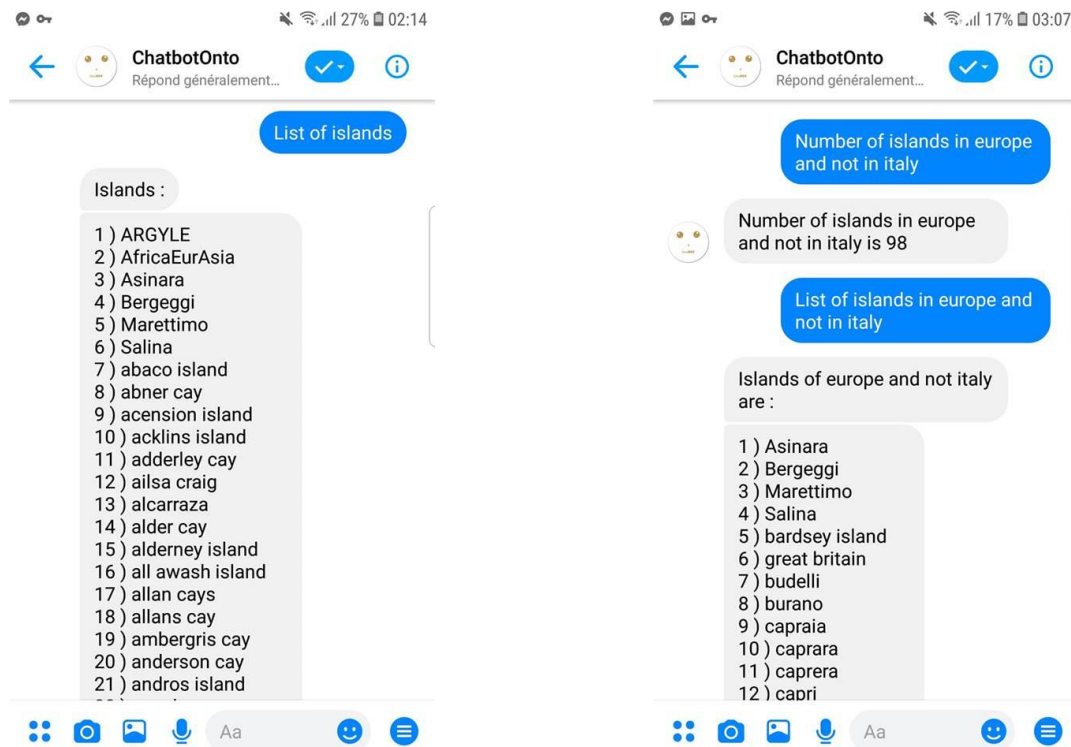


Figure 16 : Listes des « islands » et ceux en europe et pas en Italie

- **Requêtes Sparql :**

*#Liste des « islands »*

prefix cd:<<http://www.co-ode.org/roberts/travel.owl#>>

select ?island where {

  ?island a cd:Island

}

*#Nombre des « islands » en europe et pas en italie*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

select (count( distinct ?island) as ?c) where {

  ?country a cd:Country

  ?island a cd:Island

  ?island cd:isDirectPartOf cd:europe

  ?island cd:hasLocation ?country

  FILTER (?country != cd:scotland)

}

*#Liste des « islands » en europe et pas en italy*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

select distinct ?island where {

  ?country a cd:Country

  ?island a cd:Island

  ?island cd:isDirectPartOf cd:europe

  ?island cd:hasLocation ?country

  FILTER (?country != cd:scotland)

}

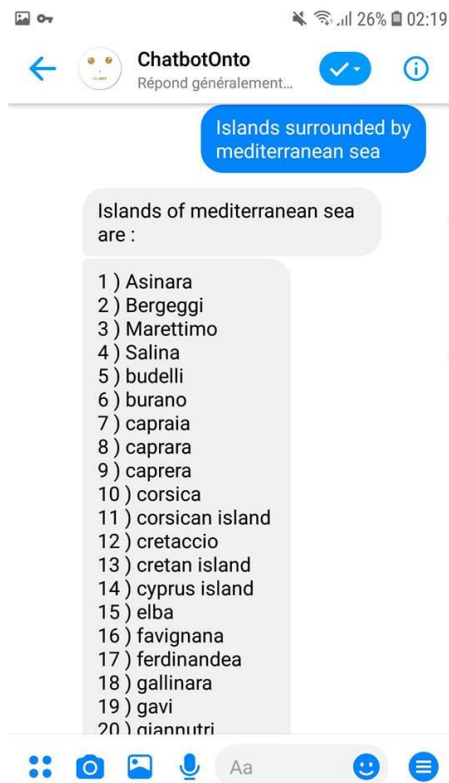


Figure 17 : Liste des « islands » et capital d'un pays

- **Requêtes Sparql :**

*#Liste des « islands » Mediterranean sea*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

```
select distinct ?island where {
  {?island a cd:Island
  filter (?country != cd:italy)}
  {?island cd:surroundedBy cd:mediterranean_sea }
}
```

*#Liste des « islands » en afrique*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

```
select ?island where {
  ?island cd:isDirectPartOf cd:africa
}
```

*#Capital d'un pays ou ses villes*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

select ?x where {

?x cd:isDirectPartOf cd:germany

}



Figure 18 : Requêtes sur les « islands »

- **Requête Sparql :**

*#Location de Alder Cay (island)*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

select ?location where {

cd:alder\_cay cd:hasLocation ?location

}

*#Le groupe auquel appartient Alder Cay (island)*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

```
select ?w where {  
  cd:alder_cay cd:isMemberIslandOf ?w  
}
```

*#Les « islands » de bahamas archipelago*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

```
select ?member where {  
  ?member cd:isMemberIslandOf cd:bahamas_archipelago  
}
```

*#Les « islands » entourés pas medeteranian sea et qui ne sont pas en italie*

prefix cd:<http://www.co-ode.org/roberts/travel.owl#>

```
select distinct ?island where {  
  {  
    ?country a cd:Country  
    ?island a cd:Island  
    ?island cd:hasLocation ?country  
    filter (?country != cd:italy)}  
  {  
    ?island cd:surroundedBy cd:mediterranean_sea  
  }  
}
```

## D. A propos des « Oceans » :



Figure 19 : Liste des oceans et auquel oceans appartient un « sea »

- **Requêtes Sparql :**

```
#Ocean d'un sea
```

```
prefix cd:<http://www.co-ode.org/roberts/travel.owl#>
```

```
select ?ocean where {
```

```
  cd:mediterranean_sea cd:isDirectPartOf ?ocean
```

```
}
```

## E. A propos des « Hills » :



Figure 20 : List des « hills » et location d'un « hill »

- **Requêtes Sparql :**

*#Liste des « islands »*

```
prefix cd:<http://www.co-ode.org/roberts/travel.owl#>
select ?hill where {
  ?hill a cd:Hill
}
```

*#Location d'un hill*

```
prefix cd:<http://www.co-ode.org/roberts/travel.owl#>
select ?loc where {
  cd:harting_hill cd:hasLocation ?loc
}
```



## F. A propos des « seas » :



Figure 21 : Liste des « seas »

- **Requêtes Sparql :**

```
#Liste des « seas »  
prefix cd:<http://www.co-ode.org/roberts/travel.owl#>  
select ?sea where {  
  ?sea a cd:Sea  
}
```

## G. A propos des « mountains » :

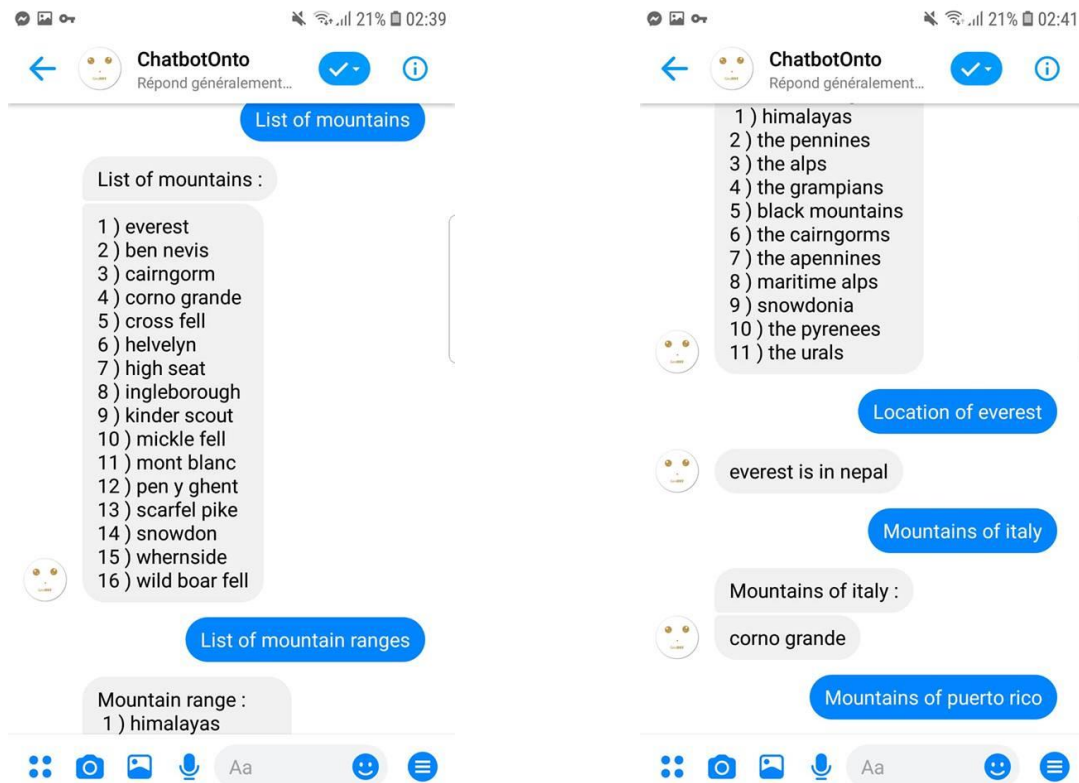


Figure 22 : Liste des montagnes et location d'une montagne

### • Requêtes Sparql :

*#Liste des « mountains »*

prefix cd:<<http://www.co-ode.org/roberts/travel.owl#>>

select ?mountain where {

  ?hill a cd:Mountain

}

*#Location d'une montagne*

prefix cd:<<http://www.co-ode.org/roberts/travel.owl#>>

select ?loc where {

  cd:Everest cd:hasLocation ?loc

}

## H. Non sens :

Le chatbot peut détecter si une phrase n'a pas de sens :

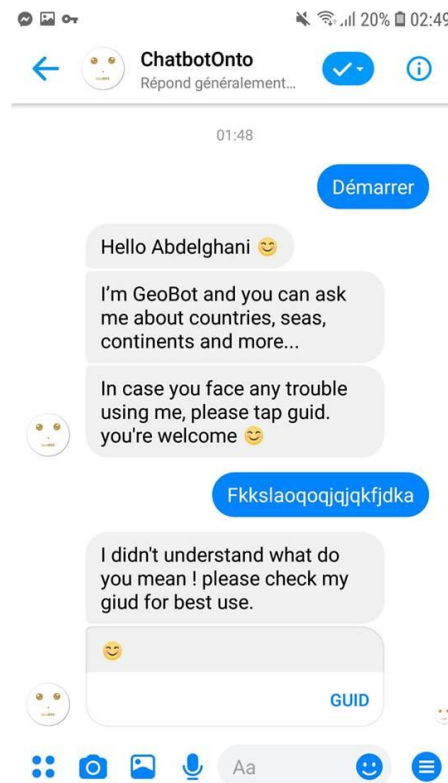


Figure 23 : Détection du non sens

- **Mode développeur :**

Le passage en mode développeur peut être posséder comme suit :

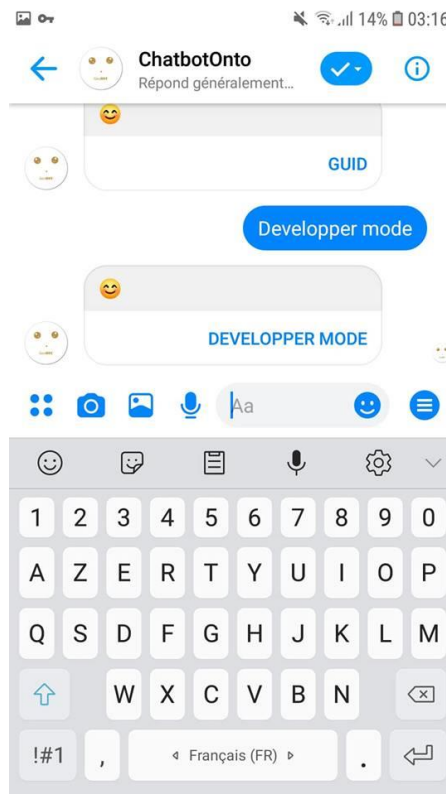


Figure 24 : Passage au mode développeur

## IV. Données analytique

La plateforme de développement de Facebook offre le service de consultation des données analytiques pour comprendre la manière dont les personnes utilisent notre App, mesurer le taux de fidélisation et suivre les conversations. Ci-dessous les résultats analytiques du chatbot pour le mois de janvier :

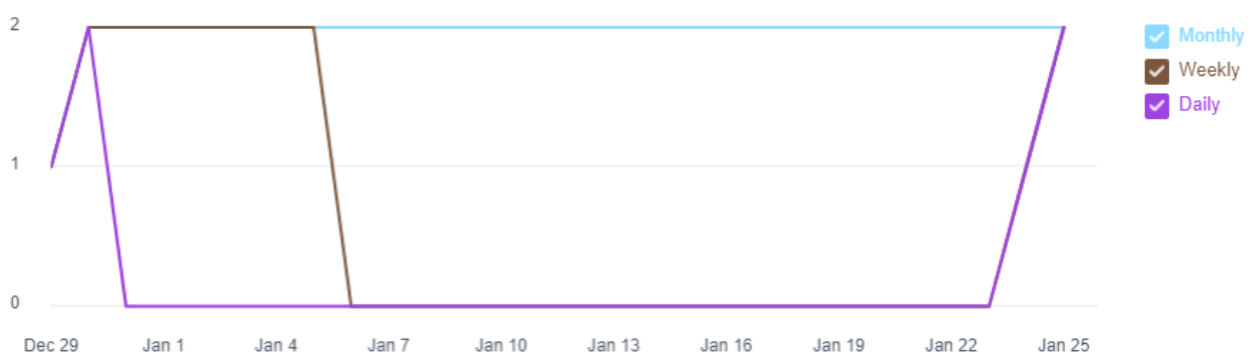


Figure 25 : Données analytiques du chatbot

# Conclusion générale

Dans ce projet, l'objectif était de concevoir et développer un chatbot intelligent en se basant sur une ontologie web.

Les contraintes confrontées au long du projet étaient principalement à cause des nouveautés des outils utilisés et le temps perdu pour la formation. Nous avons passé beaucoup de temps pour comprendre le langage et s'adapter aux outils utilisés.

L'intelligence artificielle est un univers en évolution continu, en effet plusieurs améliorations peuvent être rajoutées. Le chatbot peut commencer à apprendre tout seul sans interaction d'administrateur, il peut ainsi créer des nouvelles entités et améliorer son langage et utiliser plusieurs ontologies dans le web.

Ce projet nous a permis de découvrir de nouvelles approches de développement en ce qui concerne la programmation côté serveur, le web sémantique et nous a offert la chance d'acquérir des nouveaux termes d'intelligence artificielle et ses plateformes dédiées.

# Références

**Formation Udemy**, Disponible à l'adresse : <https://www.udemy.com/chatbots-chatbot-messenger-avec-witai-nodejs-et-heroku/learn/v4/overview>

**Emily Reese**, Utilisez des API REST dans vos projets web [en ligne], Mis à jour le : 20/06/2017.  
Disponible à l'adresse : <https://openclassrooms.com/courses/utilisez-des-api-rest-dans-vos-projets-web/>

**Facebook**, Utilisation de l'API Graph [en ligne]. Disponible à l'adresse : <https://developers.facebook.com/docs/graph-api/using-graph-api/>

**Facebook**, Référence concernant l'API Send [en ligne]. Disponible à l'adresse : <https://developers.facebook.com/docs/messenger-platform/reference/send-api/>

**Fueski**, Documentation [en ligne]. Disponible à l'adresse : [https://jena.apache.org/documentation/serving\\_data/](https://jena.apache.org/documentation/serving_data/)

**Tutoriel Sparql** [en ligne]. Disponible à l'adresse : <https://web-semantic.developpez.com/tutoriels/jena/arq/introduction-sparql/>

**Joël Ignasse**, Facebook convoque les bots pour ses nouvelles versions, 17/04/2016. Disponible à l'adresse : [https://www.sciencesetavenir.fr/high-tech/web/facebook-convoque-les-bots-pour-ses-nouvelles-versions\\_36833](https://www.sciencesetavenir.fr/high-tech/web/facebook-convoque-les-bots-pour-ses-nouvelles-versions_36833)

**Marc G Gauthier**, Gérer votre code avec Git et Github [en ligne], Mis à jour le : 12/12/2016.  
Disponible à l'adresse : <https://openclassrooms.com/courses/gerer-son-code-avec-git-et-github/>

**Mathieu Nebra**, Des applications ultra-rapides avec NodeJs [en ligne], Mis à jour le : 26/06/2018.  
Disponible à l'adresse : <https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/>

**Robin Coulet**, Avez-vous besoin d'un chatbot ? [en ligne], 22/09/2017.  
Disponible à l'adresse : <https://solutions.lesechos.fr/tech/c/avez-besoin-dun-chatbot-7150/>

**W3School**, Node.js Tutorial [en ligne]. Disponible à l'adresse : <https://www.w3schools.com/nodejs/>

**Wit.ai**, Get started with NodeJs [en ligne]. Disponible à l'adresse : <https://github.com/wit-ai/node-wit>

# Glossaire

Terme	Définition
NLP	Traitement de langage naturel
IA	Intelligence artificielle
SPARQL	Protocol and RDF Query Language
API	Interface de programmation d'application
Graph	API de facebook
V8	Moteur ultra rapide de Google