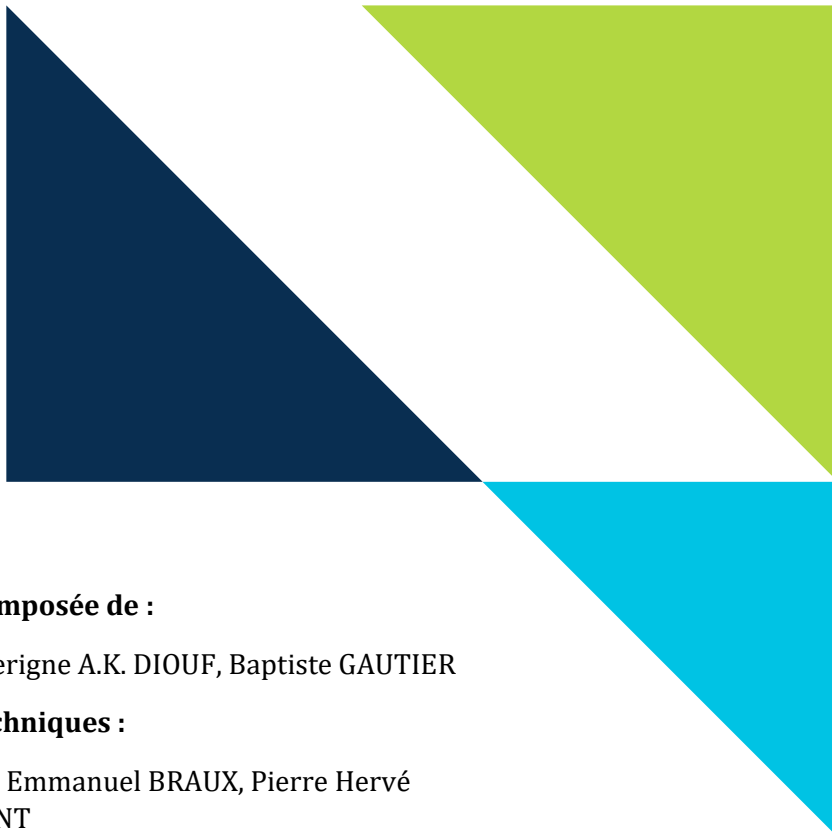


Manuel d'utilisateur projet N°60 :

Weather Chatbot

Le 28/06/2018



Equipe projet composée de :

Virgile BLUA, Xiaorui HUO, Driss OUSTI, Serigne A.K. DIOUF, Baptiste GAUTIER

Encadrants techniques :

Mathieu SIMONNET, Phillipe TANGUY, Emmanuel BRAUX, Pierre Hervé
VAILLANT

Client partenaire industriel :

Yann AMICE et Thomas LECHAUVE de Weather'n'co

Auteurs :

Virgile BLUA, Xiaorui HUO, Driss OUSTI, Serigne A.K. DIOUF, Baptiste GAUTIER

Destinataires :

Yann Amice, Thomas LECHAUVE (Weather'n'co)



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



1 Table des matières

2	Introduction :	2
3	Contenu	2
4	A installer	4
5	A prévoir	4
6	Comment démarrer	4

2 Introduction :

L'application Weather Chatbot met en oeuvre un agent conversationnel capable de répondre de façon automatique et instantanée à des question météorologiques envoyées à la page Facebook de Weather'n'co à travers la plateforme de messagerie Messenger. Ce produit développé par cinq étudiants de l'IMT Atlantique lors d'un projet pédagogique est la propriété du service partenaire et client dans ce projet (Weather'n'co) représenté par M.Yann AMICE. Il constitue une première version d'un chatbot intelligent capable de substituer parfaitement le propriétaire à l'exercice d'annonce du bulletin météo sur plateforme de messagerie.

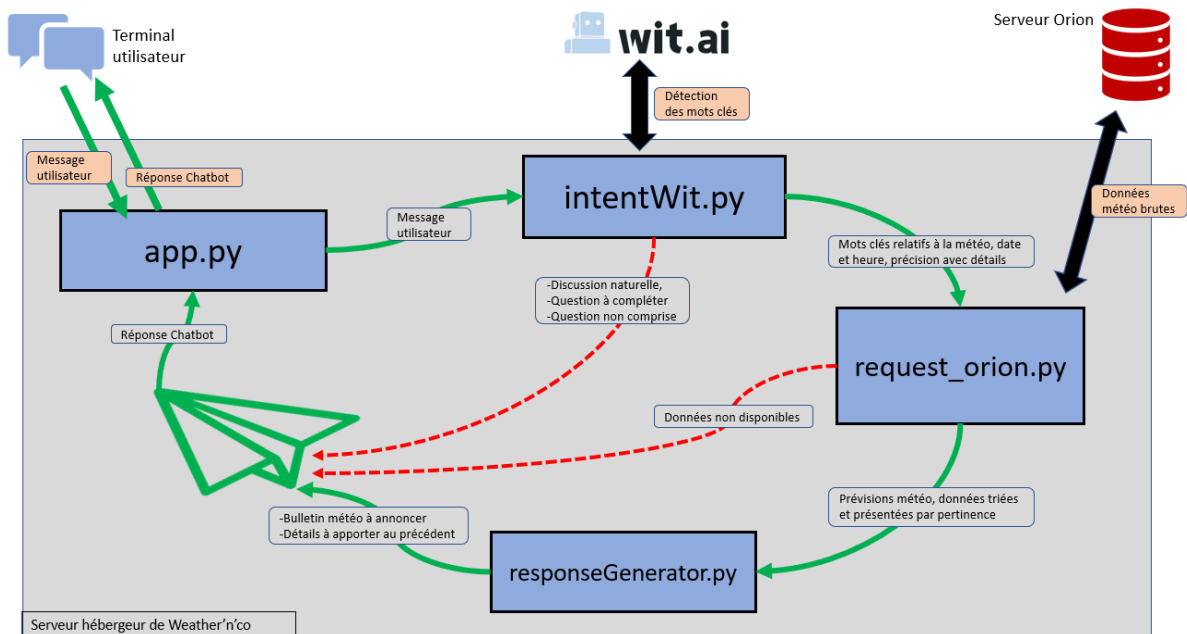
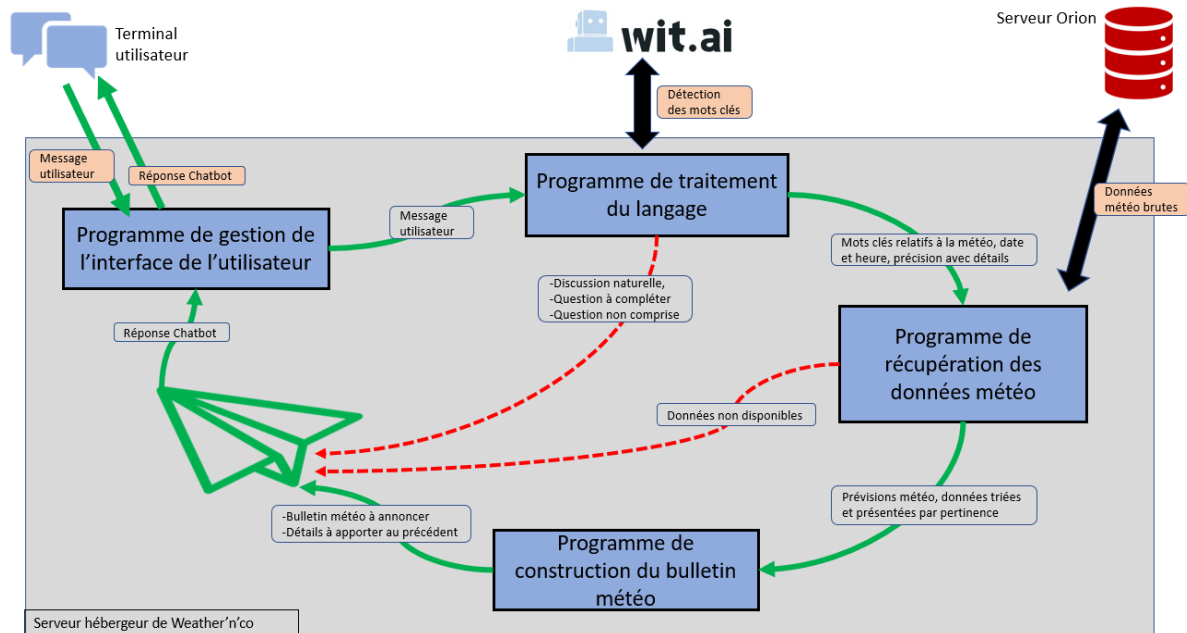
Ce manuel d'utilisation fait suite aux travaux réalisés et présenté dans le rapport technique de fin de projet est destiné au client Weather'n'co. Il a pour objet d'expliquer à celui-ci comment il pourra utiliser le code source fourni pour déployer sur sa propre page l'application développée pour lui et les différents outils à prévoir à cet effet.

3 Contenu

Le fichier archive transmis au client après la soutenance du projet contient:

- **app.py**: c'est le point d'entrée dans l'application. Il permet une authentification réciproque entre l'application sur la plateforme Développeur Facebook et le serveur hébergeur. Il réceptionne par ailleurs les messages transmis à l'application et envoie les réponses conçues par celle-ci vers la page Facebook.
- **intentWit.py**: il permet de faire des requetes vers la plateforme Wit.ai et récupérer les mots-clés météo contenus dans le message. Traite le message selon les mots-clés détectés, structure ceux-ci et les envoie vers le fichier request_orion.py si toutes les informations nécessaires (paramètres météo et information temporelle) y sont. Il peut synthétiser une réponse directe à son niveau si le message relève d'une discussion générale (sans météo, salutations par exemple), s'il doit être complété (information temporelle manquante par exemple) ou s'il n'est pas compris par l'application.
- **request_orion.py**: il reçoit en entrée les mots clés-triés dans le programme précédent, fait une requete vers l'API du client (<https://data2.weathernco.com/point>) avec les variables météo correspondantes. Les données brutes récupérées seront synthétisées et envoyées vers le programme responseGenerator.py pour construire le bulletin météo demandé. Il peut aussi synthétiser une réponse directe en cas de non disponibilités des données sur l'API du client (par exemple, une question météo qui porte sur une date très éloignée où il n'y a pas encore de prévisions disponibles ou simplement en cas de panne de l'API).
- **responseGenerator.py** : C'est l'étape ultime de la construction du bulletin météo. Il élabore un bulletin concis et compréhensible à partir des données provenant du programme précédent. Si besoin est, il donnera les détails du bulletin annoncé par heure avec des indications sur les niveaux d'alerte.
- **dateconverter.py**: C'est un petit algorithme utilisé par le programme responseGenerator pour formater les dates. Ce qui permet de connaître le nom jour sur lequel porte la question et donner une réponse intelligente.

- ***data_fetch_protocol.py***: c'est le mini-protocole en annexe du rapport technique. Elle présente comment sont formatés les données échangées entre les différents programmes
- ***install.txt***: il contient toutes les librairies et modules à installer pour faire marcher le programme
- ***lancementChatBot.txt***: il indique les étapes à suivre pour lancer l'application
- ***bot.py***: c'est un fichier modifié de la librairie pymessenger (voir la section suivante)



4 A installer

Le produit est développé sous Python 3 et l'ensemble des librairies utilisées sont installées avec pip3. L'ensemble des modules et librairies à installer son dans le fichier install.txt. Suivre les étapes suivantes pour mettre en oeuvre le programme dans le serveur hébergeur:

- Se procurer de Python3 (lien)
- installer les librairies avec pip3 (il faudra les droits d'administrateur pour le système d'exploitation Linux utilisé)
- remplacer le fichier bot.py dans la librairie pymessenger par celui dans l'archive. Il faut les droits d'administrateur

5 A prévoir

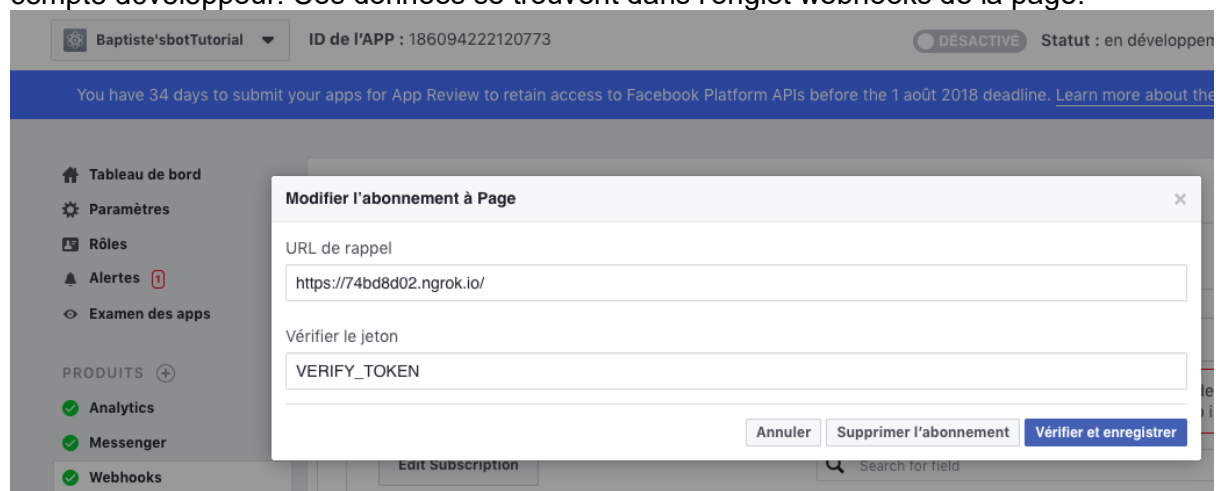
Avant la mise en marche du produit, il faut disposer:

- d'une page Facebook avec les droits d'administrateur (<https://facebook.com>): partie visible de l'application. C'est vers cette page que les utilisateurs vont envoyer leurs questions comme dans une conversation normale avec une page Facebook.
- d'un compte développeur Facebook (<https://developers.facebook.com>): il sert à créer l'application et configurer le webhook (l'adresse du serveur vers lequel elle redirige les messages reçus depuis la page Facebook et les modalités de la redirection)
- d'un compte wit (<https://wit.ai>): La plateforme nous permet de faire du traitement de langage naturel, elle reçoit le message de l'utilisateur et renvoie un dictionnaire qui contient les mots-clés sous forme d'entités.
- d'un serveur http sécurisé: c'est le serveur sur lequel tourne l'application. Elle reçoit tous les messages envoyés à la page et envoie vers la page la réponse fournie par l'application.

6 Comment démarrer

6.1 configurer le webhook (échange token)

Pour que la plateforme messenger puisse communiquer avec le bot, il faut lier l'application sur Facebook developer à l'adresse publique du point d'entrée du bot. Il faut également ajouter une clé secrète pour vérifier qu'il s'agit bien du bonne url de rappel. Ici, la clé choisie est "VERIFY_TOKEN". Il faut par conséquent avoir un compte Facebook pour avoir un compte développeur. Ces données se trouvent dans l'onglet webhooks de la page.



The screenshot shows the Facebook Developer console interface. At the top, the app name is 'Baptiste'sbotTutorial' and the ID is '186094222120773'. A status bar indicates 'DÉSACTIVÉ' and 'Statut: en développement'. A blue banner at the top states: 'You have 34 days to submit your apps for App Review to retain access to Facebook Platform APIs before the 1 août 2018 deadline. Learn more about the deadline.' On the left sidebar, there are links for 'Tableau de bord', 'Paramètres', 'Rôles', 'Alertes', 'Examen des apps', and 'PRODUITS'. Under 'PRODUITS', 'Webhooks' is selected. A modal dialog titled 'Modifier l'abonnement à Page' is open, showing the 'URL de rappel' as 'https://74bd8d02.ngrok.io/' and the 'Vérifier le jeton' as 'VERIFY_TOKEN'. At the bottom of the dialog are buttons for 'Annuler', 'Supprimer l'abonnement', and 'Vérifier et enregistrer'.

6.2 Lier l'app à messenger et à la page

Pour que le bot réponde bien à la page facebook associée. Il faut générer un token associé à la page. Le token doit être identique à celui présent dans app.py sous le nom de "ACCESS_TOKEN"



```
ACCESS_TOKEN =  
'EAACpQHABF0UBALuobvuAqzFWeyEL3jV7eGoF3ZB9iy6P77cMGw4BqDYwgnGeWzqzWZAjNoRTZB9tEpXBv21ZB0RPC9zxx1nDIRXwaksNGXbxFnAokzKHoFebE7jzcNes1h5FJP5eZAFHdbyhPvsSIck0MYmRUCsY9pACVGMPB2QZDZD'  
VERIFY_TOKEN = 'VERIFY_TOKEN'
```

6.3 Mettre le bon lien https

L'adresse publique du point d'entrée du bot doit être un lien https. Cela fait partie des obligations de Facebook

6.4 Configurer Wit (définir les entités et récupérer le token) :

- Après avoir créé un compte sur Wit, on crée une application, on choisit la langue et le fuseau horaire.
- le menu Understanding, entraîner le modèle en ajoutant des expressions et en définissant des entités associées à ces expressions.



- Pour intégrer Wit.ai dans notre application, la plateforme fournit une API et un access token qui permet de faire de l'authentification. Mettre l'access token dans la fonction `get_wit_resp` du fichier `intentWit.py`, dans le constructeur qui permet d'initialiser une instance de Wit.
- On envoie le message de l'utilisateur à la plateforme via l'API et on reçoit un dictionnaire en réponse.

- ➔ Avoir tous les programmes dans le même dossier
- ➔ Lancer app.py