

# TP - Cloud Computing

Le but du TP est de créer un ChatBot sur l'équipe progres-upmc sur Slack : <https://progres-upmc.slack.com>.

1. Commencez par créer un compte <https://progres-upmc.slack.com/signup> en utilisant votre adresse en @etu.upmc.fr  
On pourra profiter de Slack pour poser des questions sur le cours/TP/Projet.
2. Ensuite créez un compte Bot <https://my.slack.com/services/new/bot>. Choisissez un nom se terminant par \_bot .

Une documentation des bot de slack est disponible ici: <https://api.slack.com/bot-users>.

Créez un compte (si vous n'en possédez pas déjà un) sur <https://github.com>, <https://travis-ci.org> (en utilisant votre compte github), et <http://cloudsight.ai> .

## Base du Projet

1. Créez un nouveau projet nodejs dans un nouveau dossier avec la commande `npm init` .
2. Initialisez un repository git local avec la commande `git init` .
3. Créez un nouveau repository sur github.com et l'ajouter en tant que remote origin a votre repository local `git remote add origin URL`
4. installer les packages `mocha` `slackbots` `axios`
5. Ajouter un script `test` (qui appelle la commande `mocha` ) et un script `start` qui execute la commande `node` sur votre fichier javascript principal.
6. Ecrire un fichier `test.js` simple.
7. Ajouter un fichier .travis.yml qui permet de tester automatiquement votre projet à l'aide de travis.
8. Ajoutez tous les fichiers (sauf le dossier `node_modules` ) au gestionnaire de version git (commande `git add FICHIER` ).
9. Commitez et poussez les changement sur github.
10. En ce servant de la documentation du module `slackbots` qui se trouve ici: <https://github.com/mishk0/slack-bot-api>.  
Ecrire un bot dans votre fichier javascript principal qui vous envoie un message à vous (compte slack). **Le token d'identification du bot doit se trouver dans une variable d'environnement et ne doit figurer dans aucun fichier.**

## Organisation du Projet

Pour organiser correctement le projet, le but est de placer toutes les fonction du bot dans un fichier `bot.js` .

et le fichier `bot.js` aura cette forme:

```
const SlackBot = require('slackbots');

module.exports = function(params) {
  this.bot = null;
  this.connect = function() {
    this.bot = new SlackBot(params);
    this.bot.on('start', this.onStart);
    this.bot.on('message', this.onEvent);
  }
  this.onStart = function() {
    // Le bot est connecté
  }
  this.onEvent = function(event) {
    //reception d'un message
    console.log(event);
  }
}
```

```
}  
}
```

Modifiez votre fichier javascript principal pour qu'il utilise votre nouveau module.

1. La méthode `onEvent` est appelée pour tout type d'événement. Déléguez le traitement des événement de type `message` à une fonction `onMessage(message)` que vous devez créer.
2. Dans la fonction `onStart` ajouter les lignes suivantes, qui permettent d'avoir un attribut `user` sur l'objet `bot`, qui contient les informations de votre bot:

```
this.bot.getUser(this.bot.self.name).then(function(user) {  
    this.user = user;  
});
```

à l'aide de ces informations contenues dans l'attribut `bot.user`, vérifiez qu'un message provenant de votre propre bot ne soit pas délégué à la fonction `onMessage`.

1. Dans la fonction `onMessage`, distinguer le cas d'un message provenant d'un autre bot du cas d'un message provenant d'un utilisateur réel et à chaque fois qu'un message est reçu en provenance d'un humain, envoyez une réponse quelconque.
2. Dans le cas d'un message reçu en provenance d'un bot, envoyer une autre réponse **après un délai de deux secondes**.

## Utilisation d'API

1. À intervalle régulier (pas trop souvent), utiliser l'API du site [www.chucknorrisfacts.fr](http://www.chucknorrisfacts.fr) afin d'afficher une blague aléatoire.
2. Lorsqu'un utilisateur envoie une image, utiliser l'API de CloudSight pour détecter le contenu de l'image et envoyer un message disant que vous aimez ce type de contenu.
3. Regarder dans la documentation de Slack afin que votre bot ajoute un cœur (une réaction `❤️`) sur le message contenant l'image.
4. Utiliser l'API d'IMGUR pour chercher des images similaires et poster une image similaire.

## Réponse à une question

1. Si un utilisateur demande quel temps fera-t'il demain, lui répondre en utilisant une API météo quelconque.
2. Prendre en charge d'autres types de questions.

## Déploiement

Créez un compte sur <https://heroku.com> et déployez votre bot sur héroku.