Ce travail pratique compte pour 25 points de la note finale du cours. Il est la suite du travail pratique 1, si vous n’avez pas réussi à faire fonctionner votre TP1, parlez-en au prof qui vous aidera à démarrer votre TP2.

**Section I : (5 points) Documentation de votre API et création du projet dans GitHub**

Vous devez ajouter à votre travail pratique 1, une route vers la documentation et l’aide en ligne (de votre API) qui décrit comment utiliser votre API. Cette route utilise “/api/aide” de votre site Web. Cette documentation doit fournir tous les détails concernant l’utilisation et fournir des exemples au lecteur qui veut l’utiliser. Voici un exemple (un peu trop complexe peut-être, car beaucoup plus de fonctionnalités) d’une documentation d’un API REST “https://docs.github.com/en/rest”.

Vous allez créer un nouveau projet dans GitHub en utilisant comme pour votre travail précédent votre nom et TP2 comme par exemple SERVICES\_PIL\_TP2. Faites une copie de votre TP1 pour démarrer votre TP2. Il est préférable de copier seulement les fichiers que vous allez avoir de besoin et pas tout le dossier du TP1 pour démarrer un nouveau projet dans un dossier vide dans lequel vous allez copier les fichiers.

N’oubliez pas de me fournir vos informations GitHub:

1. Votre répertoire gitHub comme ceci :

git remote add origin https://github.com/alain-clg/SERVICES\_PIL\_TP2.git

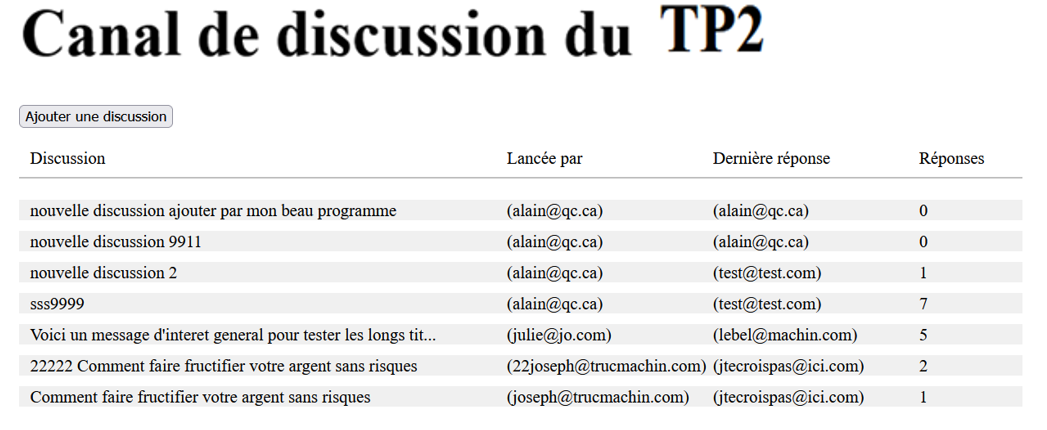
1. Votre token d’accès privé comme ceci:

ghp\_EbUOdXCBf6tDYG0pknabc123911klBT14IhN3R

**Section II : (20 points) Page Web pour tester les autres fonctions de votre API**

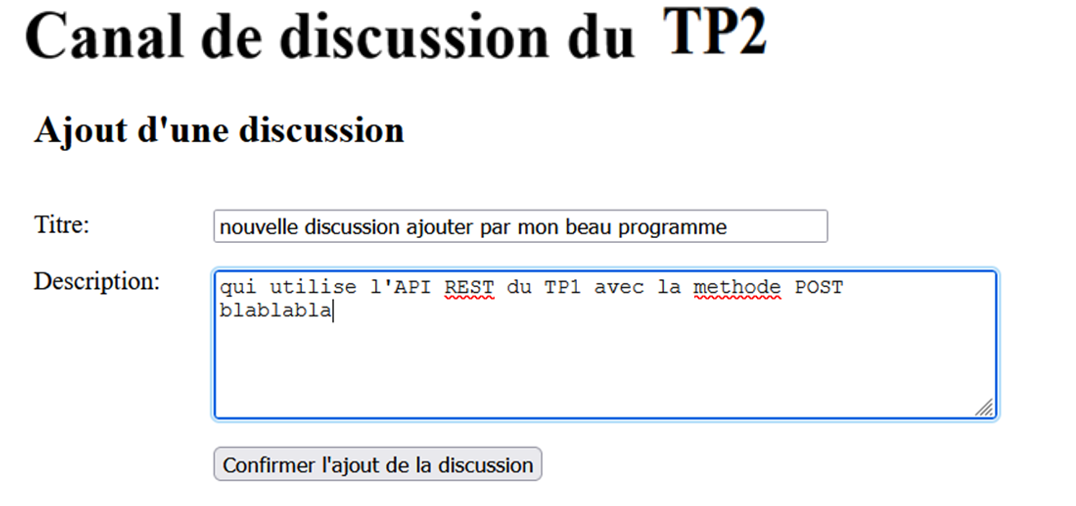
Pour cette partie, vous devrez faire une page Web index.html permettant l’interaction entre l’utilisateur et votre API. Ce programme utilisera la méthode de votre choix pour faire appel à l’API et présenter un contenu dynamique à votre utilisateur. Ce programme doit utiliser JavaScript et JQuery afin d’afficher le même canal de discussion que le TP1 et faire appels à toutes les fonctionnalités de votre API, entre autres, dans ce travail nous allons ajouter l’utilisation des méthodes « POST », « PUT » et « DELETE » de votre API. Évalué sur 9 points pour l’ajout d’une discussion vous allez utiliser « POST ». La suppression d’une discussion utilise « DELETE » et sera évaluée sur 6 points. Pour 5 points vous allez ajouter une fonction de modification des discussions qui sera utile pour ajouter une réponse à un message de discussion.

Comme pour le TP1, vous commencez par afficher le contenu de la collection messages de la base de données dans Atlas. Vous devez vous assurer que l’affichage sera trié en ordre décroissant de date, soit le plus récent en premier. Vous ajoutez en haut de la page un bouton « Ajouter une discussion » qui fera afficher un petit formulaire pour saisir le message contenu dans la discussion :

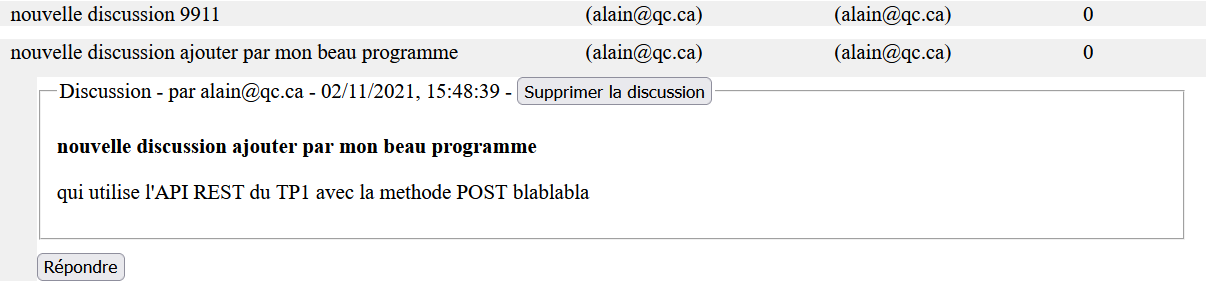


**(9 points) Ajout d’une discussion**

Voici le contenu du formulaire pour ajouter une nouvelle discussion. Pour les champs qui n’apparaissent pas dans l’écran suivant, vous pouvez tout simplement mettre des valeurs fixes (comme par exemple, auteur : jo@toto.com ou langue : « francais ». Pour l’objet commentaires il faut tout simplement le laisser vide soit : [ ];



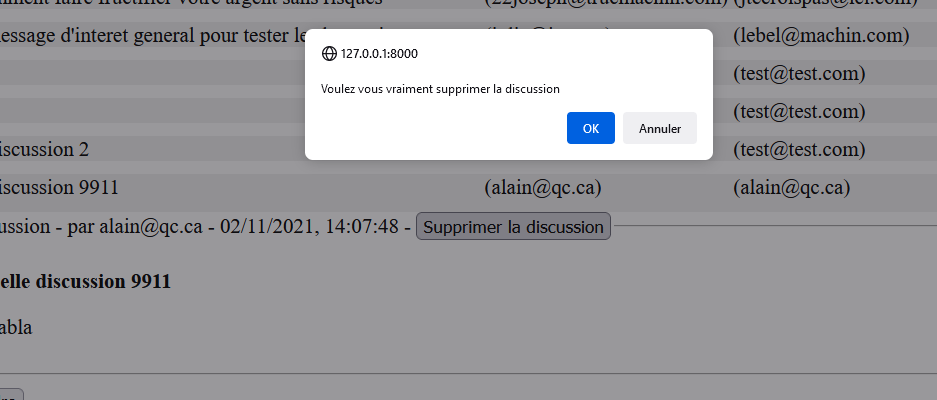
Lorsque l’utilisateur clique sur le bouton confirmer l’ajout de la discussion, il faut l’ajouter dans la base de données. Cette nouvelle discussion apparait au début de la liste par la suite…



En cliquant sur cette nouvelle discussion on obtient les détails de la discussion (avec l’auteur et la date comme dans le TP1). Deux boutons sont ajoutés par rapport au TP1, « Supprimer la discussion » et « Répondre ».

**(6 points) Suppression d’une discussion**

Le bouton « supprimer la discussion » permet de faire appel à la méthode « DELETE » de l’API, mais seulement après avoir confirmé en utilisant la fonction confirm de JavaScript :



**(5 points) Ajout d’une réponse**

Le bouton, « Répondre » permet d’ajouter une réponse dans la liste des commentaires de la discussion (dans l’objet « Commentaires »).



Une fois confirmer la réponse est ajoutée dans la liste et un appel à l’API en utilisant la méthode « PUT » est fait pour modifier le message (la discussion) dans la collection de la BD. Ainsi à chaque nouvelle réponse le message au complet (auquel on ajoute la nouvelle réponse) est renvoyé à la BD en utilisant PUT.