Ce travail pratique compte pour 25 points de la note finale du cours. Pour les deux premières sections du travail référez-vous au document TP1 partie 1.

**Section I : (3 points) Création du projet dans gitHub et Atlas**

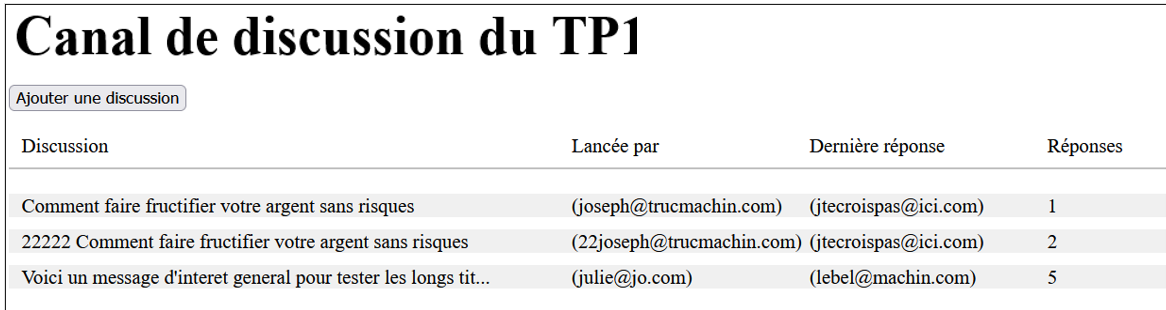
**Section II : (12 points) Projet nodeJS de services Web**

**Section III : (10 points) Ajout d’une page Web permettant d’utiliser l’API**

Pour cette partie, vous devrez faire une page Web 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 afin de faire appels à la méthode GET de votre API écrite à la section II.

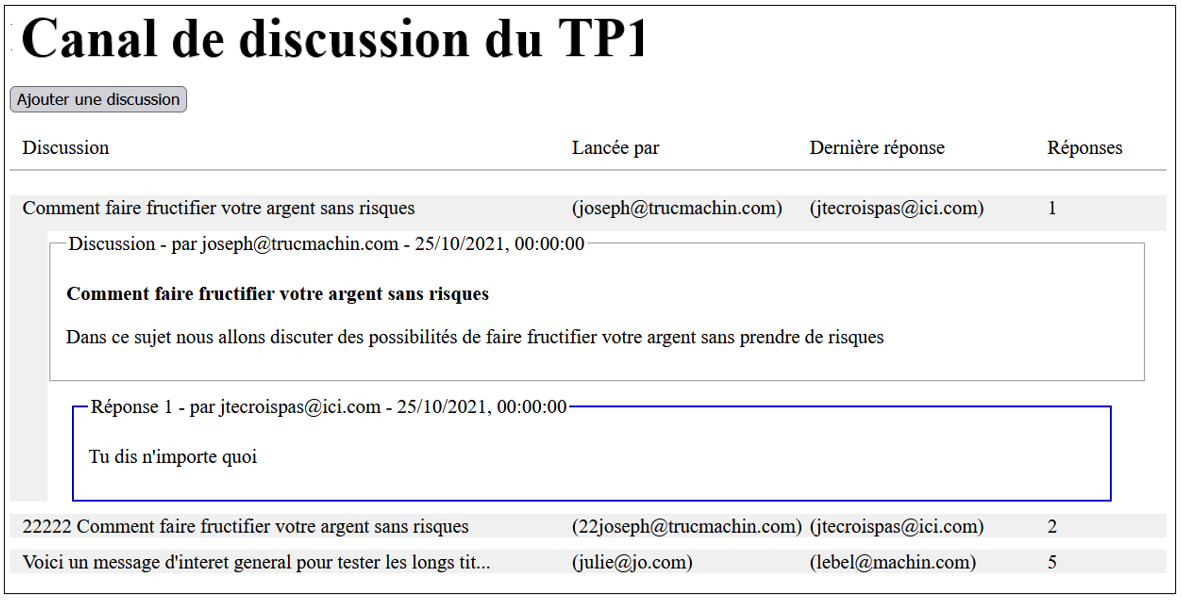
Vous devez fournir une page Web complète sous forme d’un forum qui présente le contenu de la base de données. Nous allons nous intéresser principalement à la récupération de données à partir du GET de notre API. Vous pouvez télécharger vos données avec fetch et les récupérer afin de les ajouter dans votre page Web en utilisant DOM et JQuery.

Votre interface sera simple et facile à utiliser. Vous chargez le contenu de la collection messages sur la page Web comme ceci :`

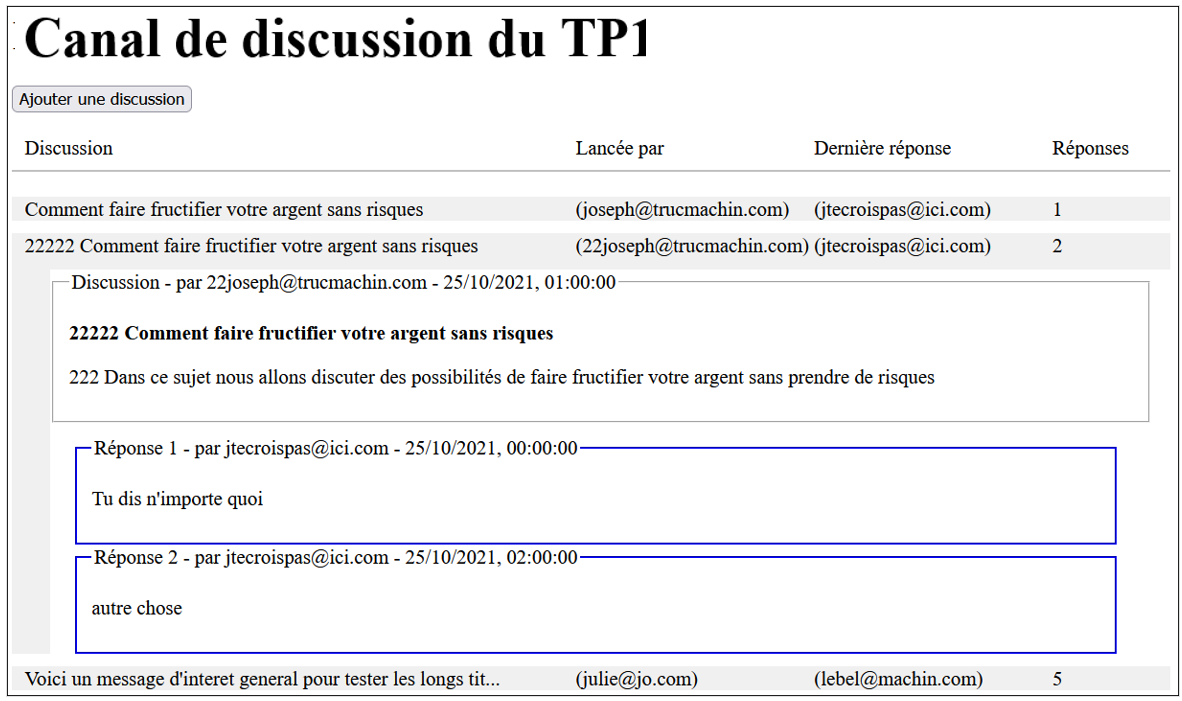


Le bouton « Ajouter une discussion » ne sera pas fonctionnel pour ce travail (il ne fait rien). La première colonne présente les titres, la seconde l’auteur du message (discussion) la troisième provient de la liste des commentaires (réponses) c’est l’auteur du dernier commentaire (si pas de commentaires du tout, on prend l’auteur du message initial) alors que la 4e et dernière colonne présente le nombre de réponses présentes (nombre de commentaires de la liste).

Lorsque l’utilisateur clique sur une ligne, la discussion correspondante s’ouvre comme ceci (après avoir cliquez sur la première ligne (première discussion) :



La discussion correspondante s’affiche avec l’auteur et la date en utilisant la légende d’un fieldset alors que le titre en gras et la description se trouvent à l’intérieur du fieldset. Par la suite chaque réponse apparaît dans son propre fieldset (dont la bordure est de couleur bleue) avec le numéro de la réponse (un compteur), l’auteur et la date dans la légende et le commentaire à l’intérieur du fieldset. Voici un exemple lorsque je clique sur la ligne de la seconde discussion :



Et dernière saisie d’écran, lorsque je clique sur la 3e discussion :



Voici le contenu du 3e document (objet) de la collection messages :

{

"\_id": "999",

"titre": "Voici un message d'interet general pour tester les longs titres qui depasse parfois les espaces prevus par les applications",

"auteur": "julie@jo.com",

"description": "courte description",

"langue": "français",

"date": "2021-10-22T05:00:00.000Z",

"commentaires": [

{

"commentaire": "ok voici une 1ere reponse",

"auteur": "bob@ici.com",

"date": "2021-10-24T04:00:00.000Z"

},

{

"commentaire": "ok voici une 2e reponse",

"auteur": "roger@ici.com",

"date": "2021-10-24T12:00:00.000Z"

},

{

"commentaire": "autre chose",

"auteur": "luc@ici.com",

"date": "2021-10-25T06:00:00.000Z"

},

{

"commentaire": "ok voici encore une reponse",

"auteur": "paul@truc.com",

"date": "2021-10-26T17:00:00.000Z"

},

{

"commentaire": "ok voici une 5e reponse",

"auteur": "lebel@machin.com",

"date": "2021-10-26T19:00:00.000Z"

}

]

}

Vous pouvez ajouter autant de messages que vous voulez afin de tester votre travail pratique convenablement.