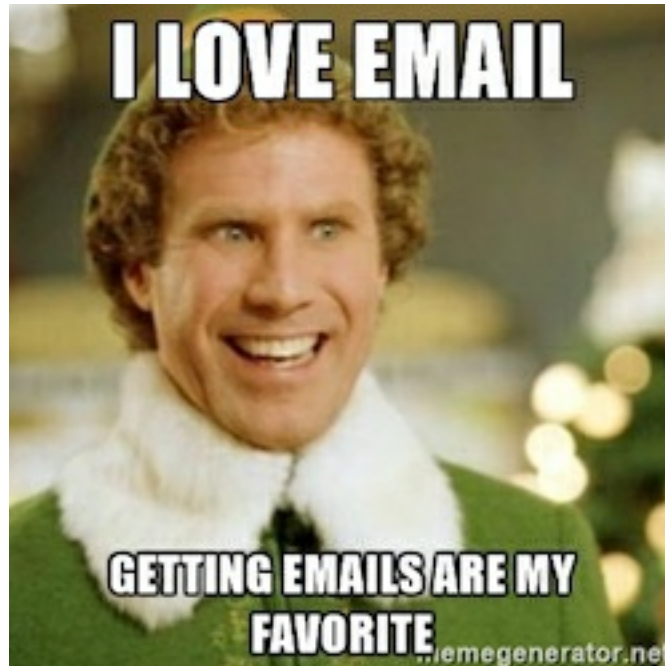


# TP 6 : Envoie d'email

---



Dans le TP de cette semaine, vous allez voir comment envoyer un email à l'utilisateur selon les cas suivants :

- Lors de l'ajout pour lui donner ses informations de login et de mot de passe
- Lors de la modification si le login ou le mot de passe ont changé (juste un email lui disant que ses informations ont changées)
- Après avoir lancé une requête de récupération de mot de passe : Effectivement, vu que les mots de passe sont cryptés en BDD, vous devrez être en mesure de changer uniquement celui-ci via une route prévue à cet effet.
- Rajouter la route `/auth` prenant en `POST` les paramètres login/password et retournant `{ msg : 'ok' }` si les informations sont bonnes et `{ msg : 'ko' }` si les informations sont incorrectes.

Pour réaliser ces emails, vous allez utiliser les modules suivants :

- `nodemailer` : library permettant d'envoyer facilement un ou plusieurs emails au travers de Node.JS
- `mailgen` : Moteur de templating d'email compatible tout client (ou presque).

Pensez à utiliser les services pour l'envoi de mail.

**Attention** : Lors de l'édition de l'utilisateur (route `PUT` ), si jamais le mot de passe n'est pas saisi ou bien s'il s'agit du mot de passe crypté, veillez à ne pas le recrypter sous peine de vous retrouver avec un résultat invalide en base de donnée.

**Attention** : Penser la partie d'envoi d'email (qui devrait être une fonction propre) réutilisable facilement et configurable via ses paramètres d'entrée vu que vous devrez en faire un service hapi indépendant au dernier TP.

## Compléments

---

Niveau organisation du code : Une route pour une partie métier. Par exemple à partir de ce TP vous devriez avoir les routes suivants (a peu de chose près) :

- users
- auth
- default

Vous devrez aussi pensez a mettre votre code dans des services, je suggère d'avoir plusieurs services:

- users
- mail
- etc..