

SnapPic - Part 3 / Consignes

Pour cette toute dernière partie de SnapPic, l'objectif est de gérer l'envoi de fichier des photos prises par l'application vers un backend chargé de l'upload auprès d'un service de stockage de médias tel que Cloudinary.



Environnement de travail

- 👉 Ajoutes un dossier **backend** à l'intérieur de ton dossier snap-pic.
- 👉 Mets en place un setup express de base avec la commande **express --no-view --git ./**
- 👉 Lances ton back avec la commande **nodemon**



Coté Backend

- 👉 Crées une route POST /upload recevant un fichier via req.files.photoFromFront et héberges cette image chez Cloudinary en suivant les étapes décrites dans le cours.

Exemple de réponse de la route POST /upload :

```
{  
  "result": true,  
  "url": "https://res.cloudinary.com/***.jpg",  
}
```



Coté Frontend

- 👉 Dans le composant SnapScreen, modifies la fonction takePicture chargée de la prise de photo afin de l'envoyer vers la route POST /upload.

L'URL à enregistrer dans le store Redux (pour l'afficher dans la galerie) devra être celle renvoyée de Cloudinary par le backend plutôt que celle renvoyée par la méthode takePictureAsync.

- 👉 Rends les informations de l'utilisateur (email et photos) persistantes via le module redux-persist afin de récupérer sa galerie et éviter de renseigner son email à chaque rechargement de l'application.

⚠️ **Attention** : certaines étapes de configuration de redux-persist devront être adaptées pour fonctionner avec React Native.



optionnel : Mets en place sur le screen GalleryScreen la possibilité de se logout.

Bonne chance 🧑