# Procédure Nuxt 3 – Création utilisateur + upload photo/CNI

## Récupérer un utilisateur

* GET: `/api/users/{id}`

**Réponse :**

|  |
| --- |
| {      "id": 1,      "email": "kuitoange@gmail.com",      "password": **null**,      "nom": "Kuito",      "prenom": "Ange",      "portable": "690123456",      "role": "CLIENT",      "statut": **true**,      "photo": **null**,      "cni": **null**  } |

**Frontend :**

NB: Les champs `photo` et `cni` dans la réponse sont des URLs directes.

* Exemple Nuxt 3 :

const { data: user, error } = await useFetch(`/api/users/${userId}`)

// user.value.photo et user.value.cni sont les URLs

## Créer l’utilisateur

* Fais une requête POST sur `/api/users` avec les infos de base (email, mot de passe, nom, prénom, portable…).
* Ne mets pas la photo ni la CNI dans cette requête.
* Récupère l’id de l’utilisateur dans la réponse.
* Exemple Nuxt 3 (Composition API, script setup)

|  |
| --- |
| const { data: user, error } = await useFetch('/api/users', {  method: 'POST',  body: {  email: 'client@exemple.com',  password: 'motdepasse',  nom: 'Doe',  prenom: 'John',  portable: '690000000'  }  })  // user.value.id => à utiliser pour l’upload des médias |

## Uploader la photo de profil et/ou la CNI

* Pour chaque média (photo ou CNI), fais un POST sur `/api/media/upload` en `multipart/form-data` :

- `file` : le fichier à envoyer

- `entityType` : `USER`

- `entityId` : l’id du user récupéré à l’étape précédente

- `description` : `"photo"` pour la photo de profil, `"cni"` pour la carte d’identité

* Exemple Nuxt 3 (upload d’un fichier) :

const uploadMedia = async (file, userId, description) => {

const formData = new FormData()

formData.append('file', file)

formData.append('entityType', 'USER')

formData.append('entityId', userId)

formData.append('description', description) // 'photo' ou 'cni'

const { data, error } = await useFetch('/api/media/upload', {

method: 'POST',

body: formData

})

return data.value // contient l’id et l’URL du média

}

## Remplacer ou supprimer un média

- Pour remplacer : refais un upload avec la même logique.

- Pour supprimer : fais un DELETE `/api/media/{mediaId}`.

* Exemple suppression :

await useFetch(`/api/media/${mediaId}`, { method: 'DELETE' })

## Résumé du workflow

1. POST `/api/users` → récupère l’id

2. POST `/api/media/upload` (photo et/ou CNI, avec l’id du user)

3. GET `/api/users/{id}` → pour afficher le profil avec les médias