Ce fichier est la pour vous aider à utiliser l’API de RESA développée par Constantin Herrmann dans le cadre du travail de diplôme Technicien ES au CFPT-I (2020).

### User

Lien : /api/v2/user/

#### Lecture

Récupérer les informations d’un user avec ses permissions

* + Lien : /api/v2/user/get/
  + Paramètres :
    - id : l’id de l’utilisateur recherché
  + Lien avec paramètres : /api/v2/user/get/?id=[id de l'utilisateur]
  + Retour :
    - Un tableau avec les champs :
      * id : l’id de l’utilisateur
      * first\_name : le prénom de l’utilisateur
      * last\_name : le nom de famille de l’utilisateur
      * phone : le numéro de téléphone de l’utilisateur
      * email : l’email de l’utilisateur
      * username : le code à 4 chiffre d’identification de l’utilisateur
      * permissions : un tableau avec toutes ses permissions. Chaque permission contient :
        + etablishment\_name : le nom de l’établissement (si il y en à un, sinon “-”)
        + permission\_name : le nom de la permission (ex: manager)

Récupérer les informations de base d’un user

* + Lien : /api/v2/user/get/
  + Paramètres :
    - user : (il n’y à pas besoin de valeur)
    - id : l’id de l’utilisateur recherché
  + Lien avec paramètres : /api/v2/user/get/?user&id=[id de l'utilisateur]
  + Retour :
    - Un tableau avec les champs :
      * id : l’id de l’utilisateur
      * first\_name : le prénom de l’utilisateur
      * last\_name : le nom de famille de l’utilisateur
      * phone : le numéro de téléphone de l’utilisateur
      * email : l’email de l’utilisateur
      * username : le code à 4 chiffre d’identification de l’utilisateur

Récupérer tous les utilisateurs avec une certaine permission

* + Lien : /api/v2/user/get/
  + Paramètres :
    - byPermission : l’id de la permission recherchée
  + Lien avec paramètres : /api/v2/user/get/?byPermission=[id de la permission]
  + Retour :
    - Un tableau de users, ou chaque user possède les champs :
      * first\_name : le prénom de l’utilisateur
      * last\_name : le nom de famille de l’utilisateur
      * phone : le numéro de téléphone de l’utilisateur
      * email : l’email de l’utilisateur
      * username : le code à 4 chiffre d’identification de l’utilisateur

Récupérer toutes les permissions d’un utilisateur

* + Lien : /api/v2/user/get/
  + Paramètres :
    - permissions : (il n’y à pas besoin de valeur)
    - id : l’id de l’utilisateur recherché
  + Lien avec paramètres : /api/v2/user/get/?permissions&id=[id de l'utilisateur]
  + Retour :
    - Un tableau de permissions, ou chaque permissions possède les champs :
      * etablishment\_name : le nom de l’établissement (si il y en à un, sinon “-”)
      * permission\_name : le nom de la permission (ex: manager)

Récupérer tous les utilisateurs

* + Lien : /api/v2/user/get/
  + Paramètres :
    - aucun
  + Lien avec paramètres : /api/v2/user/get/
  + Retour :
    - Un tableau de users, ou chaque user possède les champs :
      * id : l’id de l’utilisateur
      * first\_name : le prénom de l’utilisateur
      * last\_name : le nom de famille de l’utilisateur
      * phone : le numéro de téléphone de l’utilisateur
      * email : l’email de l’utilisateur
      * username : le code à 4 chiffre d’identification de l’utilisateur

Récupérer tous les employés d’un etablissement

* + Lien : /api/v2/user/employes/
  + Paramètres :
    - workingFor : l’id de l’etablissement
  + Lien avec paramètres : /api/v2/user/employes/?workingFor=[id de l'etablissement]
  + Retour :
    - Un tableau de users, ou chaque user possède les champs :
      * id : l’id de l’utilisateur
      * user\_firstname : le prénom de l’utilisateur
      * user\_lastname : le nom de famille de l’utilisateur
      * permission\_name : le nom de sa permission (ex: Manager)
      * permission\_level : le niveau de la permission (ex : 2)

#### Divers

Login d’un employé dans son restaurant

* + Lien : /api/v2/user/get/
  + Paramètres :
    - login\_e : (il n’y à pas besoin de valeur)
    - username : l’identifiant à 4 chiffres de l’utilisateur
    - password : le mot de passe hashé en sha256
  + Lien avec paramètres : /api/v2/user/get/?login&username=[identifiant 4 chiffres]&password=[mot de passe hashé (sha256)]
  + Retour :
    - Un tableau avec les champs :
    - id : l’id de l’utilisateur
    - first\_name : le prénom de l’utilisateur
    - last\_name : le nom de famille de l’utilisateur
    - phone : le numéro de téléphone de l’utilisateur
    - email : l’email de l’utilisateur

Générer un identifiant numérique aléatoire

* + Lien : /api/v2/user/username/generate/
  + Paramètres :
    - Aucun
  + Lien avec paramètres : /api/v2/user/username/generate/
  + Retour :
    - un numéro aléatoire

### Floor

Lien : /api/v2/etablishment/floor/

#### Lecture

Récupérer tous les étages, zones et horaires d’un etablissement

* + Lien : /api/v2/etablishment/floor/get/
  + Paramètres :
    - id : l’id de l’établissement
  + Lien avec paramètres : /api/v2/etablishment/floor/get/?id=[id de l'établissement]
  + Retour :
    - Un tableau avec comme clés les id des étages:
      * [id de l'étage] : l’id de l’étage comme index du tableau afin de facilement le retrouver
        + name : le nom de l’étage
        + zones : tableau des zones

[nom de la zone] : le nom de la zone

[xx:xx:xx] : l’heure de début (begin)

[xx:xx:xx] : l’heure de fin (end)

#### Création

Créeation d’un étage

* + Lien : /api/v2/etablishment/floor/create/
  + Paramètres :
    - create : (il n’y à pas besoin de valeur)
    - name : nom de l’étage
    - etablishment : l’id de l’établissement
  + Lien avec paramètres : /api/v2/etablishment/floor/create/?create&name=[nom de l'étage]&etablishment=[id de l'établissement]
  + Retour :
    - nothing

Création d’un étage à partir d’un formulaire

* + Lien : /api/v2/etablishment/floor/create/form/
  + Paramètres :
    - $\_POST
      * name : nom de l’étage
      * etablishment : l’id de l’établissement
  + Lien avec paramètres : /api/v2/etablishment/floor/create/form/
  + Retour :
    - nothing

### Schedule

Lien : /api/v2/etablishment/floor/zone/schedule/

#### Lecture

Récupère l’id de la dernière zone ajoutée

* + Lien : /api/v2/etablishment/floor/zone/get/
  + Paramètres :
    - last : (il n’y à pas besoin de valeur)
  + Lien avec paramètres : /api/v2/etablishment/floor/zone/get/?last
  + Retour :
    - schedules :
      * schedule\_id : l’id de l’horaire
      * begin : l’heure de début
      * end : l’heure de fin

### Images

Lien : /api/v2/images/

Récupérer les informations completes d’une image

* + Lien : /api/v2/images/get/
  + Paramètres :
    - data : (il n’y à pas besoin de valeur)
    - id : l’id de l’image
  + Lien avec paramètres : /api/v2/images/get/?data&id=[l'id de l'image]
  + Retour :
    - [le path complet]

Etre rediriger sur l’image

* + Lien : /api/v2/images/get/
  + Paramètres :
    - id : l’id de l’image
  + Lien avec paramètres : /api/v2/images/get/?id=[l'id de l'image]
  + Retour :
    - Redirection sur la page de l’image

Récupérer toutes les images d’un etablissement

* + Lien : /api/v2/images/get/
  + Paramètres :
    - etablishment : (il n’y à pas besoin de valeur)
    - id : l’id de l’établissement
  + Lien avec paramètres : /api/v2/images/get/?etablishment&id=[l'id de l'établissement]
  + Retour :
    - Un tbaleau avec tous les chemins:
      * full\_path : le lien complet dans l’API pour rejoindre l’image

Récupérer toutes les images d’un repas

* + Lien : /api/v2/images/get/
  + Paramètres :
    - dish : (il n’y à pas besoin de valeur)
    - id : l’id du repas
  + Lien avec paramètres : /api/v2/images/get/?dish&id=[l'id du repas]
  + Retour :
    - Un tbaleau avec tous les chemins:
      * full\_path : le lien complet dans l’API pour rejoindre l’image

Récupérer la photo de profil d’un utilisateur

* + Lien : /api/v2/images/get/
  + Paramètres :
    - user : (il n’y à pas besoin de valeur)
    - id : l’id de l’établissement
  + Lien avec paramètres : /api/v2/images/get/?user&id=[l'id de l'établissement]
  + Retour :
    - Un tbaleau avec tous les chemins:
      * full\_path : le lien complet dans l’API pour rejoindre l’image

Mise en ligne d’une photo

* + Afin de mettre en ligne une photo il faut :
    1. Inclure le fichier /api/v2/images/upload/index.php
    2. Dans ce fichier ce trouvent les 2 fonctions suivantes :
       1. SaveImageEtablishement($idEtbalishement, $idUploader, $file, $target\_dir, $toMove)
       2. ~~SaveImageDish($idDish, $idUploader, $file, $target\_dir, $toMove)~~ N’est plus valide
       3. ~~SaveImageUser($idDish, $idUploader, $file, $target\_dir, $toMove)~~ N’est plus valide
    - Les deux fonction prennent les paramètres suivants :
      * idEtablissement ou idDish ou idUser en fonction de quelle fonction on appelle
      * idUploader : l’id du user qui met en ligne la photo
      * file\_name : le nom de la photo à créer
      * $tmp : le fichier tempon de l’image