Compta

Dépendances

PHP, Composer

Ubuntu :

```
$ sudo apt install php php-fpm composer
```

Installation

Clonez la branche “noauth” du dépôt :

```
$ git clone -b noauth https://github.com/vv221/compta-back
$ cd compta-back
```

Ajoutez les droits d’écriture pour votre serveur web sur le répertoire logs :

```
$ chgrp www-data logs
$ chmod g+w logs
```

Renommez le fichier app/config/prod.php.example en app/config/prod.php et renseignez-y les identifiants permettant d’accéder à votre base de données. Dans ce même fichier renseignez le login et le mot de passe du compte d’administration.

Importez le fichier sql/structure.sql dans votre base pour créer la structure attendue par l’application.

Utilisez Composer pour installer les dépendances de l’application :

```
$ composer update --no-dev
```

Assignez un hôte à cette application, pointant sur web/index.php

Nginx/PHP7 :

```
server {
    listen 80;
    listen [::]:80;

    root /srv/http/silex/compta-back/web;

    index index.php;

    server_name compta;

    location / {
        autoindex off;
        try_files $uri $uri/ /index.php;
```

```
    }

    location ~ /index\.php(/|$) {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.0-fpm.sock;
        fastcgi_split_path_info ^(.+\.php)(/.*)$;
    }
}
```

Utilisation

Route /groups (GET)

Renvoie la liste des groupes existants, dans un format similaire à celui-ci :

```
{
    "records": [
        {
            "id": "1",
            "name": "Ete 2016"
        }
    ],
    "status": "OK"
}
```

Route /group/{id}/users (GET)

Renvoie la liste des utilisateurs appartenant au groupe avec l'id {id}, dans un format similaire à celui-ci :

```
{
    "records": [
        {
            "Id": "66",
            "username": "Eszter",
            "usergroup": "Ete 2016",
            "usercolor": "blue"
        },
        {
            "Id": "67",
            "username": "Erwann",
            "usergroup": "Ete 2016",
            "usercolor": "blue"
        },
        {
```

```
            "Id": "103",
            "username": "Romy",
            "usergroup": "Ete 2016",
            "usercolor": "white"
        }
    ],
    "status": "OK"
}
```


Route /group/{id}/depenses (GET)

Renvoie la liste des dépenses concernant au groupe avec l'id {id}, dans un format similaire à celui-ci :


```
{
    "records": [
        {
            "Id": "268",
            "Montant": "666.00",
            "Payeur": "67",
            "Concernes": "66,67,82,83,87,95,96,97,100,101,102,103",
            "Date": "0",
            "nbConcernes": 12,
            "usergroup": "Ete 2016",
            "Description": "Les courses du diable"
        },
        {
            "Id": "269",
            "Montant": "65.00",
            "Payeur": "83",
            "Concernes": "66",
            "Date": "0",
            "nbConcernes": 1,
            "usergroup": "Ete 2016",
            "Description": "Vaccin et croquettes"
        },
        {
            "Id": "273",
            "Montant": "23.00",
            "Payeur": "95",
            "Concernes": "82,83,103",
            "Date": "0",
            "nbConcernes": 3,
            "usergroup": "Ete 2016",
            "Description": "Tabac"
```

```
        }
    ],
    "status": "OK"
}
```

Route /login (POST)

Demande une authentification en tant que session d'administration. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "name": "admin_name",
    "password": "admin_password"
}
```

Un retour contenant une clé d'API valable pour 25 minutes vous sera renvoyée, dans un format similaire à celui-ci :

```
{
    "key": "c0hjmdQfUL7PvpDq7KgmDeA/QELP8kWEzZpUSVMIIWgJA+2XMjd8GfYOcEWbdZPeyxN85HoWVXvgQf2s
    "status": "OK"
}
```

Route /logout (GET)

Demande la révocation d'une clé d'API. Cette requête doit fournir la clé dans un header HTTP nommé 'apikey'.

Route /admin/group (POST)

Cette requête doit fournir une clé d'API valide dans un header HTTP nommé 'apikey'.

Ajoute un nouveau groupe à l'application. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "name": "Printemps 2015"
}
```

Modifie un groupe existant. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "id": "2"
    "name": "Automne 2014"
}
```

Route /admin/user (POST)

Cette requête doit fournir une clé d'API valide dans un header HTTP nommé 'apikey'.

Ajoute un nouvel utilisateur à l'application. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "username": "Benj",
    "usergroup": "Ete 2016",
    "usercolor": "red"
}
```

Modifie un utilisateur existant. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "Id": "83",
    "username": "Aurele",
    "usergroup": "Ete 2016",
    "usercolor": "red"
}
```

Route /admin/depense (POST)

Cette requête doit fournir une clé d'API valide dans un header HTTP nommé 'apikey'.

Ajoute une nouvelle dépense à l'application. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "Montant": "78.00",
    "Payeur": "97",
    "Concernes": "66,67,82,83,87,95,96,97,100,101,102,103",
    "Date": "0",
    "nbConcernes": 12,
    "usergroup": "Ete 2016",
    "Description": "Courses du samedi soir"
}
```

Modifie une dépense existante. Cette requête doit fournir un fichier json d'un format similaire à celui-ci :

```
{
    "Id": "271",
    "Montant": "100.00",
    "Payeur": "87",
```

```
    "Concernes": "67",
    "Date": "0",
    "nbConcernes": 1,
    "usergroup": "Ete 2016",
    "Description": "Dette des dernières vacances"
}
```

Route /admin/group/{id} (DELETE)

Cette requête doit fournir une clé d'API valide dans un header HTTP nommé 'apikey'.

Supprime le groupe avec l'id {id}. Les dépenses associées à ce groupe ainsi que les utilisateurs n'appartenant qu'à ce groupe seront automatiquement supprimés.

Route /admin/user/{id} (DELETE)

Cette requête doit fournir une clé d'API valide dans un header HTTP nommé 'apikey'.

Supprime l'utilisateur avec l'id {id}. Les dépenses dont cet utilisateur est le payeur ou le seul concerné seront automatiquement supprimées.

Route /admin/depense/{id} (DELETE)

Cette requête doit fournir une clé d'API valide dans un header HTTP nommé 'apikey'.

Supprime la dépense avec l'id {id}.