



## **DOCUMENTATION DE L'API**

# TABLE DES MATIERES

<b>PRESENTATION</b>	3
<b>FONCTIONNALITES</b>	3
<b>I- AUTHENTIFICATION</b>	3
<b>II- Utilisateur</b>	4
1- Liste des utilisateurs	4
2- Voir un utilisateur	4
3- Créer un utilisateur	5
4- Mettre à jour utilisateur	5
5- Mettre à jour le mot de passe d'utilisateur	6
6- Supprimer un utilisateur	6
<b>III- Visiteurs</b>	7
1- Liste des visiteurs	7
2- Voir un visiteur	8
3- Créer un visiteur	9
4- Créer un Académicien	9
5- Mettre à jour visiteur	10
6- Générer un nouveau code QR visiteur	10
7- Supprimer un visiteur	11
<b>IV- Historique</b>	11
1- Liste des Historiques	11
2- Liste historique d'un visiteur	12
3- Nouvelle entrée	13
4- Nouvelle sortie	14
5- Liste visiteurs présent (Encore dans la villa)	15
6- Liste des 10 dernières entrée	16
7- Liste des 10 dernières sorties	17
8- Liste des visiteurs présents dans la semaine	18
<b>V- Type</b>	19
1- Liste type visiteur	19
2- Liste type info	19

# PRESENTATION

L'api e-Register permettra de lier l'application web et l'application mobile afin de pouvoir utiliser les services mis à leur disposition.

## FONCTIONNALITES

### I- AUTHENTIFICATION

Ce service permet d'authentifier un utilisateur

**VERBE HTTP** => GET

**URL** => `http://adresse_du_serveur/api/v1/auth/{login}/{password}`

**SUCCES** => Retourne l'utilisateur (Objet JSON).

**ERREUR** => Retourne :

**404** => Si l'utilisateur n'est pas trouvé (Inexistant ou login ou identifiant incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    "identifiant": "ADMIN151296",
3    "nom": "M",
4    "prenoms": "Dylan",
5    "tel": "79512947",
6    "role": "Admin",
7    "login": "dydy",
8    "password": "$2y$10$Itxco.8SOLz.VCEBgN04aeKYQ1IPuTicTFuOqMrhfN9kABMsWbh5W",
9    "photo": "http://localhost/storage/users/ADMIN151296_users.png",
10   "updated_at": "2020-04-03T05:36:39.000000Z",
11   "created_at": "2020-04-03T05:36:39.000000Z",
12   "id": 1
13 }
```

## II- Utilisateur

### 1- Liste des utilisateurs

Ce service permet de recevoir la liste des utilisateurs (utilisateurs des application web et mobile)

**VERBE HTTP** => GET

**URL** => `http://adresse_du_serveur/api/v1/users`

**SUCCES** => Retourne la liste des utilisateurs (Tableau d'objet JSON).

**ERREUR** => Retourne un tableau vide si aucun utilisateur ou :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

**EXEMPLE RETOURNE EN CAS DE SUCCES (Un user dans cet exemple mais peut être plus)**

```
1  [
2    {
3      "id": 2,
4      "identifiant": "ADMIN151296",
5      "nom": "M",
6      "prenoms": "Dylan",
7      "role": "Admin",
8      "tel": "79512947",
9      "photo": "http://localhost/storage/users/ADMIN151296_users.png",
10     "login": "dydy",
11     "password": "$2y$10$v2JPJNei.RNPcSD2CfR18uE17dZe/1La3sOI0QDUAXJSSeQWnpr/.",
12     "created_at": "2020-04-03T05:41:57.000000Z",
13     "updated_at": "2020-04-03T05:42:28.000000Z"
14   }
15 ]
```

### 2- Voir un utilisateur

Ce service permet de récupérer un utilisateur

**VERBE HTTP** => GET

**URL** => `http://adresse_du_serveur/api/v1/users/{identifiant}`

**SUCCES** => Retourne l'utilisateur (Objet JSON).

**ERREUR** => Retourne :

**404** => Si l'utilisateur n'est pas trouvé (Inexistant ou identifiant incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

**EXEMPLE RETOURNE EN CAS DE SUCCES**

```
1  {
2    "identifiant": "ADMIN151296",
3    "nom": "M",
4    "prenoms": "Dylan",
5    "tel": "79512947",
6    "role": "Admin",
7    "login": "dydy",
8    "password": "$2y$10$Itxco.8SOLz.VCEBgn04aeKYQ1IPUtIcTFuQqMhFh9KABMsWb5W",
9    "photo": "http://localhost/storage/users/ADMIN151296_users.png",
10   "updated_at": "2020-04-03T05:36:39.000000Z",
11   "created_at": "2020-04-03T05:36:39.000000Z",
12   "id": 1
13 }
```

### 3- Créer un utilisateur

Ce service permet de créer un utilisateur

**VERBE HTTP** => POST

**URL** => http://adresse\_du\_serveur/api/v1/users/

**SUCCES** => Retourne l'utilisateur créé (Objet JSON).

**ERREUR** => Retourne :

**444** => Si l'utilisateur existe déjà (Identifiant ou login déjà utilisé).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    "identifiant": "ADMIN151296",
3    "nom": "M",
4    "prenoms": "Dylan",
5    "tel": "79512947",
6    "role": "Admin",
7    "login": "dydy",
8    "password": "$2y$10$Itxco.8SOLz.VCEBgn04aeKYQ1IPuTicTFu0qMrhfN9kABMsbh5W",
9    "photo": "http://localhost/storage/users/ADMIN151296_users.png",
10   "updated_at": "2020-04-03T05:36:39.000000Z",
11   "created_at": "2020-04-03T05:36:39.000000Z",
12   "id": 1
13 }
```

### 4- Mettre à jour utilisateur

Ce service permet de mettre à jour un utilisateur

**VERBE HTTP** => POST

**URL** => http://adresse\_du\_serveur/api/v1/users/update/{**identifiant**}

**SUCCES** => Retourne l'utilisateur mis à jour (Objet JSON).

**ERREUR** => Retourne :

**404** => Si l'utilisateur n'est pas trouvé (Inexistant ou identifiant incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    "identifiant": "ADMIN151296",
3    "nom": "M",
4    "prenoms": "Dylan",
5    "tel": "79512947",
6    "role": "Admin",
7    "login": "dydy",
8    "password": "$2y$10$Itxco.8SOLz.VCEBgn04aeKYQ1IPuTicTFu0qMrhfN9kABMsbh5W",
9    "photo": "http://localhost/storage/users/ADMIN151296_users.png",
10   "updated_at": "2020-04-03T05:36:39.000000Z",
11   "created_at": "2020-04-03T05:36:39.000000Z",
12   "id": 1
13 }
```

## 5- Mettre à jour le mot de passe d'utilisateur

Ce service permet de mettre à jour le mot de passe d'un utilisateur

**NB :** Nom du champ du mot de passe doit être **new\_password**

**VERBE HTTP** => POST

**URL** => http://adresse\_du\_serveur/api/v1/edit-password/{identifiant}

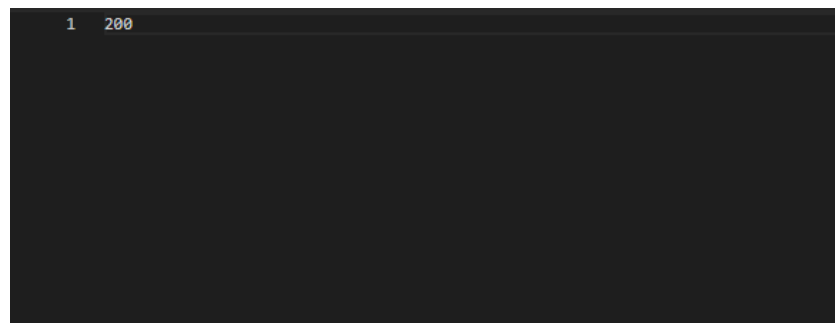
**SUCCES** => Retourne 200.

**ERREUR** => Retourne :

**404** => Si l'utilisateur n'est pas trouvé (Inexistant ou identifiant incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES



## 6- Supprimer un utilisateur

Ce service permet de supprimer un utilisateur.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/users/delete/{identifiant}

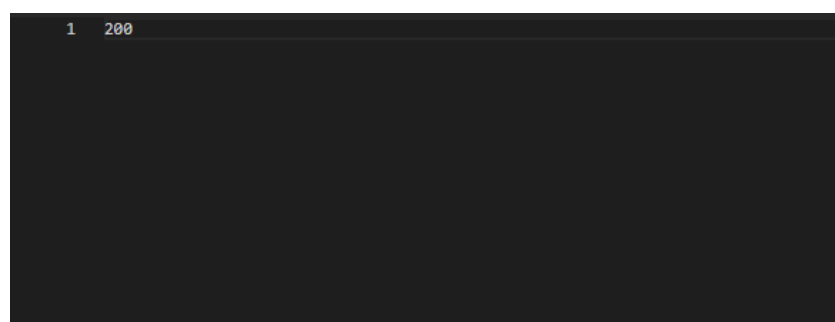
**SUCCES** => Retourne 200.

**ERREUR** => Retourne :

**404** => Si l'utilisateur n'est pas trouvé (Inexistant ou identifiant incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES



### III- Visiteurs

#### 1- Liste des visiteurs

Ce service permet de recevoir la liste des visiteurs.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/visiteurs

**SUCCES** => Retourne la liste des visiteurs (Tableau d'objet JSON).

**ERREUR** => Retourne un tableau vide si aucun visiteur ou :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

**EXEMPLE RETOURNE EN CAS DE SUCCES (Un user dans cet exemple mais peut être plus)**

```
1 {
2   {
3     "id": 1,
4     "nom": "Koukou",
5     "prenom": "Adilfired",
6     "nom_piece": "C10103072000",
7     "type_piece": "COT",
8     "type_visiteur_id": 2,
9     "created_at": "2020-04-03T04:20:36.000000Z",
10    "updated_at": "2020-04-03T04:20:36.000000Z",
11    "type_visiteur": {
12      "id": 3,
13      "libelle": "Visiteur",
14      "created_at": null,
15      "updated_at": null
16    },
17    "info_complementaire": [
18      {
19        "id": 1,
20        "valeur": "http://localhost/storage/piece/C10103072000_piece.jpg",
21        "date_debut": null,
22        "date_fin": null,
23        "valide": "1",
24        "visiteur_id": 1,
25        "type_info_id": 1,
26        "created_at": "2020-04-03T04:20:36.000000Z",
27        "updated_at": "2020-04-03T04:20:36.000000Z",
28        "type_info": {
29          "id": 1,
30          "libelle": "Piece",
31          "created_at": null,
32          "updated_at": null
33        }
34      },
35      {
36        "id": 2,
37        "valeur": "http://localhost/storage/users/user.jpg",
38        "date_debut": null,
39        "date_fin": null,
40        "valide": "1",
41        "visiteur_id": 1,
42        "type_info_id": 2,
43        "created_at": "2020-04-03T04:20:36.000000Z",
44        "updated_at": "2020-04-03T04:20:36.000000Z",
45        "type_info": {
46          "id": 2,
47          "libelle": "Photo",
48          "created_at": null,
49          "updated_at": null
50        }
51      }
52    ],
53    {
54      "id": 2,
55      "valeur": "http://localhost/storage/qr_code/C10103072000_qrCode.png",
56      "date_debut": "2020-04-03T04:20:36.000000Z",
57      "date_fin": "2020-04-04T04:36:36.000000Z",
58      "valide": "1",
59      "visiteur_id": 1,
60      "type_info_id": 3,
61      "created_at": "2020-04-03T04:20:36.000000Z",
62      "updated_at": "2020-04-03T04:20:36.000000Z",
63      "type_info": {
64        "id": 3,
65        "libelle": "Qrcode",
66        "created_at": null,
67        "updated_at": null
68      }
69    }
70  }
71 }
```

## 2- Voir un visiteur

Ce service permet de récupérer un visiteur

**VERBE HTTP** => GET

**URL** => `http://adresse_du_serveur/api/v1/visiteurs/{numero_piece}`

**SUCCES** => Retourne le visiteur (Objet JSON).

**ERREUR** => Retourne :

**404** => Si le visiteur n'est pas trouvé (Inexistant ou numéro de pièce incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2
3      "id": 1,
4      "nom": "Kiwakuu",
5      "premiere": "Nifired",
6      "num_piece": "C10103072000",
7      "type_piece": "CNI",
8      "type_visiteur_id": 1,
9      "created_at": "2020-04-03T04:20:36.000000Z",
10     "updated_at": "2020-04-03T04:20:36.000000Z",
11     "type_visiteur": {
12         "id": 1,
13         "libelle": "Visiteur",
14         "created_at": null,
15         "updated_at": null
16     },
17     "info_complementaire": [
18         {
19             "id": 1,
20             "valeur": "http://localhost/storage/piece/C10103072000_piece.jpg",
21             "date_debut": null,
22             "date_fin": null,
23             "valide": "1",
24             "visiteur_id": 1,
25             "type_info_id": 1,
26             "created_at": "2020-04-03T04:20:36.000000Z",
27             "updated_at": "2020-04-03T04:20:36.000000Z",
28             "type_info": {
29                 "id": 1,
30                 "libelle": "Piece",
31                 "created_at": null,
32                 "updated_at": null
33             }
34         },
35         {
36             "id": 2,
37             "valeur": "http://localhost/storage/users/user.jpg",
38             "date_debut": null,
39             "date_fin": null,
40             "valide": "1",
41             "visiteur_id": 1,
42             "type_info_id": 2,
43             "created_at": "2020-04-03T04:20:36.000000Z",
44             "updated_at": "2020-04-03T04:20:36.000000Z",
45             "type_info": {
46                 "id": 2,
47                 "libelle": "Photo",
48                 "created_at": null,
49                 "updated_at": null
50             }
51         }
52     ],
53     {
54         "id": 3,
55         "valeur": "http://localhost/storage/qrcode/C10103072000_qrCode.png",
56         "date_debut": "2020-04-03T04:20:36.000000Z",
57         "date_fin": "2020-04-04T04:20:36.000000Z",
58         "valide": "1",
59         "visiteur_id": 1,
60         "type_info_id": 3,
61         "created_at": "2020-04-03T04:20:36.000000Z",
62         "updated_at": "2020-04-03T04:20:36.000000Z",
63         "type_info": {
64             "id": 3,
65             "libelle": "Qrcode",
66             "created_at": null,
67             "updated_at": null
68         }
69     }
70 }
71 }
```



### 3- Créer un visiteur

Ce service permet de créer un visiteur

**VERBE HTTP** => POST

**URL** => http://adresse\_du\_serveur/api/v1/visiteurs/store

**SUCCES** => Retourne le visiteur crée (Objet JSON).

**ERREUR** => Retourne :

**444** => Si le visiteur existe déjà (numéro de pièce déjà utilisé).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1 {
2   "nom": "Kouakou",
3   "prenoms": "Wilfired",
4   "num_piece": "CI0193872898",
5   "type_piece": "CNI",
6   "type_visiteur_id": 3,
7   "updated_at": "2020-04-03T05:03:57.000000Z",
8   "created_at": "2020-04-03T05:03:57.000000Z",
9   "id": 1
10 }
```

### 4- Créer un Académicien

Ce service permet de créer un académicien.

**VERBE HTTP** => POST

**URL** => http://adresse\_du\_serveur/api/v1/academicien/store

**SUCCES** => Retourne l'académicien crée (Objet JSON).

**ERREUR** => Retourne :

**444** => Si l'académicien existe déjà (numéro de pièce déjà utilisé).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1 {
2   "nom": "Mouahet",
3   "prenoms": "Dylan",
4   "num_piece": "S01512",
5   "type_piece": "Carte OCI",
6   "type_visiteur_id": 1,
7   "updated_at": "2020-04-03T05:04:03.000000Z",
8   "created_at": "2020-04-03T05:04:03.000000Z",
9   "id": 2
10 }
```

## 5- Mettre à jour visiteur

Ce service permet de mettre à jour un visiteur

**VERBE HTTP** => POST

**URL** => `http://adresse_du_serveur/api/v1/visiteurs/update/{numero_piece}`

**SUCCES** => Retourne le visiteur mit à jour (Objet JSON).

**ERREUR** => Retourne :

**404** => Si visiteur n'est pas trouvé (Inexistant ou numéro de pièce incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    "nom": "Kouakou",
3    "prenoms": "Wilfred",
4    "num_piece": "CI0193872898",
5    "type_piece": "CNI",
6    "type_visiteur_id": 3,
7    "updated_at": "2020-04-03T05:03:57.000000Z",
8    "created_at": "2020-04-03T05:03:57.000000Z",
9    "id": 1
10 }
```

## 6- Générer un nouveau code QR visiteur

Ce service permet de générer un nouveau code QR pour le visiteur

**VERBE HTTP** => POST

**URL** => `http://adresse_du_serveur/api/v1/visiteurs/new-qrcode/{numero_piece}`

**SUCCES** => Retourne le visiteur (Objet JSON).

**ERREUR** => Retourne :

**404** => Si visiteur n'est pas trouvé (Inexistant ou numéro de pièce incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    "nom": "Kouakou",
3    "prenoms": "Wilfred",
4    "num_piece": "CI0193872898",
5    "type_piece": "CNI",
6    "type_visiteur_id": 3,
7    "updated_at": "2020-04-03T05:03:57.000000Z",
8    "created_at": "2020-04-03T05:03:57.000000Z",
9    "id": 1
10 }
```

## 7- Supprimer un visiteur

Ce service permet de supprimer un visiteur.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/visiteurs/delete/{num\_piece}

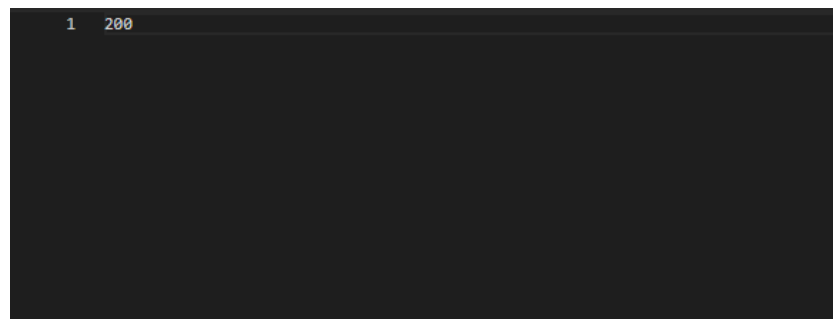
**SUCCES** => Retourne 200.

**ERREUR** => Retourne :

**404** => Si le visiteur n'est pas trouvé (Inexistant ou numéro de pièce incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES



## IV- Historique

### 1- Liste des Historiques

Ce service permet de recevoir la liste des historiques.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/in\_out

**SUCCES** => Retourne la liste des historiques (Tableau d'objet JSON).

**ERREUR** => Retourne un tableau vide si aucun historique ou :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES (Un historique dans cet exemple mais peut être plus)

```

1  {
2      {
3          "id": 1,
4          "motif": "Visite personnel",
5          "date_arrive": "2020-04-03 05:48:08",
6          "date_depart": null,
7          "visiteur_id": 1,
8          "user_id": 2,
9          "created_at": "2020-04-03T05:48:08.000000Z",
10         "updated_at": "2020-04-03T05:50:44.000000Z",
11         "visiteur": {
12             "id": 1,
13             "nom": "Kouakou",
14             "prenoms": "Wilfred",
15             "num_piece": "CI0193872898",
16             "type_piece": "CNI",
17             "type_visiteur_id": 3,
18             "created_at": "2020-04-03T05:03:57.000000Z",
19             "updated_at": "2020-04-03T05:03:57.000000Z",
20             "type_visiteur": {
21                 "id": 3,
22                 "libelle": "Visiteur",
23                 "created_at": null,
24                 "updated_at": null
25             }
26         },
27         "user": {
28             "id": 2,
29             "identifiant": "ADMIN151296",
30             "nom": "M",
31             "prenoms": "Dylan",
32             "role": "Admin",
33             "tel": "79512947",
34             "photo": "http://localhost/storage/users/ADMIN151296_users.png",
35             "login": "dydy",
36             "password": "$2y$10$v2JPJNei.RNPcSD2CfR18uE17dZe/1La3s0IOQDUAXJSSeQWnpr/.",
37             "created_at": "2020-04-03T05:41:57.000000Z",
38             "updated_at": "2020-04-03T05:42:28.000000Z"
39         }
40     }
41 }

```

## 2- Liste historique d'un visiteur

Ce service permet de recevoir la liste des historiques d'un visiteur.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/in\_out/{num\_piece}

**SUCCES** => Retourne l'historique du visiteur (Objet JSON).

**ERREUR** => Retourne :

**404** => Si le visiteur n'est pas trouvé (Inexistant ou numéro de pièce incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```

1  {
2    "id": 1,
3    "nom": "Kouakou",
4    "prenoms": "Wilfired",
5    "num_piece": "CI0193872898",
6    "type_piece": "CNI",
7    "type_visiteur_id": 3,
8    "created_at": "2020-04-03T05:03:57.000000Z",
9    "updated_at": "2020-04-03T05:03:57.000000Z",
10   "type_visiteur": {
11     "id": 3,
12     "libelle": "Visiteur",
13     "created_at": null,
14     "updated_at": null
15   },
16   "in_out": [
17     {
18       "id": 1,
19       "motif": "Visite personnel",
20       "date_arrive": "2020-04-03 05:48:08",
21       "date_depart": "2020-04-03 10:48:08",
22       "visiteur_id": 1,
23       "user_id": 2,
24       "created_at": "2020-04-03T05:48:08.000000Z",
25       "updated_at": "2020-04-03T05:50:44.000000Z",
26       "visiteur": {
27         "id": 1,
28         "nom": "Kouakou",
29         "prenoms": "Wilfired",
30         "num_piece": "CI0193872898",
31         "type_piece": "CNI",
32         "type_visiteur_id": 3,
33         "created_at": "2020-04-03T05:03:57.000000Z",
34         "updated_at": "2020-04-03T05:03:57.000000Z",
35         "type_visiteur": {
36           "id": 3,
37           "libelle": "Visiteur",
38           "created_at": null,
39           "updated_at": null
40         }
41       },
42       "user": {
43         "id": 2,
44         "identifiant": "ADMIN151296",
45         "nom": "M",
46         "prenoms": "Dylan",
47         "role": "Admin",
48         "tel": "79512947",
49         "photo": "http://localhost/storage/users/ADMIN151296_users.png",
50         "login": "dydy",
51         "password": "$2y$10$v23PJNei.RNPcSD2CFR1BuE17dZe/1La3s0IOQDUAXJSSeQWnpr/.",
52         "created_at": "2020-04-03T05:41:57.000000Z",
53         "updated_at": "2020-04-03T05:42:28.000000Z"
54       }
55     }
56   ]
57 }

```

### 3- Nouvelle entrée

Ce service permet d'enregistrer l'entrée d'un visiteur.

**NB :** Si c'est une visite a un responsable ODA, prévoir le champ **user\_id** pour envoyer l'id du responsable en question afin de pouvoir lui envoyer une notification.

**VERBE HTTP => POST**

**URL =>**

http://adresse\_du\_serveur/api/v1/in\_out/in/{num\_piece\_visiteur}/{identifiant\_user}

**SUCCES =>** Retourne l'enregistrement (Objet JSON).

**ERREUR =>** Retourne :

**404 =>** Si le visiteur ou l'utilisateur n'est pas trouvé (Inexistant ou numéro de pièce ou identifiant incorrect).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1 {
2   "motif": "Visite personnel",
3   "date_arrive": "2020-04-03T20:13:05.139652Z",
4   "visiteur_id": 1,
5   "user_id": 2,
6   "updated_at": "2020-04-03T20:13:05.000000Z",
7   "created_at": "2020-04-03T20:13:05.000000Z",
8   "id": 2
9 }
```

## 4- Nouvelle sortie

Ce service permet d'enregistrer la sortie d'un visiteur.

**VERBE HTTP** => POST

**URL** => http://adresse\_du\_serveur/api/v1/in\_out/out/{num\_piece\_visiteur}

**SUCCES** => Retourne l'enregistrement (Objet JSON).

**ERREUR** => Retourne :

**404** => Si le visiteur n'est pas trouvé (Inexistant ou numéro de pièce ou identifiant incorrect).

**460** => Si le visiteur n'est pas encore entré (Possède une entrée mais pas de sortie de cette entrée).

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1 {
2   "id": 1,
3   "motif": "Visite personnel",
4   "date_arrive": "2020-04-03 05:48:08",
5   "date_depart": "2020-04-03T05:50:44.658484Z",
6   "visiteur_id": 1,
7   "user_id": 2,
8   "created_at": "2020-04-03T05:48:08.000000Z",
9   "updated_at": "2020-04-03T05:50:44.000000Z",
10  "visiteur": {
11    "id": 1,
12    "nom": "Kouakou",
13    "prenoms": "Wilfred",
14    "num_piece": "CI0193872898",
15    "type_piece": "CNI",
16    "type_visiteur_id": 3,
17    "created_at": "2020-04-03T05:03:57.000000Z",
18    "updated_at": "2020-04-03T05:03:57.000000Z",
19    "type_visiteur": {
20      "id": 3,
21      "libelle": "Visiteur",
22      "created_at": null,
23      "updated_at": null
24    }
25  },
26  "user": {
27    "id": 2,
28    "identifiant": "ADMIN151296",
29    "nom": "M",
30    "prenoms": "Dylan",
31    "role": "Admin",
32    "tel": "79512947",
33    "photo": "http://localhost/storage/users/ADMIN151296_users.png",
34    "login": "dydy",
35    "password": "$2y$10$v2JPJNeI.RNPcSD2CfR1BuE17dZe/1La3sOIQQDUAXJSSeQWnpr/. ",
36    "created_at": "2020-04-03T05:41:57.000000Z",
37    "updated_at": "2020-04-03T05:42:28.000000Z"
38  }
39 }
```

## 5- Liste visiteurs présent (Encore dans la villa)

Ce service permet d'obtenir la liste des visiteurs présent.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/in\_out/get/in

**SUCCES** => Retourne la liste des visiteurs présent (Objet JSON).

**ERREUR** => Retourne :

**400** => Mauvaise requête.

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    "nombre_present": 1,
3    "visiteurs": [
4      {
5        "id": 1,
6        "motif": "Visite personnel",
7        "date_arrive": "2020-04-03 05:48:08",
8        "date_depart": null,
9        "visiteur_id": 1,
10       "user_id": 2,
11       "created_at": "2020-04-03T05:48:08.000000Z",
12       "updated_at": "2020-04-03T05:50:44.000000Z",
13       "visiteur": {
14         "id": 1,
15         "nom": "Kouakou",
16         "prenoms": "Wilfred",
17         "num_piece": "CI0193872898",
18         "type_piece": "CNI",
19         "type_visiteur_id": 3,
20         "created_at": "2020-04-03T05:03:57.000000Z",
21         "updated_at": "2020-04-03T05:03:57.000000Z",
22         "type_visiteur": {
23           "id": 3,
24           "libelle": "Visiteur",
25           "created_at": null,
26           "updated_at": null
27         }
28       },
29       "user": {
30         "id": 2,
31         "identifiant": "ADMIN151296",
32         "nom": "M",
33         "prenoms": "Dylan",
34         "role": "Admin",
35         "tel": "79512947",
36         "photo": "http://localhost/storage/users/ADMIN151296_users.png",
37         "login": "dydy",
38         "password": "$2y$10$v2JPJNei.RNPcSD2CfR1BuE17dZe/1La3s0IOQUAXJSSeQWnpr/.",
39         "created_at": "2020-04-03T05:41:57.000000Z",
40         "updated_at": "2020-04-03T05:42:28.000000Z"
41       }
42     ]
43   }
44 }
```

## 6- Liste des 10 dernières entrée

Ce service permet d'obtenir la liste des 10 derniers visiteurs entré.

**VERBE HTTP** => GET

**URL** => `http://adresse_du_serveur/api/v1/in_out/get/last_arrived`

**SUCCES** => Retourne la liste des 10 derniers visiteurs entré (Tableau d'objet JSON).

**ERREUR** => Retourne :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2
3    "id": 1,
4    "motif": "Visite personnel",
5    "date_arrive": "2020-04-03 05:48:08",
6    "date_depart": null,
7    "visiteur_id": 1,
8    "user_id": 2,
9    "created_at": "2020-04-03T05:48:08.000000Z",
10   "updated_at": "2020-04-03T05:50:44.000000Z",
11   "visiteur": {
12     "id": 1,
13     "nom": "Kouakou",
14     "prenoms": "Wilfried",
15     "num_piece": "CI0193872898",
16     "type_piece": "CNI",
17     "type_visiteur_id": 3,
18     "created_at": "2020-04-03T05:03:57.000000Z",
19     "updated_at": "2020-04-03T05:03:57.000000Z",
20     "type_visiteur": {
21       "id": 3,
22       "libelle": "Visiteur",
23       "created_at": null,
24       "updated_at": null
25     }
26   },
27   "user": {
28     "id": 2,
29     "identifiant": "ADMIN151296",
30     "nom": "M",
31     "prenoms": "Dylan",
32     "role": "Admin",
33     "tel": "79512947",
34     "photo": "http://localhost/storage/users/ADMIN151296_users.png",
35     "login": "dydy",
36     "password": "$2y$10$v2JPJNei.RNPcSD2CfR1BuE17dZe/1La3s0IOQQUAXJ5SeQhnp/.",
37     "created_at": "2020-04-03T05:41:57.000000Z",
38     "updated_at": "2020-04-03T05:42:28.000000Z"
39   }
40 }
41
```



## 7- Liste des 10 dernières sorties

Ce service permet d'obtenir la liste des 10 derniers visiteurs sortie.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/in\_out/get/last\_depart

**SUCCES** => Retourne la liste des 10 derniers visiteurs sortie (Tableau d'objet JSON).

**ERREUR** => Retourne :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2
3      "id": 1,
4      "motif": "Visite personnel",
5      "date_arrive": "2020-04-03 05:48:08",
6      "date_depart": null,
7      "visiteur_id": 1,
8      "user_id": 2,
9      "created_at": "2020-04-03T05:48:08.000000Z",
10     "updated_at": "2020-04-03T05:50:44.000000Z",
11     "visiteur": {
12         "id": 1,
13         "nom": "Kouakou",
14         "prenoms": "Wilfried",
15         "num_piece": "CI0193872898",
16         "type_piece": "CNI",
17         "type_visiteur_id": 3,
18         "created_at": "2020-04-03T05:03:57.000000Z",
19         "updated_at": "2020-04-03T05:03:57.000000Z",
20         "type_visiteur": {
21             "id": 3,
22             "libelle": "Visiteur",
23             "created_at": null,
24             "updated_at": null
25         }
26     },
27     "user": {
28         "id": 2,
29         "identifiant": "ADMIN151296",
30         "nom": "M",
31         "prenoms": "Dylan",
32         "role": "Admin",
33         "tel": "79512947",
34         "photo": "http://localhost/storage/users/ADMIN151296_users.png",
35         "login": "dydy",
36         "password": "$2y$10$V2JPJNei.RNPcSD2CFR1BuE17dZe/1La3s0IQDUAXJSSeQWnpr/.",
37         "created_at": "2020-04-03T05:41:57.000000Z",
38         "updated_at": "2020-04-03T05:42:28.000000Z"
39     }
40 }
41 }
```

## 8- Liste des visiteurs présents dans la semaine

Ce service permet d'obtenir la liste des visiteurs présents dans la semaine.

**VERBE HTTP** => GET

**URL** => `http://adresse_du_serveur/api/v1/in_out/get/in_out_of_week`

**SUCCES** => Retourne la liste des visiteurs présents dans la semaine (Objet JSON).

**ERREUR** => Retourne :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1 {
2   "Total": 1,
3   "inOut": {
4     "1": [
5       {
6         "id": 1,
7         "motif": "Visite personnel",
8         "date_arrive": "2020-04-03 05:48:08",
9         "date_depart": "2020-04-03 10:48:08",
10        "visiteur_id": 1,
11        "user_id": 2,
12        "created_at": "2020-04-03T05:48:08.000000Z",
13        "updated_at": "2020-04-03T05:50:44.000000Z",
14        "visiteur": {
15          "id": 1,
16          "nom": "Kouakou",
17          "prenoms": "Wilfried",
18          "num_piece": "CI0193872898",
19          "type_piece": "CNI",
20          "type_visiteur_id": 3,
21          "created_at": "2020-04-03T05:03:57.000000Z",
22          "updated_at": "2020-04-03T05:03:57.000000Z",
23          "type_visiteur": {
24            "id": 3,
25            "libelle": "Visiteur",
26            "created_at": null,
27            "updated_at": null
28          }
29        },
30        "user": {
31          "id": 2,
32          "identifiant": "ADMIN151296",
33          "nom": "M",
34          "prenoms": "Dylan",
35          "role": "Admin",
36          "tel": "79512947",
37          "photo": "http://localhost/storage/users/ADMIN151296_users.png",
38          "login": "dydy",
39          "password": "$2y$10$v2JPJNei.RNPcSD2CfR18uE17dZe/1La3sOI0QUAXJ5SeQWnpr/.",
40          "created_at": "2020-04-03T05:41:57.000000Z",
41          "updated_at": "2020-04-03T05:42:28.000000Z"
42        }
43      }
44    ]
45  }
46 }
```

## V- Type

### 1- Liste type visiteur

Ce service permet d'obtenir la liste des types de visiteur.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/type\_visiteur

**SUCCES** => Retourne la liste des types de visiteur (Tableau d'objet JSON).

**ERREUR** => Retourne :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

#### EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    {
3      "id": 1,
4      "libelle": "Academicien",
5      "created_at": null,
6      "updated_at": null
7    },
8    {
9      "id": 2,
10     "libelle": "OCI",
11     "created_at": null,
12     "updated_at": null
13   },
14   {
15     "id": 3,
16     "libelle": "Visiteur",
17     "created_at": null,
18     "updated_at": null
19   },
20   {
21     "id": 4,
22     "libelle": "Personnel",
23     "created_at": null,
24     "updated_at": null
25   }
26 }
```

### 2- Liste type info

Ce service permet d'obtenir la liste des types d'info.

**VERBE HTTP** => GET

**URL** => http://adresse\_du\_serveur/api/v1/type\_info

**SUCCES** => Retourne la liste des types d'info (Tableau d'objet JSON).

**ERREUR** => Retourne :

**500** => Erreur interne (Une erreur s'est produite ou le serveur est injoignable).

## EXEMPLE RETOURNE EN CAS DE SUCCES

```
1  {
2    {
3      "id": 1,
4      "libelle": "Piers",
5      "created_at": null,
6      "updated_at": null
7    },
8    {
9      "id": 2,
10     "libelle": "Photo",
11     "created_at": null,
12     "updated_at": null
13   },
14   {
15     "id": 3,
16     "libelle": "Qrcode",
17     "created_at": null,
18     "updated_at": null
19   }
20 }
```