

API Rest Example

Vue générale

API permettant de restituer des informations sur les clients.

Information de version

Version : 1.0.0

Information de contact

Contact : Support

Information de licence

Licence : License of API

Licence URL : API license URL

Conditions de service : Terms of service

Schéma d'URI

Serveur : localhost:8080

Chemin de base : /

Tags

- *person-resource* : API pour la gestion des personnes.

Ressources

Person-resource

API pour la gestion des personnes.

Permet d'enregistrer une nouvelle personne

POST /person

Description

Enregistre la personne en base de données. Le resultat de la requete contient une entete Location indiquant l'url pour la récupération du nouveau client.

Paramètres

Type	Nom	Description	Schéma
Body	person <i>requis</i>	person	Person

Réponses

Code HTTP	Description	Schéma
200	OK	Pas de contenu
201	Entité créée	Pas de contenu
401	Unauthorized	Pas de contenu
403	Forbidden	Pas de contenu
404	Not Found	Pas de contenu
500	Erreur interne serveur	Pas de contenu

Consomme

- `application/json`

Produit

- `application/json`

Exemple de requête HTTP

Requête path

```
/person
```

Requête body

```
{
  "identifiant" : "string",
  "civilite" : "string",
  "nom" : "string",
  "nomMarital" : "string",
  "prenom" : "string",
  "adresse" : {
    "adresse1" : "string",
    "adresse2" : "string",
    "cp" : "string",
    "pays" : "string",
    "ville" : "string"
  }
}
```

Retourne l'ensemble des personnes

GET /person

Réponses

Code HTTP	Description	Schéma
200	OK	< Person > array
401	Unauthorized	Pas de contenu
403	Forbidden	Pas de contenu
404	Not Found	Pas de contenu

Consomme

- `application/json`

Produit

- `application/json`

Exemple de requête HTTP

Requête path

/person

Exemple de réponse HTTP

Réponse 200

```
[ {
  "identifiant" : "string",
  "civilite" : "string",
  "nom" : "string",
  "nomMarital" : "string",
  "prenom" : "string",
  "adresse" : {
    "adresse1" : "string",
    "adresse2" : "string",
    "cp" : "string",
    "pays" : "string",
    "ville" : "string"
  }
} ]
```

Permet de rechercher une personne en fonction de son identifiant

```
GET /person/{id}
```

Paramètres

Type	Nom	Description	Schéma
Path	id <i>requis</i>	Identifiant fonctionnel de la personne recherchée	string

Réponses

Code HTTP	Description	Schéma
200	OK	Person
401	Unauthorized	Pas de contenu
403	Forbidden	Pas de contenu
404	Not Found	Pas de contenu

Consomme

- `application/json`

Produit

- `application/json`

Exemple de requête HTTP

Requête path

```
/person/string
```

Exemple de réponse HTTP

Réponse 200

```
{
  "identifiant" : "string",
  "civilite" : "string",
  "nom" : "string",
  "nomMarital" : "string",
  "prenom" : "string",
  "adresse" : {
    "adresse1" : "string",
    "adresse2" : "string",
    "cp" : "string",
    "pays" : "string",
    "ville" : "string"
  }
}
```

Définitions

Adresse

Nom	Description	Schéma
adresse1 <i>requis</i>	Adresse (rue, avenue, boulevard) Exemple : <code>"string"</code>	string
adresse2 <i>optionnel</i>	Information complémentaire sur l'adresse Exemple : <code>"string"</code>	string
cp <i>requis</i>	Code Postal Exemple : <code>"string"</code>	string

Nom	Description	Schéma
pays <i>requis</i>	Pays Exemple : "string"	string
ville <i>requis</i>	Ville Exemple : "string"	string

Iterable«Person»

Type : object

Optional«Person»

Nom	Description	Schéma
present <i>optionnel</i>	Exemple : true	boolean

Person

Entité représentant une personne physique

Nom	Description	Schéma
adresse <i>optionnel</i>	Adresse de la personne Exemple : Adresse	Adresse
civilite <i>optionnel</i>	Civilite Exemple : "string"	string
identifiant <i>optionnel</i>	Identifiant Exemple : "string"	string
nom <i>requis</i>	Nom Exemple : "string"	string
nomMarital <i>optionnel</i>	Nom marital Exemple : "string"	string
prenom <i>requis</i>	Prenom Exemple : "string"	string