

1. Présentation de notre API

Notre API permet de gérer des clients(utilisateurs) et les annonces en utilisant les méthodes HTTP telles que: GET,PATCH,PUT,DELETE et POST.

client

Comme énoncé précédemment, cette api permet de gérer les annonces. Elle possède les propriétés et les contraintes suivantes:

username: le nom d'utilisateur du client. Il ne peut être vide et doit être unique.

password: le mot de passe du client. Il ne peut être vide et doit respecter le format des mot de passe

nom: nom du client. Il ne peut être vide

prenoms: les prénoms du client. Il ne peut être vide

enabled: Montre si le client est active ou non

accountExpired: Montre si le compte du client a été expirée ou pas

accountLocked: Montre si le compte de l'utilisateur est verrouillé ou pas

passwordExpired: indique si le mot de passe du client a été expirée ou pas

Annonce:

Comme énoncé précédemment, cette api permet de gérer les annonces. Elle possède les propriétés et les contraintes suivantes:

titre: Le titre de l'annonce. Ce dernier ne peut être vide, ne peut dépasser plus de 50 caractères et doit être renseigné.

description: La description de l'annonce. Cette propriété ne peut être vide et doit être renseignée

price: le prix de l'annonce. Cette propriété ne peut être inférieure à 0 et doit être renseigné

isActive: Montre si l'annonce est active ou pas

dateCreated: C est la date de création de l'annonce

lastUpdated: la date de la dernière mise à jour de l'annonce

illustrations: la liste des illustrations associées à l'annonce. A savoir qu'une annonce peut avoir plusieurs illustrations(hasMany) mais n'appartient qu'à un seul utilisateur(belongsToMany).

author: l'auteur de l'annonce

2. Structure des données

Toutes les données sont envoyées et reçues en format JSON

3. Ses fonctionnalités

Récupération d'une liste d'annonce

Pour récupérer une liste d'annonce, envoyez une requête GET à l'endpoint suivant sans données:

<http://localhost:8080/api/annonces>

Vous recevrez pour réponse un format JSON qui contiendra les informations des annonces.

Par exemple:

1. Récupération de la liste des annonces

2. Une requête HTTP GET est envoyée à l'URL "<http://localhost:8080/api/annonces>" sans données.

3. Le serveur envoie une réponse avec le code de statut HTTP 200(OK) et le contenu suivant:

```
[
  {
    "id": 1,
    "title": "Titre annonce 1 de nom1",
    "isActive": false,
    "dateCreated": "2023-01-08T17:35:05Z",
    "price": 100,
    "lastUpdated": "2023-01-08T17:35:05Z",
    "illustrations": [
      {
        "id": 1
      },
      {
        "id": 2
      },
      {
        "id": 3
      },
      {
        "id": 4
      },
      {
        "id": 5
      }
    ],
    "description": "Description annonce 1 de nom1",
    "author": {
      "id": 4
    }
  }
]
```

```
    },
    {"id": 2,
      "title": "Titre annonce 2 de nom1",
      "isActive": false,
      "dateCreated": "2023-01-08T17:35:05Z",
      "price": 200,
      "lastUpdated": "2023-01-08T17:35:05Z",
      "illustrations": [
        {
          "id": 6
        },
        {
          "id": 7
        },
        {
          "id": 8
        },
        {
          "id": 9
        },
        {
          "id": 10
        }
      ],
      "description": "Description annonce 2 de nom1",
      "author": {
        "id": 4
      }
    },
    {
      "id": 3,
      "title": "Titre annonce 3 de nom1",
      "isActive": false,
      "dateCreated": "2023-01-08T17:35:05Z",
      "price": 300,
      "lastUpdated": "2023-01-08T17:35:05Z",
      "illustrations": [
```

```

{
  "id": 11
},
{
  "id": 12
},
{
  "id": 13
},
{
  "id": 14
},
{
  "id": 15
}
],
"description": "Description annonce 3 de nom1",
"author": {
  "id": 4
}
}
]

```

Récupération d'une annonce

Pour récupérer une annonce, envoyez une requête GET à l'endpoint suivant sans données:

<http://localhost:8080/api/annonce/{id}>

A noter que {id} est l'identifiant du client que vous voulez récupérer.

Vous recevrez pour réponse un format JSON qui contiendra les informations de l'annonce.

Par exemple:

1. Récupération de l'annonce 1.

2. Une requête HTTP GET est envoyée à l'URL "<http://localhost:8080/api/annonce/1>" sans données.

3. Le serveur envoie une réponse avec le code de statut HTTP 200(OK) et le contenu suivant:

```

{
  "id": 1,

```

```
"title": "Titre annonce 1 de nom1",
"isActive": false,
"dateCreated": "2023-01-08T17:35:05Z",
"price": 100,
"lastUpdated": "2023-01-08T17:35:05Z",
"illustrations": [
  {
    "id": 1
  },
  {
    "id": 2
  },
  {
    "id": 3
  },
  {
    "id": 4
  },
  {
    "id": 5
  }
],
"description": "Description annonce 1 de nom1",
"author": {
  "id": 4
}}
```

Création d'une annonce

Pour créer une annonce, envoyez une requête POST à l'endpoint suivant avec les données nécessaires:

<http://localhost:8080/api/annonces>

Vous recevrez pour réponse un format JSON qui contiendra les informations de l'annonce.

Par exemple:

1.Création d'une annonce

2.Une requête HTTP POST est envoyée à l'URL "<http://localhost:8080/api/annonces>"

avec les données suivantes:

```
{
  "title": "Le guide de developpeur",
  "price": 500,
  "isActive": true,
  "description": "Pour les débutant en informatique",
  "author.id": 5
}
```

3. Le serveur envoie une réponse avec le code de statut HTTP 201 (Créé) et le contenu suivant qui inclut les informations du nouvel utilisateur, telles que son ID, son titre, son prix...

```
{
  "id": 101,
  "title": "Le guide de developpeur",
  "isActive": true,
  "dateCreated": "2023-01-08T20:09:32Z",
  "price": 500,
  "lastUpdated": "2023-01-08T20:09:32Z",
  "illustrations": null,
  "description": "Pour les débutant en informatique",
  "author": {
    "id": 5
  }
}
```

Mise à jour d'une annonce

Pour mettre à jour une annonce, envoyez une requête PUT à l'endpoint suivant avec les données nécessaires:

<http://localhost:8080/api/annonce/{id}>

A noter que {id} est l'identifiant du client que vous voulez modifier.

Par exemple:

1. Mise à jour de l'annonce d'id 6

2. Une requête HTTP PUT est envoyée à l'URL "<http://localhost:8080/api/annonce/6>" avec les données suivantes:

```
{
  "title": "Le guide de developpeur pro",
  "price": 1500,
  "isActive": true,
  "description": "Pour les expert en informatique",
  "author": {
    "id": 7
  }
}
```

3. Le serveur envoie une réponse avec le code de statut HTTP 200(OK) et le contenu suivant

```
{
  "id": 6,
  "title": "Le guide de developpeur pro",
  "isActive": true,
  "dateCreated": "2023-01-08T17:35:05Z",
  "price": 1500,
  "lastUpdated": "2023-01-08T20:25:46Z",
  "illustrations": [
    {
      "id": 26
    },
    {
      "id": 27
    },
    {
      "id": 28
    },
    {
      "id": 29
    },
    {
      "id": 30
    }
  ]
}
```

```
}
],
"description": "Pour les expert en informatique",
"author": {
  "id": 7
}
}
```

Modification partielle des informations d'une annonce

Pour modifier partiellement les informations d'une annonce , envoyez une requête PATCH à l'endpoint suivant et les donnees necessaires:

<http://localhost:8080/api/annonce/{id}>

A noter que {id} est l'identifiant du client que vous voulez modifier partiellement.

Par exemple:

1.Mise à jour de l'auteur et du prix de l'annonce d'id 6

2.Une requête HTTP PATCH est envoyée à l'URL "<http://localhost:8080/api/annonce/6>" avec les données suivantes:

```
{
  "price": 800,
  "author": {
    "id": 4
  }
}
```

3.Le serveur envoie une réponse avec le code de statut HTTP 200(OK) avec le contenu suivant:

```
{
  "id": 6,
  "title": "Titre annonce 1 de nom2",
  "isActive": true,
  "dateCreated": "2023-01-08T17:35:05Z",
  "price": 800,
  "lastUpdated": "2023-01-08T20:18:16Z",
```



```
"illustrations": [  
  {  
    "id": 26  
  },  
  {  
    "id": 27  
  },  
  {  
    "id": 28  
  },  
  {  
    "id": 29  
  },  
  {  
    "id": 30  
  }  
],  
"description": "Description annonce 1 de nom2",  
"author": {  
  "id": 4  
}
```

Suppression d'une annonce

Pour supprimer une annonce, envoyez une requête DELETE à l'endpoint suivant sans données:

<http://localhost:8080/api/annonce/{id}>

A noter que {id} est l'identifiant de l'annonce que vous voulez supprimer.

Par exemple:

1. Suppression de l'utilisateur 10

2. Une requête HTTP DELETE est envoyée à l'URL "<http://localhost:8080/api/annonce/10>" sans données.

3. Le serveur envoie une réponse avec le code de statut HTTP 200 (OK) et aucun contenu. Ainsi l'annonce à l'ID 10 a été supprimé avec succès.

Récupération des informations d'un utilisateur

Pour récupérer les informations d'un utilisateur , envoyez une requête GET à l'endpoint suivant:

<http://localhost:8080/api/user/id>

A noter que {id} est l'identifiant du client que vous voulez récupérer.

Vous recevrez pour réponse un format JSON qui contiendra les informations du client.

Mise à jour des informations d'un utilisateur

Pour mettre à jour les informations d'un utilisateur , envoyez une requête PUT à l'endpoint suivant avec les données nécessaires:

<http://localhost:8080/api/user/{id}>

A noter que {id} est l'identifiant du client que vous voulez modifier.

Envoyez les nouvelles données du client dans la requête au format JSON.

Par exemple:

1.Mise à jour du nom et prénoms de l'utilisateur 6.

2.Une requête HTTP PUT est envoyée à l'URL “<http://localhost:8080/api/user/6>” avec les données suivantes:

```
{  
  "nom":"Dje Bi",  
  "prenoms":"Mointi"  
}
```

3.Le serveur envoie une réponse avec le code de statut HTTP 200(OK) et le contenu suivant qui inclut les informations de mises à jour de l'utilisateur, telles que son ID, son nom d'utilisateur, son nom , ses prénoms, sa date de création:

```
{"id":6,  
"dateCreated":"2023-01-08T17:35:05Z",  
"passwordExpired":false,  
"lastUpdated":"2023-01-08T17:35:05Z",  
"photo":"users/3.jpeg",  
"accountExpired":false,
```

```
"nom":"Dje Bi",
"annonces":[{"id":13},
{"id":12},
{"id":14},
{"id":15},
{"id":11}],
"prenoms":"Mointi",
"username":"client3",
"accountLocked":false,
"password":"$2a$10$UjhKv9.3CZMjF.T.xBNO/eH6W8kJIZjR0UwCsuZtAuaV2M6s/Yg16"
,
"enabled":true}
```

Création d'un utilisateur

Pour créer un utilisateur , envoyez une requête POST à l'endpoint suivant:

<http://localhost:8080/api/users>

Par exemple:

1.Création d'un utilisateur

2.Une requête HTTP POST est envoyée à l'URL “<http://localhost:8080/api/users>” avec les données suivantes:

```
{
  "password":"admin",
  "nom":"Benoit",
  "prenoms":"Linsay",
  "username":"lili3",
  "photo":"avatar.png"
}
```

3.Le serveur envoie une réponse avec le code de statut HTTP 201 (Créé) et le contenu suivant qui inclut les informations du nouvel utilisateur, telles que son ID, son nom d' utilisateurs, son nom , ses prénoms, sa photo:

```
{
  "id": 26,
```

```
"dateCreated": "2023-01-08T18:28:04Z",
"passwordExpired": false,
"lastUpdated": "2023-01-08T18:28:04Z",
"photo": "avatar.png",
"accountExpired": false,
"nom": "Benoit",
"annonces": null,
"prenoms": "Linsay",
"username": "lili3",
"accountLocked": false,
"password":
"$2a$10$lZIXCmQrXMfQPKxKkwKsa.ifGIM4zd/lHI2r5O2fF.AInD1d2BX2m",
"enabled": true
}
```

Modification partielle des informations d'un client

Pour modifier partiellement les informations d'un utilisateur , envoyez une requête PATCH à l'endpoint suivant:

<http://localhost:8080/api/user/{id}>

A noter que {id} est l'identifiant du client que vous voulez modifier partiellement. Envoyez les nouvelles données du client dans la requête au format JSON.

Par exemple:

1.Mise à jour de l'utilisateur 19

2.Une requête HTTP PATCH est envoyée à l'URL "<http://localhost:8080/api/user/19>" avec les données suivantes:

```
{
  "nom":"Dje Lou"
}
```

3.Le serveur envoie une réponse avec le code de statut HTTP 200(OK).

Suppression des informations d'un client

Pour supprimer les informations d'un utilisateur , envoyez une requête DELETE à l'endpoint suivant sans données:

<http://localhost:8080/api/user/{id}>

A noter que {id} est l'identifiant du client que vous voulez supprimer.

Par exemple:

1. Suppression de l'utilisateur 19

2. Une requête HTTP DELETE est envoyée à l'URL "<http://localhost:8080/api/user/19>" sans données.

3. Le serveur envoie une réponse avec le code de statut HTTP 200 (OK) et aucun contenu. Ainsi l'utilisateur à l'ID 19 a été supprimé avec succès.

Récupération d'une liste d'utilisateurs

Pour récupérer une liste d'utilisateurs, envoyez une requête GET à l'endpoint suivant sans données:

<http://localhost:8080/api/users>

Vous recevrez pour réponse un format JSON qui contiendra les informations des utilisateurs.

Erreur et statut HTTP

Ci-dessous se trouvent les codes d'erreurs et de statuts HTTP que l'API peut renvoyer:

'500' : erreur de serveur

'404 notFound': Si l'instance n'est pas trouvée. Accès interdit

'400 error' : Erreur de la requête si aucun identifiant est passé. Il peut intervenir lorsqu'il y'a un problème lors de la suppression, mise à jour ou insertion d'un élément

'201' : pour la création

'200': Lorsque l'instance est trouvée. Tout est ok