

Introduction et Préparation

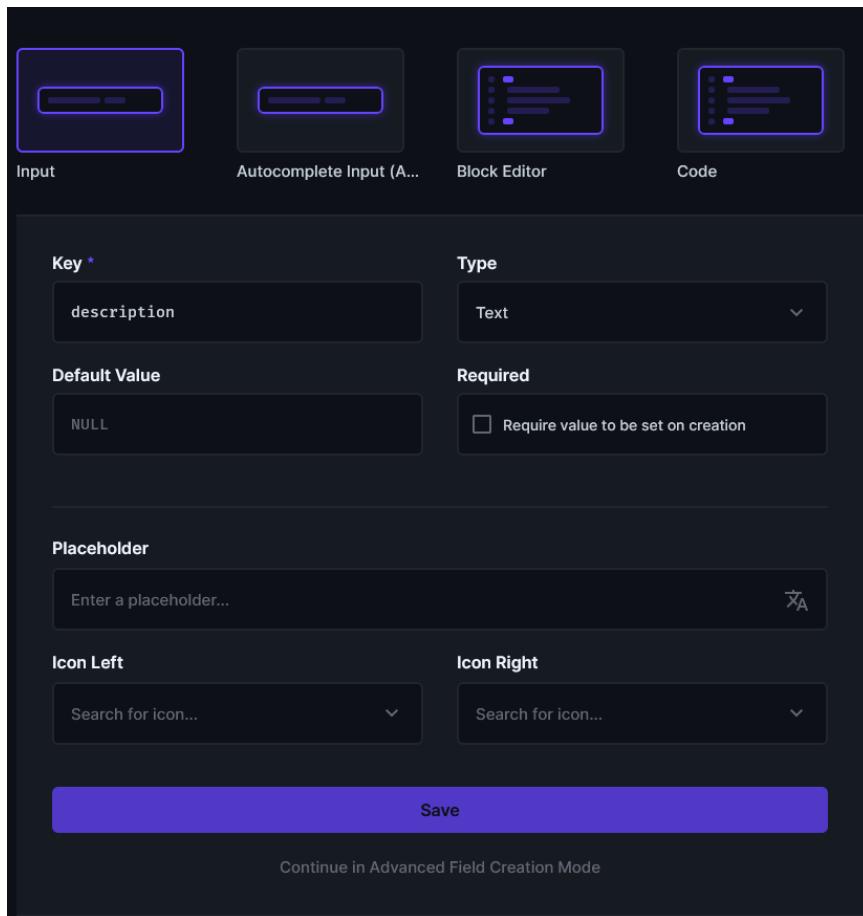
L'objectif de ce TP est de mettre en place un backend de gestion pour des praticiens de santé sans avoir à coder toute l'infrastructure serveur, en utilisant le CMS Directus.

Lien GitHub du projet : https://github.com/Iclerc561/DWM_ParadigmeBDD.git

L'installation a été réalisée via Docker. Après configuration, l'outil est accessible en local. Toutes les requêtes API REST présentées ci-dessous ont été effectuées via des requêtes GET sur l'adresse de base : <http://localhost:8055/items/>

I. Création et Modélisation des Données

1. Modélisation (Schéma UML) : Nous avons reproduit le modèle de données relationnel (Praticien, Spécialité, Structure, etc.) directement dans l'interface de Directus. Nous avons configuré les types de champs (UUID pour les IDs, chaînes de caractères, booléens) et les relations (One-to-Many entre Structure et Praticien, Many-to-One entre Praticien et Spécialité). [Insérer l'image du diagramme UML fourni]



Creating New Collection

Name the collection and setup its unique "key" field...

Name * Praticien

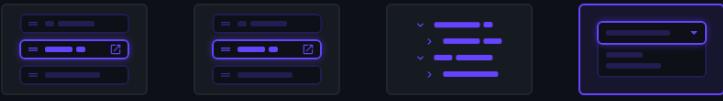
Singleton Treat as single object

Collection names are case sensitive

Primary Key Field id

Type Generated UUID

Création d'associations:



Many to Many One to Many Tree View Many to One

Key * specialite

Type Integer

Default Value NULL

Required Require value to be set on creation

Related Collection * Specialite

Display Template

Creating Items Enable Create Button

Selecting Items Enable Select Button

Filter No configured rules

Add Filter ▾

Item link Show a link to the item

2. Import des données : Pour peupler la base, nous avons utilisé la fonction d'import CSV de Directus, en veillant à respecter l'ordre des contraintes (importer les structures et spécialités avant les praticiens).

The screenshot shows the Directus Content Management System interface. At the top, there's a navigation bar with icons for user management, content types, and search. The current page is 'Content > Praticien2motif'. On the right side, a sidebar titled 'Layout Options' includes 'Auto Refresh' and 'Import / Export' sections. The 'Import' section shows a file named 'praticien2motifs.csv' (XLS format) selected, with a 'Start Import' button below it. A 'Export Items' button is also present. Below the sidebar, a large dark area displays a circular icon with an upward arrow and the text 'No Items' followed by the message 'There are no items in this collection yet.' A 'Create Item' button is located at the bottom of this area.

3. Configuration des droits (Sécurité) : Afin de pouvoir interroger l'API publiquement pour ce test, nous avons modifié la "Public Policy" pour autoriser la lecture (Read) sur les collections concernées.

The screenshot shows the 'Access Policies' section with the 'Public Policy' selected. The 'Policy Name' is set to 'Public' and has a 'Public' icon. The 'Description' field contains the text: 'Controls what API data is available without authenticating.' In the 'Permissions' section, there is a table mapping collections to actions. The 'Actions' column for all collections includes 'Create', 'Read', 'Update', 'Delete', 'Share', and a delete 'X' button. The 'Read' button is highlighted in blue for all collections listed.

Collection	Actions
MotifVisite	Create, Read, Update, Delete, Share, X
MoyenPaiement	Create, Read, Update, Delete, Share, X
Praticien	Create, Read, Update, Delete, Share, X
Specialite	Create, Read, Update, Delete, Share, X
Structure	Create, Read, Update, Delete, Share, X
praticien2motif	Create, Read, Update, Delete, Share, X
praticien2moyen	Create, Read, Update, Delete, Share, X

Add Collection ▾

II. Interrogation de l'API REST

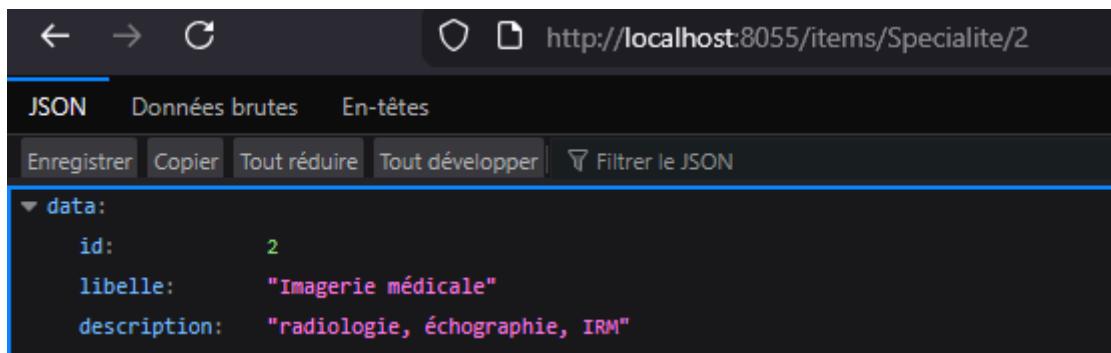
1. Liste des praticiens : Nous récupérons l'intégralité de la collection Praticien sans filtre.

GET <http://localhost:8055/items/Praticien>

```
JSON  Données brutes  En-têtes
Enregistrer Copier Tout réduire Tout développer Filtrer le JSON
▼ data:
  ▼ 0:
    id: "03f24d7b-efda-3fbe-ae53-0c0c0edef089"
    nom: "Grenier"
    prenom: "Josette"
    email: "Josette.Grenier@wanadoo.fr"
    telephone: "06 43 92 58 43"
    ville: "Moreno"
    rpps_id: "72002111175"
    titre: "Dr."
    organisation: false
    accepte_nouveau_patient: true
    specialite: 2
    structure: "c302d11b-d5b8-310a-bbc5-9dd4400eb2c9"
    ▼ motif_visite:
      0: 79
      1: 80
    ▼ praticien2moyen:
      0: 92
  ▼ 1:
    id: "05644ac0-c4cb-36bf-a0cf-409075eb20c4"
    nom: "Leduc"
    prenom: "Aimée"
    email: "Aime.Leduc@laposte.net"
    telephone: "+33 (0)4 85 19 99 00"
    ville: "Carre-sur-Turpin"
    rpps_id: "14015596432"
    titre: "Dr."
    organisation: false
    accepte_nouveau_patient: true
    specialite: 4
    structure: "5f7d3d80-3c4b-386a-a23e-74de4a1e0b19"
    ▼ motif_visite:
      0: 60
      1: 61
    ▼ praticien2moyen:
      0: 67
      1: 68
      2: 69
      3: 70
      4: 71
      5: 72
  ▼ 2:
    id: "0768e78b-12c8-3694-b89b-d0a7451a8fb1"
    nom: "Descamps"
    prenom: "Alexandria"
    email: "Alexandria.Descamps@club-internet.fr"
    telephone: "+33 (0)1 75 74 98 33"
    ville: "Joubert"
    rpps_id: "79003481902"
    titre: "Dr."
    organisation: false
    accepte_nouveau_patient: true
    specialite: 4
    structure: "372c1de7-4d31-3d01-9c92-77e69467d4e3"
    ▼ motif_visite:
      0: 28
    ▼ praticien2moyen:
      0: 29
      1: 30
  ▼ 3:
    id: "085c44e7-6245-3b10-9c1a-92f09524ef2a"
    nom: "Dumont"
    prenom: "Pierre"
    email: "Pierre.Dumont@club-internet.fr"
    telephone: "+33 (0)1 75 74 98 33"
    ville: "Joubert"
    rpps_id: "79003481902"
    titre: "Dr."
    organisation: false
    accepte_nouveau_patient: true
    specialite: 4
    structure: "372c1de7-4d31-3d01-9c92-77e69467d4e3"
```

2. Récupérer une spécialité par son ID : Nous ciblons un élément précis (l'ID 2) en l'ajoutant à l'URL.

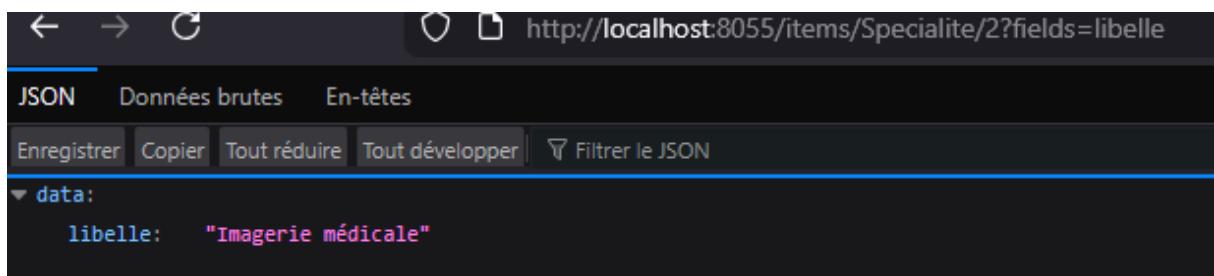
GET <http://localhost:8055/items/Specialite/2>



```
JSON Données brutes En-têtes
Enregistrer Copier Tout réduire Tout développer Filtrer le JSON
▼ data:
  id: 2
  libelle: "Imagerie médicale"
  description: "radiologie, échographie, IRM"
```

3. La spécialité d'ID 2 (uniquement le libellé) : Pour optimiser la réponse, nous utilisons le paramètre fields pour ne demander que le champ libelle.

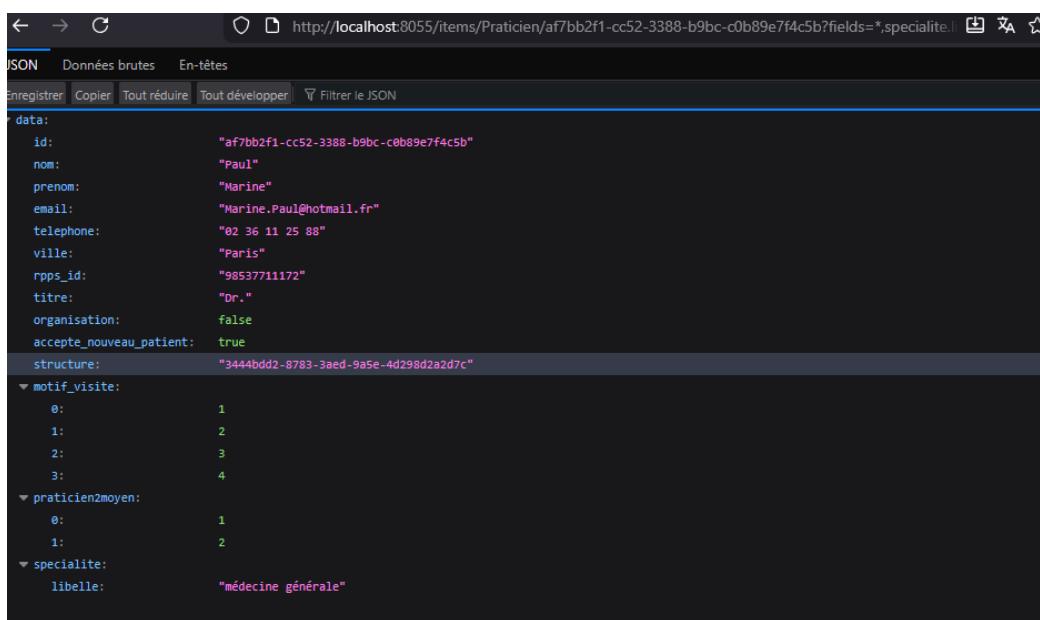
GET <http://localhost:8055/items/Specialite/2?fields=libelle>



```
JSON Données brutes En-têtes
Enregistrer Copier Tout réduire Tout développer Filtrer le JSON
▼ data:
  libelle: "Imagerie médicale"
```

4. Un praticien avec sa spécialité (libellé uniquement) : Nous récupérons un praticien par son ID et demandons le détail de sa relation specialite. L'étoile * récupère tous les champs du praticien, et le point ". " permet de naviguer dans la relation.

GET http://localhost:8055/items/Praticien/<ID_PRATICIEN>?fields=*,specialite.libelle

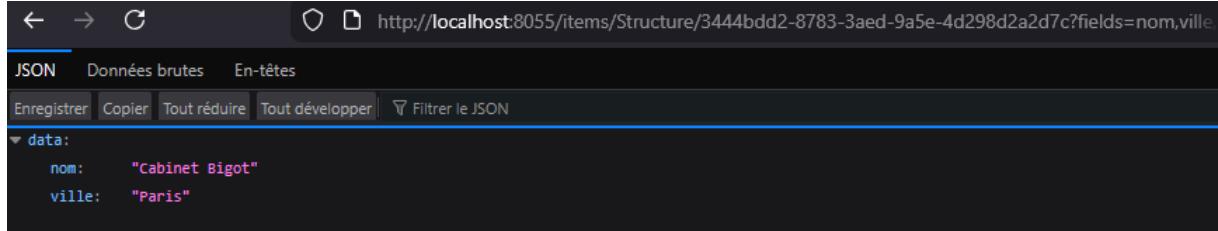


```
JSON Données brutes En-têtes
Enregistrer Copier Tout réduire Tout développer Filtrer le JSON
▼ data:
  id: "af7bb2f1-cc52-3388-b9bc-c0b89e7f4c5b"
  nom: "Paul"
  prenom: "Marine"
  email: "Marine.Paul@hotmail.fr"
  telephone: "02 36 11 25 88"
  ville: "Paris"
  rpps_id: "98537711172"
  titre: "Dr."
  organisation: false
  accepte_nouveau_patient: true
  structure: "3444bdd2-8783-3aed-9a5e-4d298d2a2d7c"
  ▼ motif_visite:
    0: 1
    1: 2
    2: 3
    3: 4
  ▼ praticien2moyen:
    0: 1
    1: 2
  ▼ specialite:
    libelle: "médecine générale"
```

5. Une structure et ses praticiens rattachés : Nous partons d'une structure et demandons la liste inversée des praticiens qui y travaillent.

GET

http://localhost:8055/items/Structure/<ID_STRUCTURE>?fields=nom,ville,praticiens.nom,praticiens.prenom



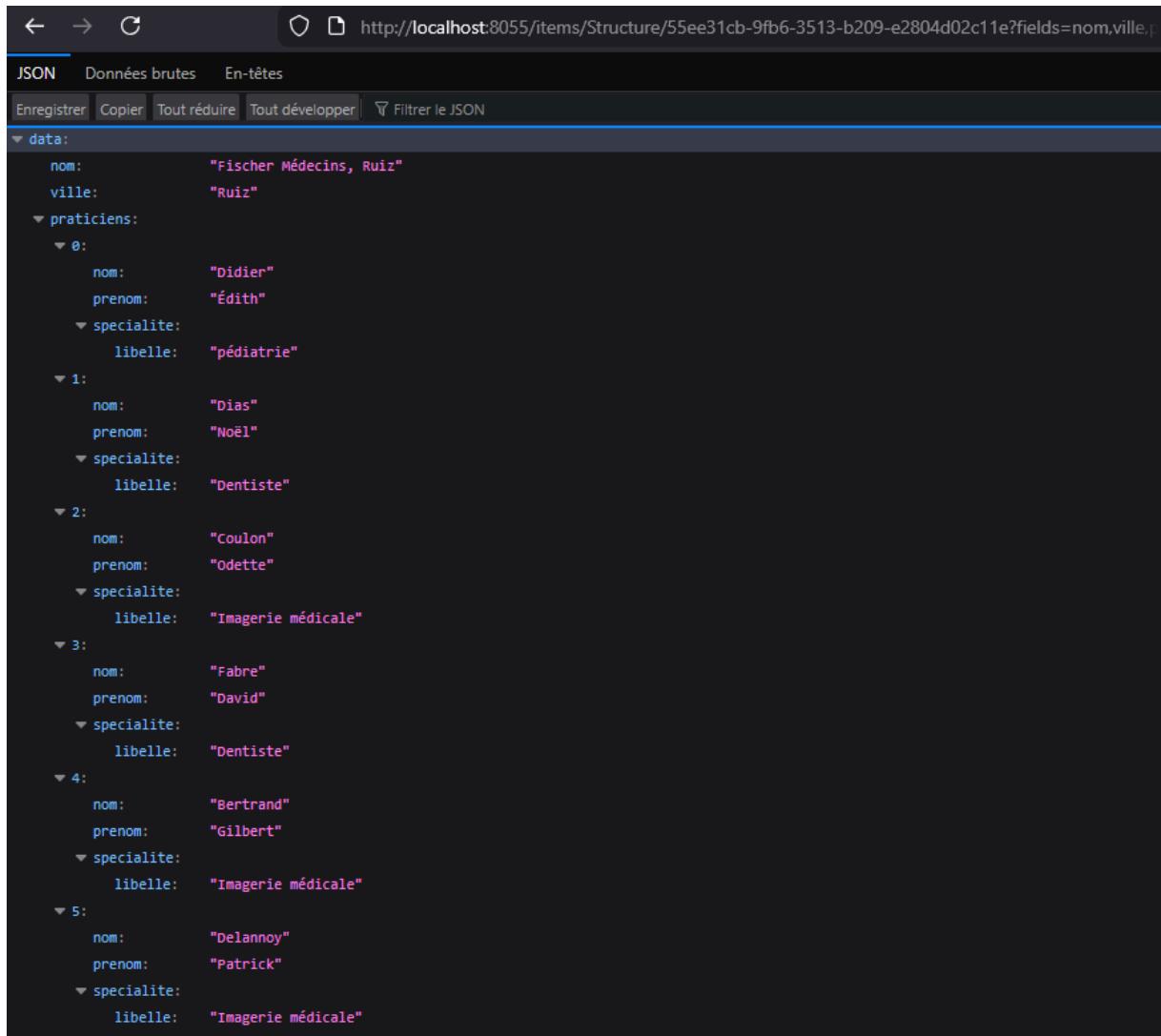
The screenshot shows a browser's developer tools JSON viewer. The URL is `http://localhost:8055/items/Structure/3444bdd2-8783-3aed-9a5e-4d298d2a2d7c?fields=nom,ville`. The JSON response is:

```
data:
  nom: "Cabinet Bigot"
  ville: "Paris"
```

6. Structure, praticiens et leurs spécialités (Imbrication multiniveau) : Nous allons un cran plus loin en demandant le libellé de la spécialité pour chaque praticien de la structure.

GET

http://localhost:8055/items/Structure/<ID_STRUCTURE>?fields=nom,ville,praticiens.nom,praticiens.prenom,praticiens.specialite.libelle



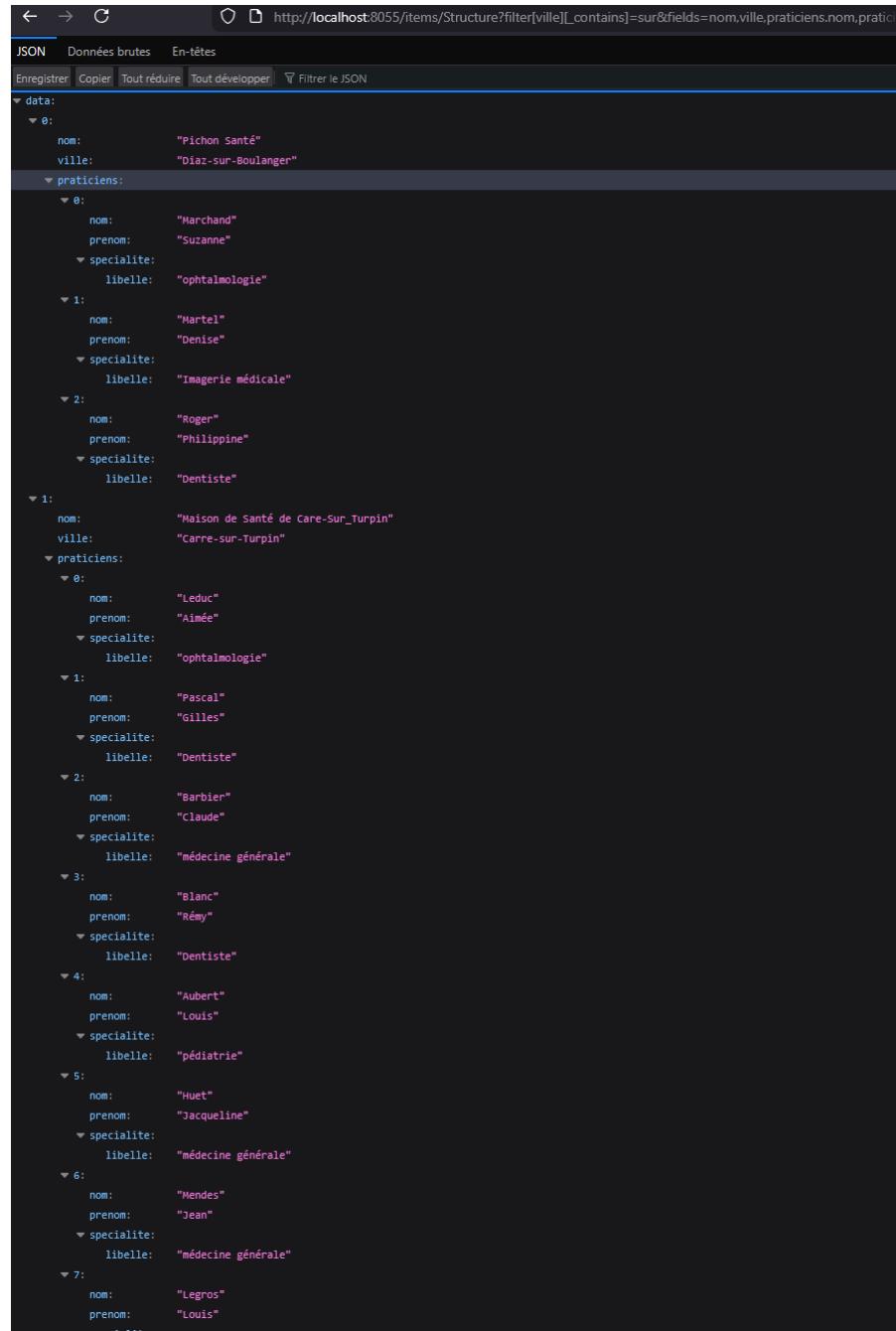
The screenshot shows a browser's developer tools JSON viewer. The URL is `http://localhost:8055/items/Structure/55ee31cb-9fb6-3513-b209-e2804d02c11e?fields=nom,ville`. The JSON response is:

```
data:
  nom: "Fischer Médecins, Ruiz"
  ville: "Ruiz"
  praticiens:
    ▼ 0:
      nom: "Didier"
      prenom: "Édith"
      ▼ specialite:
        libelle: "pédiatrie"
    ▼ 1:
      nom: "Dias"
      prenom: "Noël"
      ▼ specialite:
        libelle: "Dentiste"
    ▼ 2:
      nom: "Coulon"
      prenom: "Odette"
      ▼ specialite:
        libelle: "Imagerie médicale"
    ▼ 3:
      nom: "Fabre"
      prenom: "David"
      ▼ specialite:
        libelle: "Dentiste"
    ▼ 4:
      nom: "Bertrand"
      prenom: "Gilbert"
      ▼ specialite:
        libelle: "Imagerie médicale"
    ▼ 5:
      nom: "Delannoy"
      prenom: "Patrick"
      ▼ specialite:
        libelle: "Imagerie médicale"
```

7. Recherche textuelle (Filtre) Nous filtrons les structures dont le champ `ville` contient la chaîne de caractères "sur".

GET

[http://localhost:8055/items/Structure?filter\[ville\]\[_contains\]=sur&fields=nom,ville,praticiens.nom,praticiens.prenom,praticiens.specialite.libelle](http://localhost:8055/items/Structure?filter[ville][_contains]=sur&fields=nom,ville,praticiens.nom,praticiens.prenom,praticiens.specialite.libelle)



The screenshot shows a JSON viewer interface with the following details:

- URL: `http://localhost:8055/items/Structure?filter[ville][_contains]=sur&fields=nom,ville,praticiens.nom,praticiens.prenom,praticiens.specialite.libelle`
- JSON View:

```
data:
  0:
    nom: "Pichon Santé"
    ville: "Diaz-sur-Boulanger"
    praticiens:
      0:
        nom: "Marchand"
        prenom: "Suzanne"
        specialite:
          libelle: "ophtalmologie"
      1:
        nom: "Martel"
        prenom: "Denise"
        specialite:
          libelle: "Imagerie médicale"
      2:
        nom: "Roger"
        prenom: "Philippine"
        specialite:
          libelle: "Dentiste"
    1:
      nom: "Maison de Santé de Care-Sur-Turpin"
      ville: "Carre-sur-Turpin"
      praticiens:
        0:
          nom: "Leduc"
          prenom: "Aimée"
          specialite:
            libelle: "ophtalmologie"
        1:
          nom: "Pascal"
          prenom: "Gilles"
          specialite:
            libelle: "Dentiste"
        2:
          nom: "Barbier"
          prenom: "Claude"
          specialite:
            libelle: "médecine générale"
        3:
          nom: "Blanc"
          prenom: "Cémy"
          specialite:
            libelle: "Dentiste"
        4:
          nom: "Aubert"
          prenom: "Louis"
          specialite:
            libelle: "pédiatrie"
        5:
          nom: "Huet"
          prenom: "Jacqueline"
          specialite:
            libelle: "médecine générale"
        6:
          nom: "Mendes"
          prenom: "Jean"
          specialite:
            libelle: "médecine générale"
        7:
          nom: "Legros"
          prenom: "Louis"
          specialite:
            libelle: "médecine générale"
```