Logo

Description automatically generated

**Gestion des médecins**

**Alexis DUCHEMANN**

**SIO2B**

Sommaire

[Introduction 3](#_Toc99565053)

[Contexte 3](#_Toc99565054)

[Arborescence 4](#_Toc99565055)

[Cas d’utilisation 4](#_Toc99565056)

[Technologies utilisées 4](#_Toc99565057)

[Documents 5](#_Toc99565058)

[Application 6](#_Toc99565059)

[Création d’une API 6](#_Toc99565060)

[Connexion 6](#_Toc99565061)

[Navigation à travers le site 6](#_Toc99565062)

[Consultation des médecins 6](#_Toc99565063)

[Création des médecins / Modifications des médecins 6](#_Toc99565064)

[Affichage des départements 6](#_Toc99565065)

[Affichage des Pays 6](#_Toc99565066)

# Introduction

## Contexte

## Arborescence

Création d’un médecin

1



Page de connexion

Liste des pays

Liste des départements

Liste des médecins

Département par pays

Médecin par département

Information d’un médecin

+ modification

## Cas d’utilisation

## Technologies utilisées

## Documents

Diagram

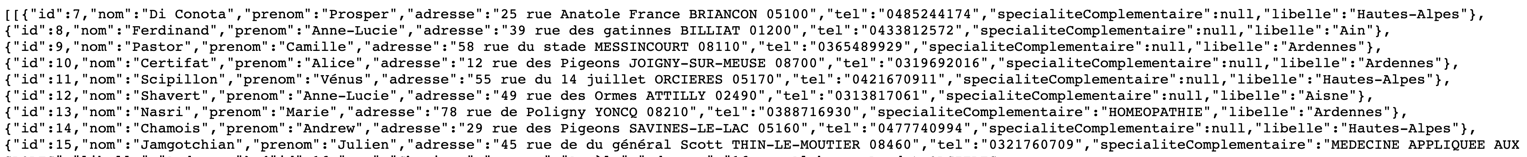
Description automatically generated

# Application

## Création d’une API

* Création d’une API avec express et NodeJS pour l’affichage des objets :

Exemple : requête Get créant les objets médecins et les retournant en JSON



* Récupération des données avec axios sur la partie front :

useEffect(async() => {

const datas = *await* Axios.get('http://localhost:3001/medecins');

setMedecins(datas.data[0]);

setLoading(false);

}, []);

## Connexion

## Navigation à travers le site

## Consultation des médecins

* Affichage des donneés en tableau :

const columns = [

{ field: 'id', headerName: 'Identifiant', flex: 0.3 },

{ field: 'nom', headerName: 'Nom', flex: 0.5 },

{ field: 'prenom', headerName: 'Prénom', flex: 0.5 },

{ field: 'adresse', headerName: 'Adresse', flex: 1},

{ field: 'tel', headerName: 'Téléphone', flex: 0.5 },

{ field: 'specialiteComplementaire', headerName: 'Specialité Complémentaire', flex: 0.6 },

{ field: 'libelle', headerName: 'Département', flex: 0.5 },

{ field: "Modify",

renderCell: (cellValues) => {

const idMedecin = cellValues.row.id;

*return* (<Link className="modify" to={`/medecininfo/${idMedecin}`}>Modifier</Link>)}

},

{ field: "Delete",

renderCell: (cellValues) => {

const idMedecin = cellValues.row.id;

*return* (<Link className="delete" to={`/medecininfo/${idMedecin}`}>Supprimer</Link>)}

}

];

## Création des médecins / Modifications des médecins

## Affichage des départements

## Affichage des Pays