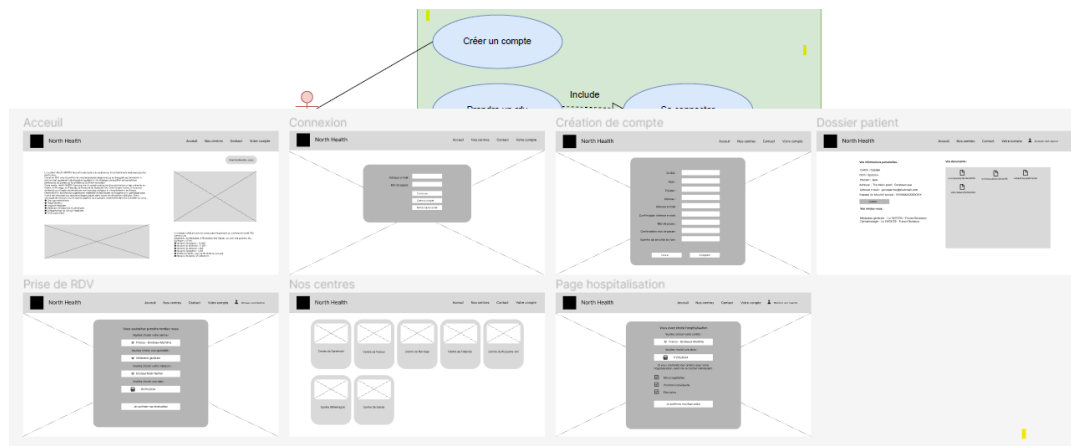


Rapport de mission : Health North

Dossier 1

Mission 1 : Réalisez les maquettes/mockup des différentes IHM (web, mobile et lourde) et réalisez l'intégration

Réalisation des mockups avec le logiciel FIGMA



Réalisation des IHM avec VS code, j'ai fait le choix d'utiliser la bibliothèque de composant bootstrap pour monter en compétence avec son utilisation ainsi que pour gagner du temps dans la conception des IHM.

Mission 2 : Réalisez le ou les modélisations de données (mcd, diagramme UML, etc.)

Diagrammes de cas d'utilisation Web et mobile :

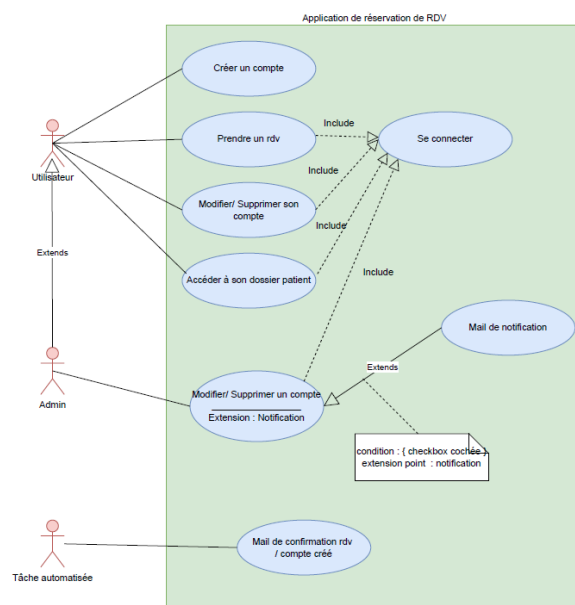
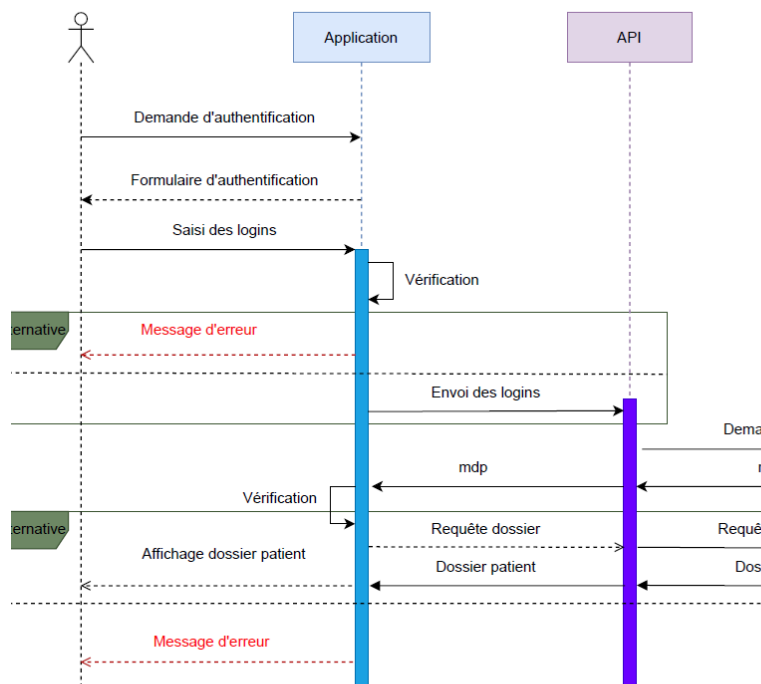
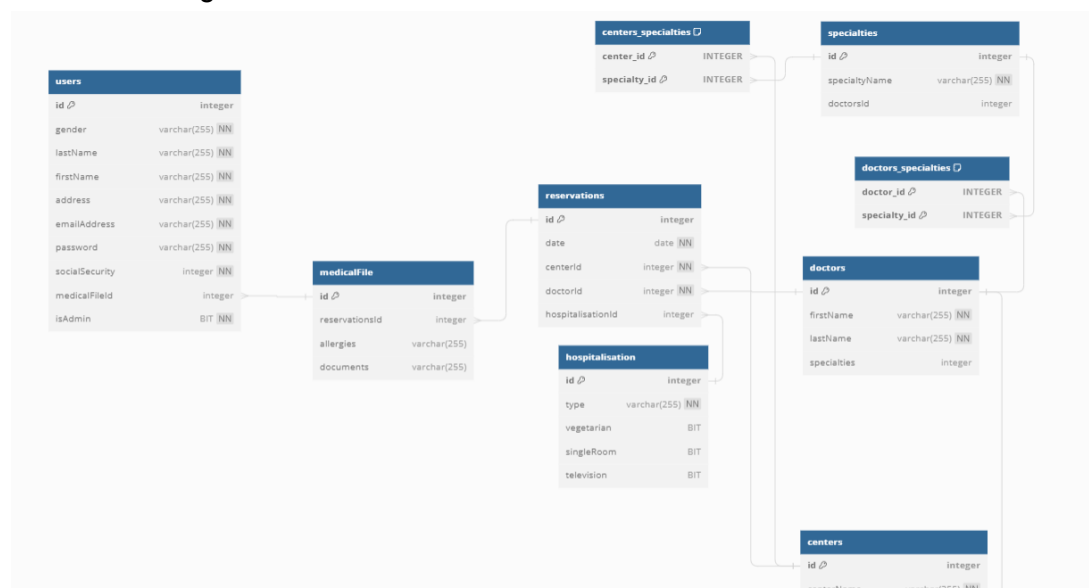


Diagramme de séquence :



Mission 3 : Diagramme de classe



Mission 4 : Générez le script de création de la base de données

Ayant choisi d'utiliser le framework Symfony pour réaliser mon API, j'ai opté pour l'utilisation de l'ORM Doctrine afin de gérer les entités et la base de données de mon application.

Cette technologie permet de générer directement les scripts SQL requis pour créer la base de donnée.

```
1 CREATE TABLE center (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, city VARCHAR(255) NOT NULL, country VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE
2 CREATE TABLE doctor (id INT AUTO_INCREMENT NOT NULL, center_id INT DEFAULT NULL, gender VARCHAR(255) NOT NULL, first_name VARCHAR(255) NOT NULL, last_name VARCHAR(255) NOT NULL, INDEX IDX_1FC0F36
3 CREATE TABLE hospitalisation (id INT AUTO_INCREMENT NOT NULL, type VARCHAR(255) NOT NULL, vegetarian TINYINT(1) NOT NULL, single_room TINYINT(1) NOT NULL, television TINYINT(1) NOT NULL, PRIMARY
4 CREATE TABLE medical_file (id INT AUTO_INCREMENT NOT NULL, allergies VARCHAR(255) DEFAULT NULL, documents VARCHAR(255) DEFAULT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE 'utf8mb
5 CREATE TABLE patient (id INT AUTO_INCREMENT NOT NULL, medical_file_id INT DEFAULT NULL, gender VARCHAR(20) NOT NULL, last_name VARCHAR(255) NOT NULL, first_name VARCHAR(255) NOT NULL, address VAR
6 CREATE TABLE reservation (id INT AUTO_INCREMENT NOT NULL, medical_file_id INT NOT NULL, hospitalisation_id INT DEFAULT NULL, doctor_id INT NOT NULL, center_id INT NOT NULL, date DATE NOT NULL, IN
7 CREATE TABLE specialty (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE 'utf8mb4 unicode-ci' ENGINE = InnoDB;
8 CREATE TABLE specialty_doctor (specialty_id INT NOT NULL, doctor_id INT NOT NULL, INDEX IDX_2DBBE9AC9A353316 (specialty_id), INDEX IDX_2DBBE9AC87F4FB17 (doctor_id), PRIMARY KEY(specialty_id, doct
9 CREATE TABLE specialty_center (specialty_id INT NOT NULL, center_id INT NOT NULL, INDEX IDX_728BF1E29A353316 (specialty_id), INDEX IDX_728BF1E25932F377 (center_id), PRIMARY KEY(specialty_id, cent
10 CREATE TABLE user (id INT AUTO_INCREMENT NOT NULL, email VARCHAR(180) NOT NULL, roles JSON NOT NULL, password VARCHAR(255) NOT NULL, UNIQUE INDEX UNIQ_8D93D649E7927C74 (email), PRIMARY KEY(id)) D
11 ALTER TABLE doctor ADD CONSTRAINT FK_1FC0F36A5932F377 FOREIGN KEY (center_id) REFERENCES center (id);
12 ALTER TABLE patient ADD CONSTRAINT FK_1ADAD7EB05C999A2 FOREIGN KEY (medical_file_id) REFERENCES medical_file (id);
13 ALTER TABLE reservation ADD CONSTRAINT FK_42CB4955D5C999A2 FOREIGN KEY (medical_file_id) REFERENCES medical_file (id);
14 ALTER TABLE reservation ADD CONSTRAINT FK_42CB49555331F4C5 FOREIGN KEY (hospitalisation_id) REFERENCES hospitalisation (id);
15 ALTER TABLE reservation ADD CONSTRAINT FK_42CB495587F4FB17 FOREIGN KEY (doctor_id) REFERENCES doctor (id);
16 ALTER TABLE reservation ADD CONSTRAINT FK_42CB49555932F377 FOREIGN KEY (center_id) REFERENCES center (id);
17 ALTER TABLE specialty_doctor ADD CONSTRAINT FK_2DBBE9AC9A353316 FOREIGN KEY (specialty_id) REFERENCES specialty (id) ON DELETE CASCADE;
18 ALTER TABLE specialty_center ADD CONSTRAINT FK_2DBBE9AC87F4FB17 FOREIGN KEY (doctor_id) REFERENCES doctor (id) ON DELETE CASCADE;
19 ALTER TABLE specialty_center ADD CONSTRAINT FK_728BF1E29A353316 FOREIGN KEY (specialty_id) REFERENCES specialty (id) ON DELETE CASCADE;
```

Dossier 2 :

Mission 1 : Implémentez un service de base de données (ex : mariadb, postgresql, mongodb, etc)

Pour travailler au développement de mon application, j'ai implémenté une base de donnée MySQL, les étapes de création grâce à Symfony et Doctrine sont grandement simplifiées, il suffit dans un premier temps de bien renseigner ses identifiants de connexion dans le fichier .env :

```
# DATABASE_URL="sqlite://%kernel.project_dir%/var/data.db"
DATABASE_URL="mysql://root@127.0.0.1:3306/north_health?serverVersion=8.0.32&charset=utf8mb4"
# DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/app?serverVersion=10.11.2-MariaDB&charset=utf8mb4"
# DATABASE_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=15&charset=utf8"
```

Nous devons ensuite demander à Doctrine de créer une base de donnée en exécutant :

```
php bin/console doctrine:database:create
```

Mission 2 : Implémentez un service d'hébergement d'application en lien avec les technologies que vous souhaitez implémenter (ex : tomcat, apache2, nodejs, etc.)

Le framework symfony propose un serveur incorporé qui s'exécute à l'aide de la commande :

```
symfony server:start
```

```
968.log)
Following PHP-CGI log file (C:\Users\Gérald\.symfony5\log\0dcf0ac30448dde3480f86046e70
36be244ee968f79ca75f9e90b4126a5955a335ea6a41ec5e854698.log)

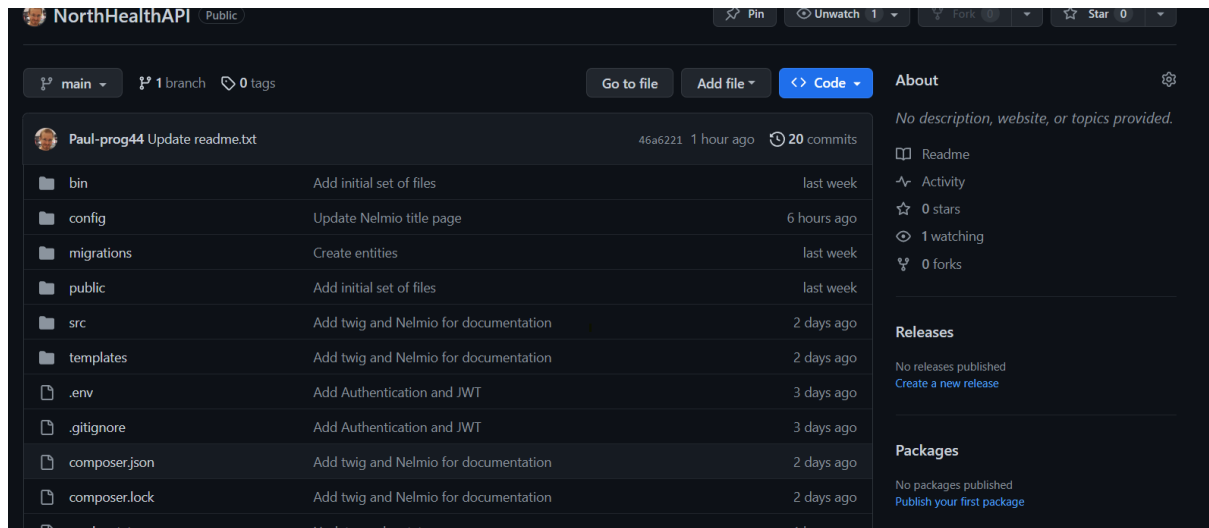
[WARNING] The local web server is optimized for local development and MUST never be u
sed in a production setup.

[OK] Web server listening
The Web server is using PHP CGI 8.2.4
https://127.0.0.1:8000

[Web Server ] Sep 25 16:00:18 |DEBUG | PHP      Reloading PHP versions
[Web Server ] Sep 25 16:00:18 |DEBUG | PHP      Using PHP version 8.2.4 (from default v
ersion in $PATH)
[Web Server ] Sep 25 16:00:18 |INFO | PHP      listening path="C:\\xampp\\php\\php-cgi
.exe" php="8.2.4" port=53148
]
```

Mission 3 : Implémentez un service de gestion de versionning (ex : gitlab, gitea, github, etc.) et déployez un premier code source pour chacune de vos applications (mobile, web, lourde, api ainsi que le code sql de votre base)

Afin d'assurer le suivi des versions de l'application, j'ai choisi d'utiliser GitHub, l'intégralité du dossier est disponible à cette url : <https://github.com/Paul-prog44/NorthHealthAPI> .



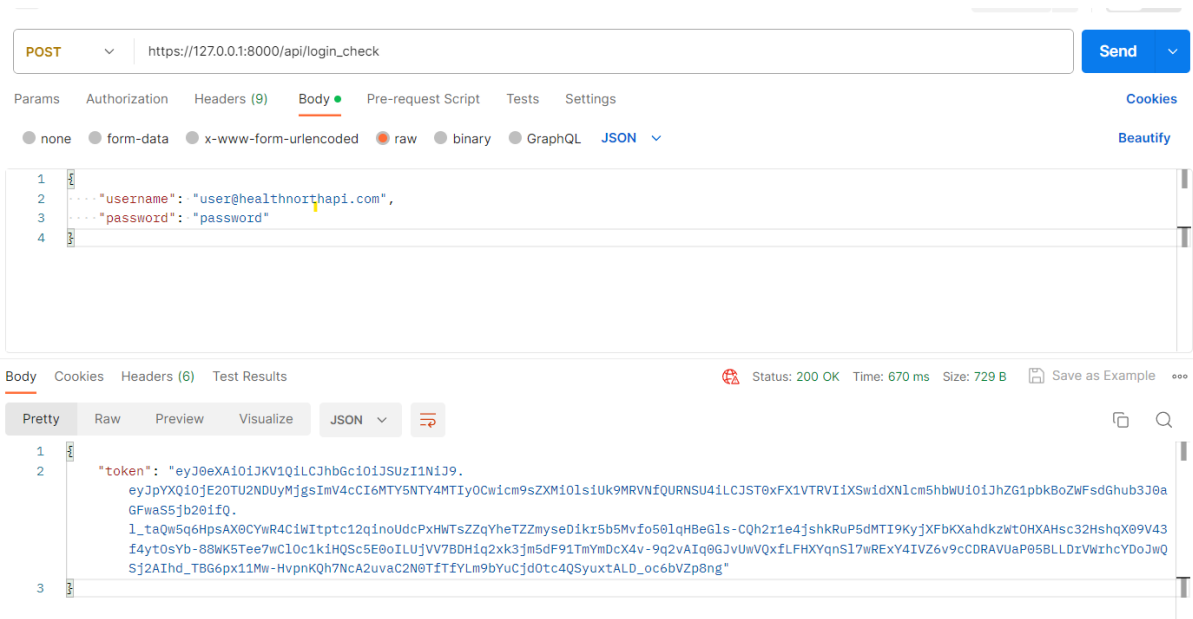
Mission 5 : Service d'authentification (ex : openldap, active directory, etc.)

L'accès à l'API requiert une authentification qui se fait grâce au composant [Jason Web Token \(JWT\)](#) .

Il s'agit d'un composant de sécurité qui repose sur l'utilisation d'un token d'identification à durée limitée et qui est généré grâce à des clés privées/publiques.

On demande à l'API de nous identifier en ce rendant à cette URL : https://127.0.0.1:8000/api/login_check

En envoyant un JSON contenant notre identifiant de connexion et notre mot de passe, nous obtenons un token permettant de requêter la base de donnée :

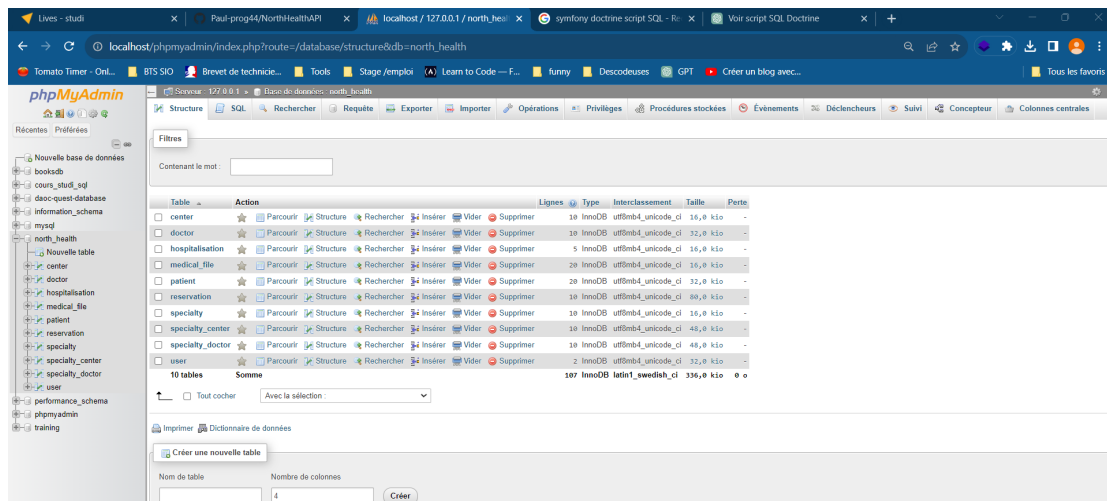


Dossier 3 :

Mission 1 : À l'aide du script du dossier 1 et du service de base de données implémenté en mission 2, déployez votre base de données et ajoutez des valeurs de tests dans vos tables.

Nous chargeons les tables dans la base de donnée précédemment créée en utilisant Doctrine :

```
php bin/console doctrine:schema:update --force
```



| | id | medical_file_id | gender | last_name | first_name | address | email_address | password | social_security |
|-------------------------|----|-----------------|--------|------------------|------------|-----------|----------------------------|------------|-----------------|
| Éditer Copier Supprimer | 1 | 1 | Mme | Nom de famille0 | Prénom0 | Adresse0 | randommail0@anonymous.com | password0 | 01234567890 |
| Éditer Copier Supprimer | 2 | 2 | M | Nom de famille1 | Prénom1 | Adresse1 | randommail1@anonymous.com | password1 | 01234567891 |
| Éditer Copier Supprimer | 3 | 3 | Mlle | Nom de famille2 | Prénom2 | Adresse2 | randommail2@anonymous.com | password2 | 01234567892 |
| Éditer Copier Supprimer | 4 | 4 | Mlle | Nom de famille3 | Prénom3 | Adresse3 | randommail3@anonymous.com | password3 | 01234567893 |
| Éditer Copier Supprimer | 5 | 5 | Mme | Nom de famille4 | Prénom4 | Adresse4 | randommail4@anonymous.com | password4 | 01234567894 |
| Éditer Copier Supprimer | 6 | 6 | M | Nom de famille5 | Prénom5 | Adresse5 | randommail5@anonymous.com | password5 | 01234567895 |
| Éditer Copier Supprimer | 7 | 7 | Mlle | Nom de famille6 | Prénom6 | Adresse6 | randommail6@anonymous.com | password6 | 01234567896 |
| Éditer Copier Supprimer | 8 | 8 | Mlle | Nom de famille7 | Prénom7 | Adresse7 | randommail7@anonymous.com | password7 | 01234567897 |
| Éditer Copier Supprimer | 9 | 9 | M | Nom de famille8 | Prénom8 | Adresse8 | randommail8@anonymous.com | password8 | 01234567898 |
| Éditer Copier Supprimer | 10 | 10 | Mlle | Nom de famille9 | Prénom9 | Adresse9 | randommail9@anonymous.com | password9 | 01234567899 |
| Éditer Copier Supprimer | 11 | 11 | M | Nom de famille10 | Prénom10 | Adresse10 | randommail10@anonymous.com | password10 | 012345678910 |
| Éditer Copier Supprimer | 12 | 12 | Mme | Nom de famille11 | Prénom11 | Adresse11 | randommail11@anonymous.com | password11 | 012345678911 |
| Éditer Copier Supprimer | 13 | 13 | M | Nom de famille12 | Prénom12 | Adresse12 | randommail12@anonymous.com | password12 | 012345678912 |
| Éditer Copier Supprimer | 14 | 14 | M | Nom de famille13 | Prénom13 | Adresse13 | randommail13@anonymous.com | password13 | 012345678913 |
| Éditer Copier Supprimer | 15 | 15 | M | Nom de famille14 | Prénom14 | Adresse14 | randommail14@anonymous.com | password14 | 012345678914 |
| Éditer Copier Supprimer | 16 | 16 | Mme | Nom de famille15 | Prénom15 | Adresse15 | randommail15@anonymous.com | password15 | 012345678915 |
| Éditer Copier Supprimer | 17 | 17 | Mme | Nom de famille16 | Prénom16 | Adresse16 | randommail16@anonymous.com | password16 | 012345678916 |
| Éditer Copier Supprimer | 18 | 18 | Mme | Nom de famille17 | Prénom17 | Adresse17 | randommail17@anonymous.com | password17 | 012345678917 |
| Éditer Copier Supprimer | 19 | 19 | M | Nom de famille18 | Prénom18 | Adresse18 | randommail18@anonymous.com | password18 | 012345678918 |
| Éditer Copier Supprimer | 20 | 20 | M | Nom de famille19 | Prénom19 | Adresse19 | randommail19@anonymous.com | password19 | 012345678919 |

Afin de d'ajouter des valeurs de tests, j'ai installé le [bundle fixture](#) qui permet de générer de fausses données et de tester la base de donnée et les relations, on les génèrent à l'aide la commande :

```
php bin/console doctrine:fixtures:load
```

| | id | medical_file_id | gender | last_name | first_name | address | email_address | password | social_security |
|-------------------------|----|-----------------|--------|------------------|------------|-----------|----------------------------|------------|-----------------|
| Éditer Copier Supprimer | 1 | 1 | Mme | Nom de famille0 | Prénom0 | Adresse0 | randommail0@anonymous.com | password0 | 01234567890 |
| Éditer Copier Supprimer | 2 | 2 | M | Nom de famille1 | Prénom1 | Adresse1 | randommail1@anonymous.com | password1 | 01234567891 |
| Éditer Copier Supprimer | 3 | 3 | Mlle | Nom de famille2 | Prénom2 | Adresse2 | randommail2@anonymous.com | password2 | 01234567892 |
| Éditer Copier Supprimer | 4 | 4 | Mlle | Nom de famille3 | Prénom3 | Adresse3 | randommail3@anonymous.com | password3 | 01234567893 |
| Éditer Copier Supprimer | 5 | 5 | Mme | Nom de famille4 | Prénom4 | Adresse4 | randommail4@anonymous.com | password4 | 01234567894 |
| Éditer Copier Supprimer | 6 | 6 | M | Nom de famille5 | Prénom5 | Adresse5 | randommail5@anonymous.com | password5 | 01234567895 |
| Éditer Copier Supprimer | 7 | 7 | Mlle | Nom de famille6 | Prénom6 | Adresse6 | randommail6@anonymous.com | password6 | 01234567896 |
| Éditer Copier Supprimer | 8 | 8 | Mlle | Nom de famille7 | Prénom7 | Adresse7 | randommail7@anonymous.com | password7 | 01234567897 |
| Éditer Copier Supprimer | 9 | 9 | M | Nom de famille8 | Prénom8 | Adresse8 | randommail8@anonymous.com | password8 | 01234567898 |
| Éditer Copier Supprimer | 10 | 10 | Mlle | Nom de famille9 | Prénom9 | Adresse9 | randommail9@anonymous.com | password9 | 01234567899 |
| Éditer Copier Supprimer | 11 | 11 | M | Nom de famille10 | Prénom10 | Adresse10 | randommail10@anonymous.com | password10 | 012345678910 |
| Éditer Copier Supprimer | 12 | 12 | Mme | Nom de famille11 | Prénom11 | Adresse11 | randommail11@anonymous.com | password11 | 012345678911 |
| Éditer Copier Supprimer | 13 | 13 | M | Nom de famille12 | Prénom12 | Adresse12 | randommail12@anonymous.com | password12 | 012345678912 |
| Éditer Copier Supprimer | 14 | 14 | M | Nom de famille13 | Prénom13 | Adresse13 | randommail13@anonymous.com | password13 | 012345678913 |
| Éditer Copier Supprimer | 15 | 15 | M | Nom de famille14 | Prénom14 | Adresse14 | randommail14@anonymous.com | password14 | 012345678914 |
| Éditer Copier Supprimer | 16 | 16 | Mme | Nom de famille15 | Prénom15 | Adresse15 | randommail15@anonymous.com | password15 | 012345678915 |
| Éditer Copier Supprimer | 17 | 17 | Mme | Nom de famille16 | Prénom16 | Adresse16 | randommail16@anonymous.com | password16 | 012345678916 |
| Éditer Copier Supprimer | 18 | 18 | Mme | Nom de famille17 | Prénom17 | Adresse17 | randommail17@anonymous.com | password17 | 012345678917 |
| Éditer Copier Supprimer | 19 | 19 | M | Nom de famille18 | Prénom18 | Adresse18 | randommail18@anonymous.com | password18 | 012345678918 |
| Éditer Copier Supprimer | 20 | 20 | M | Nom de famille19 | Prénom19 | Adresse19 | randommail19@anonymous.com | password19 | 012345678919 |

2) Implémentez un service d'API permettant d'obtenir des accès (CRUD) sur les bases principales de votre application.

A ce stade, si l'authentification à réussie, il est désormais possible de requêter l'API.

Le serveur est désormais à répondre aux requêtes :

POST https://127.0.0.1:8000 GET https://127.0.0.1:8000/api/doctors

https://127.0.0.1:8000/api/doctors

GET https://127.0.0.1:8000/api/doctors

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Type Bearer Token Token eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.e...

The authorization header will be automatically generated when you send the request. Learn

Body Cookies Headers (6) Test Results Status: 200 OK Time: 533 ms Size: 1.64 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "gender": "Mme",
4   "firstName": "Prénom",
5   "lastName": "NomDeFamille",
6   "specialties": [
7     {
8       "name": "dentaire"
9     }
10  ],
11  "center": {
12    "name": "NomDuCentre"
13  }
14 },
15 }
```

Extra :

Le [bundle Nelmio](#) nous permet de générer automatiquement une documentation de l'API affichant toutes les routes possible ainsi que toutes les actions disponibles pour chaque ressource :

North Health 1.0.0 GRASP

Api to access North Health company database

| default | |
|---------|----------------------------|
| GET | /api/centers |
| POST | /api/centers |
| GET | /api/centers/{id} |
| PUT | /api/centers/{id} |
| DELETE | /api/centers/{id} |
| GET | /api/doctors |
| POST | /api/doctors |
| GET | /api/doctors/{id} |
| PUT | /api/doctors/{id} |
| DELETE | /api/doctors/{id} |
| GET | /api/hospitalisations |
| POST | /api/hospitalisations |
| GET | /api/hospitalisations/{id} |
| PUT | /api/hospitalisations/{id} |

Afin de réaliser cette mission, je me suis aidé de :

Studi.com

Openclassrom

Stack overflow

Symfony.com

ChatGPT