Nom : Kane

Prénom : Fatou Aicha

Compte Rendu du Projet Cabinet Médical

1. La Partie Secrétaire

* secretary.component.ts

Dans ce fichier, j’ai fait l’importation de :

* Du service CabinetMedicalService :
* Des interfaces : CabinetInterface, InfirmierInterface, PatientInterface
* De NgForm qui est nécessaire pour la gestion des formulaires

Ce fichier est un composant donc il a le décorateur @Component. Dans ce décorateur j’ai mis le selector (qui permet d’utiliser ce composant dans un fichier html), un templateUrl (qui donne le lien vers le template html lié à ce composant).

Après le décorateur @Component, je fais la déclaration de la classe SecretaryComponent, qui implémente OnInit, et qui sera exportable.

J’ai créé le tableau infirmiers qui est l’instanciation de InfirmierInterface en faisant = infirmiers: InfirmierInterface [] = [];

Et patientsNonAffectés qui est l’instanciation de PatientInterface en faisant = PatientInterface [] = [];

Infirmiers et patientsNonAffectés contiendront les données des infirmiers et des patients issus du fichier XML cabinetInfirmier.xml contenu dans le serveur.

Comme paramètre du constructeur du composant Secretary, j’ai mis = (cs : CabinetMedicalService) qui est une dépendance au service cabinet-medical (cs est un paramètre de type CabinetMedicalService). Ceci va permettre de pouvoir utiliser les fonctions et données qui sont dans CabinetMedicalService.

Avec ngOnInit() on initialise le composant =

On a récupérer les données qui sont dans le fichier cabinetInfirmier.xml à travers une URL, on a mis ces données dans l’interface cabinet. On met ensuite toutes les données d’infirmiers qui sont dans cabinet, dans infirmiers en faisant : this.infirmiers = cabinet.infirmiers; et toute les données de patients non affectés, dans patientNonAffectés en faisant : this.patientsNonAffectés = cabinet.patientsNonAffectés;  
La fonction getInfirmiers() : permet de retourner tous les infirmiers.

La fonction getPatients(): permet de retourner tous les patients non affectées.

La fonction AjouterPatient: permet d’ajouter un patient dans le fichier XML, à travers un formulaire, il prend en paramètre NgForm qui a été renvoyé par la méthode ngsubmit lors de la validation du formulaire. J’ai créé une autre fonction AjouterPatient dans CabinetMedicalService, c’est pourquoi j’ai fait this.cs.AjouterPatient(f) pour envoyer les données du formulaire comme paramètre de la fonction AjouterPatient contenu dans CabinetMedicalService. Ensuite à travers un push j’ai ajouté le patient dans patientsNonAffectés : this.patientsNonAffectés.push(patient);

* secretary.component.html

Dans ce fichier j’ai le template html du composant secretary, il y a le code html et le css pour : la liste des patients non Affectés, la liste des infirmières, un formulaire d’ajout d’un patient, un formulaire d’affectation des infirmières à un patient.

Pour le formulaire d’ajout d’un Patient, j’ai crée une variable #newPatientForm ou je mets ngForm, ensuite à la soumission du formulaire, la fonction AjouterPatient sera appelé avec comme paramètre cette variable. Le Code : <form #newPatientForm="ngForm" (ngSubmit)="AjouterPatient(newPatientForm)">.

Par défaut, lorsqu’on ajoute un Patient, il est affecté à tous les infirmiers.

Pour afficher la liste des Patients, j’ai utilisé ngFor et j’ai récupérer tous les patients en utilisant la fonction getPatient que j’ai implémenté auparavant. Le code : \*ngFor="let patient of getPatients()"

1. La Partie Cabine-Médical

cabinet-medical.service.ts : ce fichier est un service qui permet de récupérer les données du serveur.

J’ai importé les interfaces de cabinet, nurse, patient, sexe, adress, j’ai aussi importé http qui va me permettre de faire des requêtes http pour récupérer les données du serveur via une url.

J’ai ensuite créé une fonction getdata qui permet de récupérer les données du serveur, il a comme paramètre l’url vers le fichier XML. Les données récupérées seront stockés dans une interface cabinet, qui sera renvoyé par la fonction. Les données infirmiers seront stockées dans des tableaux infirmiers et patientsNonAffectés que j’ai créé. J’ai mis le texte récupérer dans une constante text, puis j’ai construit un document DOM à partir des chaîne de caractères en utilisant la méthode parseFromString qui rend en paramètre le texte text à analyser et le type du document (ici: “text/xml”).

C’est dans un fichier que j’ai aussi implémenté la fonction AjoutPatient() qui permet d’ajouter un patient. Cette fonction prend on paramètre ngForm qui contient le contenu des données envoyés à travers le formulaire. J’ai créé une interface Patient appelé patient ou j’ai mis toute les données de patient issus du formulaire lors de sa soumission. Cette fonction retourne un patient.

1. App

* app.module.ts : Dans ce fichier j’ai importé le service CabinetMedicalService, le composant SecretaryComponent, NgModule et FormsModule qui permettent la gestion des formulaires
* app.component.html : Dans ce fichier on a ajouté <**app-secretary**></**app-secretary**> qui permet d’avoir au lancement le contenu html du composant Secretary.