

DOCUMENT D'ARCHITECTURE

Architecture du site :

Le site internet est construit avec la framework Angular et est donc basé sur l'architecture des projets Angular de base :

Dans le dossier app situé dans le dossier src on peut y trouver trois autres dossiers : component, model, et service.

Dans le dossier component on peut y retrouver d'autres dossiers contenant plusieurs modules de code pour des pages internet correspondantes au nom du dossier.

Chacun de ces dossiers contiennent donc des fichiers html et css ainsi que des fichiers Typescript afin de pouvoir interagir avec la page, mais aussi pour manipuler les données transmises par les autres modules du projet Angular.

Dans le dossier model, on y trouve tout un tas de fichiers Typescript contenant ce que l'on appelle des interfaces. Celles-ci fonctionnent un peu comme des structures en C ou des classes en Programmation Orienté objet, sauf qu'elles ne possèdent pas de méthodes mais juste des variables.

Ce sont donc des modèles dont le but est de créer des instances afin de manipuler des données plus facilement.

Leur utilisation aura plus de sens lorsque l'on aura terminé de parler des services.

Dans le dossier service se trouve des fichiers Typescript dont le but est d'instancier les modèles cités précédemment et de stocker des fonctions permettant d'accéder à ces instances.

A chaque modèle correspond un service, cela permet de s'y retrouver plus facilement dans le code.

Les services sont ensuite réutilisés par les composants afin de manipuler et afficher les instances créées.

Pour résumer :

Les composants manipulent des objets construits sur la base des model et fournis ainsi qu'instancié par les services.

Architecture de l'api :

L'API permet de manipuler des éléments d'une base de données no SQL. Pour cela on a décidé de faire une API-REST.

Le dossier contenant le code de l'api contient deux fichiers principaux : app.js et calendar.js

Calendar.js :

Ce fichier contient le code afin de discuter avec le google-calendar du compte ayant fourni un 'refresh-token' (ce qui permet de connecter le compte google à notre api).

Il possède donc une fonction qui construit un «event» grâce aux données passées par la requête.

La fonction regarde ensuite si l'agenda est libre sur l'horaire de l'«event» initialisé et ajoute l'«event» en question si c'est le cas. Ainsi la fonction retournera le message « évènement ajouté ».

Dans le cas contraire on essaye de chercher un nouvel horaire de libre avec une boucle while et en incrémentant l'horaire de passage d'une heure.

Une fois qu'un horaire est trouvé, la fonction renvoie un message contenant l'horaire à partir de laquelle le propriétaire du calendrier est disponible.

App.js

Ce fichier contient toutes les méthodes d'accès et de manipulation des données de la base de données (donc les méthodes get, post, put, delete pour recevoir, créer, modifier et/ou supprimer).

Elle possède en plus une méthode post qui permet d'appeler la fonction créée dans Calendar.js

Le dossier de l'api contient aussi un sous-dossier « site » contenant le code générant une page permettant de manipuler les données de la base de données.

Voici un peu plus d'explication :

Notre client souhaitait que les utilisateurs du site puissent poser des questions afin d'imiter le principe d'une FAQ.

Ainsi dans la page contact les utilisateurs posent une question qui est envoyée à la base de données pour ensuite être affichée dans la page principale du site si l'un des membres du cabinet a répondu à la question.

Le fait de devoir répondre à la question avant que celle-ci soit affichée sur le site, permet à notre client et ses coéquipiers de jouer un rôle de modérateur. Cela nous dispense de devoir mettre en place une modération automatique sur un certain vocabulaire, ce qui peut se montrer très compliqué.

Pour que les kinésithérapeutes puissent répondre aux questions il faut une interface, et c'est là que le code contenu dans le dossier site intervient. La page générée par ce code permet de visualiser, répondre et/ou supprimer les questions stockées dans la base de données.

Si cette page ne se situe pas sur le site principal, c'est avant tout par manque de temps.

En effet, il aura fallu mettre en place une partie admin sur le site, et il aurait fallu que l'on puisse accéder à cette partie à l'aide d'identifiants afin d'empêcher n'importe qui d'accéder à la base de données.

Mais même avec cette solution nous ne pouvons pas exclure un problème de cybersécurité, étant donné que nous n'avons pas reçu de formations pour protéger correctement un site ou une base de données contre certaines attaques.

Mise en ligne :

L'api est hébergée grâce à cyclic : <https://www.cyclic.sh>

Le site est hébergé grâce à Netlify : <https://app.netlify.com>

La base de données est une base de données MongoDB hébergé sur MongoDB Atlas

Source pour savoir utiliser google calendar :

<https://developers.google.com/calendar/api/guides/overview?hl=fr>

<https://www.youtube.com/watch?v=zrLf4KMs71E>