

Test technique - suite

Contexte

Maintenant que la logique de concaténation d'horaires est implémentée, nos utilisateurs nous demandent de leur proposer une interface pour être autonomes dans la modification de leurs horaires !

Objectif

En utilisant le module de traitement des horaires développé précédemment, construire une interface permettant aux utilisateurs de gérer les horaires de leur établissement à partir d'un fichier CSV qu'ils fournissent de manière autonome :

- Lorsque l'utilisateur se connecte sur l'outil, on lui propose deux sections :
 - Une section "Gestion des horaires existants"
 - Une section "Ajout d'horaires"

Timetables manager

1

2

Selected daterange

Opening datetime	Closing datetime
<input checked="" type="checkbox"/> 2020-10-01 10:00:00	2020-10-01 14:00:00
<input checked="" type="checkbox"/> 2020-10-01 18:00:00	2020-10-02 04:00:00
<input type="checkbox"/> 2020-10-02 12:00:00	2020-10-02 18:00:00
<input type="checkbox"/> 2020-10-03 10:00:00	2020-10-03 18:00:00
<input type="checkbox"/>

1

2

3

Upload your file Here

Upload

Opening datetime	Closing datetime
2020-10-01 10:00:00	2020-10-01 14:00:00
2020-10-01 18:00:00	2020-10-02 04:00:00
2020-10-02 12:00:00	2020-10-02 18:00:00
2020-10-03 10:00:00	2020-10-03 18:00:00
...	...

Do you want to save those timetables ? It will overwrite existing data

Yes

Cancel

Gestion des horaires existants

Cette section permet à l'utilisateur de visualiser les horaires qu'il a précédemment enregistré.

1. Un sélecteur de date invite l'utilisateur à préciser les dates pour lesquelles il souhaite visualiser les horaires.
2. Une fois la plage de dates sélectionnée, on affiche à l'utilisateur les horaires actuellement enregistrés sous forme de tableau. Il a la possibilité de sélectionner un ou plusieurs créneaux d'ouverture et de les supprimer en masse.

Section Ajout d'horaires

Cette section permet à l'utilisateur d'envoyer un fichier csv (formaté selon les spécifications précédentes) et de préciser si le contenu du fichier doit écraser les horaires précédemment enregistrés ou non :

1. L'utilisateur télécharge son fichier CSV.
2. Une fois le fichier téléchargé et traité (concaténation des horaires contiguës, suppression des jours fermés, ...), on affiche le résultat sous forme de tableau
3. L'utilisateur confirme qu'il souhaite sauvegarder les horaires renseignés.
 - a. S'il confirme, on écrase tous les horaires précédemment enregistrés. Par exemple, si le fichier comporte des horaires entre le 01/10 et le 31/10, on supprime tous les horaires dans cette plage de dates précédemment enregistrés et on les remplace par les nouveaux horaires renseignés.
 - b. S'il annule, on réinitialise le formulaire sans altérer les horaires précédemment enregistrés

Contraintes techniques

Tu dois utiliser Angular 9+ pour le front et Node.js/Typescript en back. Toutes les libraires sont acceptées à condition qu'elles soient justifiables. Les données doivent être persistées sur une base de données de ton choix (mysql conseillé).

Une procédure de déploiement (un readme par exemple) doit être fournie pour permettre de lancer et de tester le projet.