



## Documentation code IHM Visualisation

Document : Documentation Code

Auteur : LOMELET Guilhem

Date : 12/05/2023

### SOMMAIRE

<b>Introduction</b>	<b>2</b>
<b>1 - Comprendre le code</b>	<b>3</b>
LES COMPOSANTS	3
API	3
BOUTON AFFICHAGE	4
DÉPLACEMENT DANS LES SEMAINES	4
CSS	4
JAVASCRIPT	4
<b>2 - Ce qu'il reste à faire</b>	<b>5</b>
LES FILTRES / TRIS	5
DRAG AND DROP	5

# Introduction

Ce document a deux objectifs majeurs :

- Expliquer ce qui a été fait et comment cela fonctionne;
- Expliquer ce qu'il manque, à quoi cela doit rassembler et comment le faire.

Ce document a été rédigé en fin de Projet Transversal par Guilhem LOMELET pour permettre la compréhension de ce qui a été fait et la suite du développement. Ils se concentrent sur la partie visualisation des cours et ne traite pas de la partie Ajout de cours.

Le code se trouve dans le git suivant : [https://gitlab.univ-nantes.fr/E209929J/flop-edt\\_ptrans/](https://gitlab.univ-nantes.fr/E209929J/flop-edt_ptrans/)

Comme dit dans le readMe pour lancer le code et le tester :

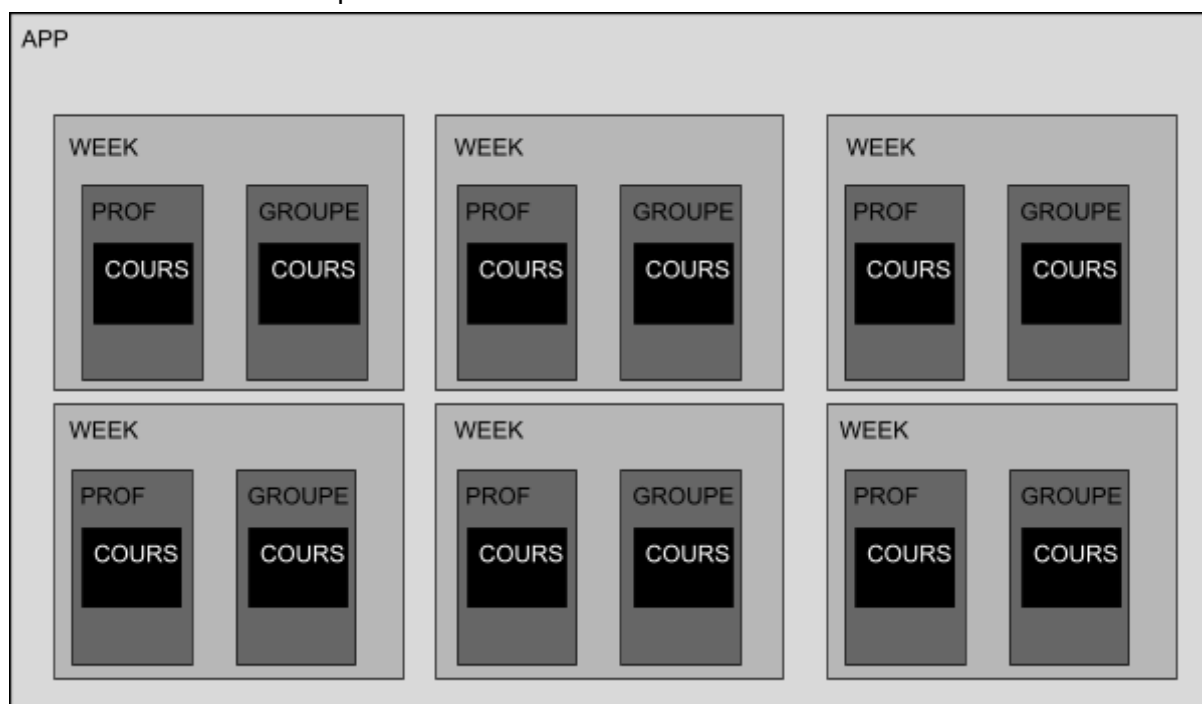
- Aller dans le repertoire VueJS
- Installer les dépendances avec la commande : npm install
- Lancer le serveur de développement avec la commande : npm run dev

# 1 - Comprendre le code

Le code a été développé en Vue.JS ([doc](#)). Il est donc organisé en composant qui s'appelle entre eux et sont imbriqués les uns dans les autres. Chaque composant est organisé en trois parties : HTML; JavaScript; CSS. Chaque composant est représenté par un fichier ".vue".

## LES COMPOSANTS

Voici un schéma des composants et de leurs imbrications :



Chaque composant est appelé par son composants mère (i.e celui dans lequel il est directement imbriqué).

Le composant APP crée donc 6 composants WEEK. Les composants WEEK créent autant de composants PROF et GROUPE qu'il en existe en base. Les composants PROF et GROUPE créent autant de composants COURS qu'il en existe en base lié à ce dit PROF ou GROUPE.

## API

L'appel à l'API se fait lors de la création des composants WEEK. Une requête est faite chaque semaine. Cette requête est la suivante : `fetch/scheduledcourses` avec les paramètres `year`, `week` et `dept`. Cette requête renvoie un objet JSON qui est traité pour en tirer 4 listes :

- 2 listes qui contiennent l'entièreté des Groupes pour l'une et des profs pour l'autre (Utile à la génération des composants PROF et GROUPE).
- 1 tableau `CoursByGroupe` qui stocke l'ensemble des cours pour chaque groupe et pour chaque semaine. Permet d'accéder à l'ensemble des cours d'un groupe à partir

de son nom (CoursByGroupe[[NuméroDeSemaine]][NomDuGroupe]) pour générer les composants COURS.

- 1 tableau CoursByProf qui stocke l'ensemble des cours pour chaque prof et pour chaque semaine. Permet d'accéder à l'ensemble des cours d'un prof à partir de son nom (CoursByProf[[NuméroDeSemaine]][NomDuProf]) pour générer les composants COURS.

Les fonctions qui font appel à l'api ainsi que l'URL de l'api sont déclarées dans le fichier main.js. Elles sont ensuite utilisées dans les composants WEEK, PROF et GROUPE.

**A noter :** Il faudra ajouter des fonctions pour récupérer l'année actuelle ainsi que le dept qui sont pour l'instant passer en dur dans le composant WEEK ligne 49.

## BOUTON AFFICHAGE

Le bouton affichage en haut à droite permet de choisir le type d'affichage que l'on souhaite avoir. Soit on décide d'afficher tous les cours pour chaque semaine par prof, soit par groupe.

Il faut savoir que dans tous les cas l'ensemble des composants PROF et GROUPE sont créés peu importe l'affichage. L'affichage est géré par une des propriétés de Vue.JS (v-if) qui permet d'afficher ou non les composants suivant la valeur d'un booléen. On vient donc changer la valeur des booléens lorsque que l'on clique sur nos boutons en haut à gauche.

## DÉPLACEMENT DANS LES SEMAINES

Le déplacement dans les semaines est géré par deux boutons dans le composant principal APP. Ces boutons font appel à la fonction getListeWeek(delay) qui permet de modifier la liste des semaines à afficher. On peut facilement changer le pas de décalage en changeant le paramètre delay.

## CSS

Le css des composants est déclaré dans la partie style de ces derniers. L'ensemble du CSS est réalisé sans Framework extérieur.

## JAVASCRIPT

La majorité des fonctions javascript sont déclarées dans les composants javascript où elles sont utilisées. Exception est faite pour l'API avec des fonctions déclarés dans le fichier main.js

## 2 - Ce qu'il reste à faire

### LES FILTRES / TRIS

Une partie importante de ce qu'il reste à faire est l'ajout du fonctionnement des boutons de filtres en haut à droite. Voici une idée du fonctionnement que cela devrait avoir : [maquette figma](#). L'objectif est ici de faire apparaître au premier clic sur l'un des 3 boutons l'ensemble des groupes/module/prof présent en base puis lorsque l'on en sélectionne un, ou plusieurs ne garder que cela à l'affichage.

Voici les différentes étapes pour réaliser cela :

- Utiliser les listes de prof et groupe déjà créés avec les fonctions getListeProf, getListeGroupe et créer la fonction getListeModule
- Créer de nouveau composant Vue que l'on fait apparaître au clic (grâce à la propriété v-if) sur les boutons avec la liste correspondantes
- Ajouter des v-if sur les composants COURS pour pouvoir les faire apparaître/disparaître lors de la sélection des filtres.

### DRAG AND DROP

Une seconde fonctionnalité importante est le drag and drop pour déplacer les cours de semaine et rendre cette visualisation flexible. Le code a normalement été pensé pour rendre cet ajout facile.

Voici les étapes pour réaliser cela :

- Pour cela une bibliothèque existe pour faciliter le développement de cette feature "vuedraggable". Il faudra donc installer cette bibliothèque (npm install vuedraggable).
- Il faut ensuite importer vuedraggable dans notre composant COURS (import draggable from 'vuedraggable';)
- Changer dans le composant COURS le code HTML en y ajoutant des bornes <draggable> aux extrémités
- Il faudra ensuite ajouter les méthodes suivantes à la partie script du composant COURS :

```
onDrop(event) {
  // Logique à exécuter lorsqu'un élément est déposé (MODIFICATION DES
  DONNEES DANS LE BASE / UTILISATION DE L'API)
  console.log('Drop:', event);
},
onDrag(event) {
  // Logique à exécuter lorsqu'un élément est en cours de glissement
  console.log('Drag:', event);
}
```

- On peut également modifier le CSS pour changer le style lors du drag avec la classe “.is-dragging” à utiliser comme un hover.