

# TP 3

## EFREI 3A WEB

Louis Cherel - 08/2020

### Préambule

Merci de lire **toutes les consignes**, elles sont écrites pour être lues.

### Les TP sont tous ramassés

Tous les TP sont ramassés et doivent être rendus au plus tard à 23h55 le soir de la séance sur l'espace Moodle prévu à cet effet.

**Chaque jour de retard vous retire 1 point** sur votre note de TP, avec une limite à 3 jours de retard, ce qui, passé ce délai, vous vaudra une note de 0 sur votre TP.

Seul un des TPs sera noté, mais ne pas rendre les autres vous vaudra quand même 0.

Les TP et le projet sont à réaliser en binôme ou trinômes, et doivent être enregistrés auprès de votre enseignant-e avant la fin de la première séance. Sans votre enregistrement d'équipe, vous ne pourrez pas rendre votre travail.

### Les ressources conseillées

Vos armes seront donc le site officiel de Vue.js, Mozilla Developer Network (MDN), Stack Overflow, et les recherches en anglais. **Évitez au maximum** les forums (pas les cours) des sites comme OpenClassrooms ou Comment Ça Marche, qui proposent des solutions souvent peu fiables. W3Schools peut être une bonne ressource, mais les bonnes pratiques et les standards ne sont pas leur fort. Leur préférer le MDN lorsque possible.

Quantité de blogs sont très bons, notamment alsacreations (en français) ou css-tricks (en anglais).



Cette œuvre est mise à disposition par Louis Cherel selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#)

# Installations préalables

Pour réaliser ce TP, vous aurez besoin d'installer sur votre ordinateur:

- [Node.js](#): ce sera notre serveur Web

## Exercices

### Exercice 1 (⌚ 20 min pratique)

- 1) Téléchargez l'archive [suivante](#) qui contient le squelette du TP. Cette archive contient les éléments nécessaires au bon fonctionnement du reste du TP. Assurez-vous de bien avoir installé Node.js avant toute chose.
- 2) Extrayez l'archive dans un dossier quelconque de votre PC.
- 3) Ouvrez PowerShell (windows), ou un terminal sur linux/mac, naviguez jusqu'au nouveau dossier créé, et installez les dépendances du projet en tapant la commande suivante depuis le dossier créé: `npm install`
- 4) Si tout s'est bien passé, un nouveau dossier `node_modules` devrait être apparu dans le dossier du projet. Ce dossier contient tous les modules externes à notre propre code.
- 5) Pour lancer notre serveur, entrez la commande suivante: `npm start`
- 6) Ouvrez votre navigateur web et entrez l'URL suivante:  
<http://localhost:3000>
- 7) Si tout s'est bien passé, vous devriez constater qu'il y a indiqué "Hello, World!", ainsi que "Hi everyone it works" sur la page
- 8) Si l'une de ces étapes n'a pas fonctionné, demandez à votre responsable de TP.

### Exercice 2 (⌚ 30 min lecture + ⌚ 30 min pratique)

Lecture préliminaire: [Vue.js: Composants](#)

**Note:** Vous avez plusieurs façons d'approcher la question des composants. Vue permet deux manières différentes, soit via la méthode présentée dans le guide, soit via les Single File Components. Vous avez un exemple de SFC (fichier .vue) dans client/components/HelloOtherWorld.vue



Cette œuvre est mise à disposition par Louis Cherel selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#)

Vous avez le choix de l'une ou l'autre méthode, la méthode présentée dans le guide est la plus simple, mais les SFC permettent d'avoir la coloration syntaxique pour la partie template du composant (grâce au plugin **Vetur** de VSCode).

Le dossier **client/** contient:

- index.html: contient le template principal de l'application Vue
- vue-application.js: contient la racine de l'application, à savoir l'instance de Vue et un composant
- **components/**
  - HelloOtherComponent.vue: un composant sous la forme de SFC

- 1) Nous voulons afficher le nombre de jours qu'il y a dans le mois actuel.  
Dans index.html, affichez dynamiquement le nombre de jours du mois courant grâce à une méthode créée dans vue-application.js
- 2) Créez une nouvelle méthode daysOfMonth qui prend en paramètre le mois et l'année actuels, et qui retourne un tableau rempli des nombres représentant les jours du mois (par ex 1, 2, 3, ..., 31)
- 3) Dans le template, utilisez v-for afin d'afficher la liste des numéros de jours du mois courant dans un <ul>
- 4) Pour chaque jour du mois courant, ajoutez le nom du jour (lundi, mardi, etc. l'anglais est également accepté), par ex:

1: Mardi

2: Mercredi

...

30: Jeudi

- 5) Ajoutez une liste déroulante affichant la liste des mois d'une année, afin que l'utilisateur puisse choisir le mois dont il souhaite afficher les jours. Le choix dans cette liste doit faire évoluer la liste des jours affichée
- 6) Ajoutez également une liste déroulante pour l'année
- 7) Déplacez tout votre code lié à l'affichage des jours du mois dans un composant days-of-month. Votre composant doit contenir le HTML et les méthodes Javascript, ainsi que des props permettant de faire changer les valeurs du mois et de l'année
- 8) Appelez votre composant depuis votre application, et vérifiez qu'il fonctionne correctement



Cette œuvre est mise à disposition par Louis Cherel selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#)

## Exercice 3 (⌚ 20 min pratique)

- 1) Réutilisez votre composant days-of-month afin d'afficher les jours du mois de tous les mois de l'année
- 2) En utilisant du CSS et des Balises HTML, faites en sorte qu'on puisse visualiser tous les jours de l'année d'un seul coup, en les réorganisant spatialement
- 3) En haut de chaque liste de jours de mois, affichez le nom du mois en question
- 4) En haut de votre liste de jours, affichez l'année
- 5) Encapsulez l'intégralité des questions 1,2,3,4 dans un composant appelé days-of-year qui aura une prop *year*
- 6) Appelez votre composant days-of-year depuis votre application pour qu'il fonctionne
- 7) Réutilisez votre composant afin d'afficher la liste des jours des années 2010 à 2020. Prennez le temps d'avoir un affichage agréable avec du CSS

## Exercice 4 (⌚ 20 min lecture + ⌚ 30 min pratique)

Lecture préalable nécessaire: [Using fetch](#)

- 1) L'API publique date.nager.at permet de récupérer les jours fériés d'une année en France. L'URL est:  
<http://vps-4401e6e0.vps.ovh.net/api/v2/PublicHolidays/2020/FR>
- 2) Utilisez fetch pour récupérer les valeurs d'une année donnée dans le composant days-of-year, en fonction de la prop *year*. Nappelez l'API qu'une seule fois par année, il vous vaudra trouver une façon de transmettre l'information au composant days-of-month
- 3) Cette API retourne un tableau d'objets Javascript contenant les propriétés suivantes:

```
{  
  "date": "2020-12-25",  
  "localName": "Noël",  
  "name": "Christmas Day",  
  "countryCode": "FR",  
  "fixed": true,  
  "global": true,  
}
```



Cette œuvre est mise à disposition par Louis Cherel selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#)

```

    "counties": null,
    "launchYear": null,
    "type": "Public"
}

```

- 4) Les propriétés qui nous intéressent sont *date* et *localName*. Pour tous les jours d'une année qui sont des jours fériés, mettez-les en gras et affichez leur localName à côté du nom du jour (mardi, etc.), mais vous êtes libres de la présentation ex:

1: Mardi: Jour de l'an

2: Mercredi

...

30: Jeudi

## **Exercice 5 (⌚ 30 min lecture + ⌚ 30 min pratique)**

Lectures préalables nécessaires:

- [Vue Router: Pour commencer](#)

**Nous souhaitons avoir plusieurs “pages” dans notre application Vue.** En effet, nous aimerais avoir une page #/hello-world et une page #/days/:year (où :year doit être remplacé par une année)

- 1) En suivant le guide de Vue, ajoutez le router dans vue-application.js ainsi que dans index.html, afin que le composant hello-world soit affiché uniquement lorsque l'URL est #/hello-world
- 2) Ajoutez un bouton dans votre application qui permette d'aller au à #/hello-world, en utilisant <router-link> pour cela
- 3) Créez un composant show-days-of-year, dont le seul but est d'encapsuler le composant days-of-year avec a prop year="2020"
- 4) Ajoutez une route #/days dans votre router qui affiche votre composant show-days-of-year
- 5) Lisez la documentation [Concordance dynamique de route](#)
- 6) Modifiez votre route #/days en #/days/:year, afin que le composant days-of-year soit appelé en fonction de l'année indiquée dans l'URL



Cette œuvre est mise à disposition par Louis Cherel selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#)