



Beste de savoir

Vue.js, TypeScript et données

26 décembre 2018

Table des matières

1. Introduction	1
---------------------------	---

1. Introduction

Je suis toujours sur mon projet qui utilise le combo Vue.js + TypeScript en front. Aujourd'hui, je voulais remplir une *data table* avec des données reçues par appel à une API. Côté réception des données, aucun soucis, ils sont bien là. Par contre, pour l'affichage, ça été plus compliqué.

Voici à quoi ressemble la partie `script` de mon fichier `.vue` en cours d'édition.

```
1 export default {
2   mounted()
3   {
4     this.getAllPosts();
5   },
6   data()
7   {
8     return {
9       headers: [
10        { text: 'ID', value: 'id' },
11        { text: 'Title', value: 'title' },
12      ],
13
14      posts: [] as Post[]
15    }
16  },
17  methods:
18  {
19    getAllPosts(): void
20    {
21      api.entities.posts.getAll().then((response)
22        => { this.posts = response as Post[]
23        });
24    }
25  };
26 }
```

L'affichage n'est pas important, pas plus que la moitié du code. Ce qui compte, c'est la méthode `getAllPosts`. Sauf que celle-ci ne fonctionnait pas. Impossible de savoir pourquoi, mais pour le

1. Introduction

compilateur, **this.posts** n'existe pas et je n'ai accès qu'à `getAllPosts` quand je fais `this.` + `Ctrl`+`Space`. Bizarre quand même!

De plus, impossible de comprendre pourquoi tous les codes sur Internet, si identiques, sont fonctionnels. Et mon frère me pose alors une question qui m'éclaire : « Mais ton code, c'est du JavaScript ou du TypeScript ? » Parce que oui, tous les codes que j'ai vu sont en JavaScript 100% standard. Alors il me dirige vers la [documentation](#) . Je note un léger changement par rapport à mon code.

```
1 import Vue, { VNode } from 'vue'
2
3 const Component = Vue.extend({
4   /* ... */
5 });
```

En fait, avec TypeScript, il y a deux solutions. Soit utiliser `Vue.extend`, soit à l'aide de décorateurs, pour créer des classes et non plus de simples objets.

```
1 import { Vue, Component, Prop } from 'vue-property-decorator';
2
3 @Component
4 export default class Posts extends Vue {
5   /* ... */
6 }
```

Je vous invite à lire [cet article](#) qui donne quelques précisions à mes propos. En tout cas, Vue.js a un avantage, c'est d'avoir **une documentation claire, propre et assez exhaustive**.