Dear Diary!

So I finished my REST service which will deliver my search results. But what good is that without an appealing frontend.

After I was nearly blown away by the overwhelming good critics about **Vue.js** at last year's **enterjs** I took the Vue-way immediately.

I think, Vue is *very* easy compared to other frameworks out there and also very well structured.

Also, it has proper Typescript support and good documentation.

So I **started a Vue project** using **Vue CLI** (which has an amazing new UI btw).

A simple test of the REST calls worked fine, but I wanted something more appealing.

Luckily, **Vuetify** has some nice Material Design components ready for Vue and also a nice autocomplete component.

This method watches the query input field and uses the REST backend to fill the results. I used a delayed promise there to allow the user to input the complete search text before I use the text to search the backend.

The autocomplete than renders **a so-called „slot"** with the item results.

Selecting a result would trigger a location change to the actual terraform documentation page.

The hard thing here was to get my head around how Vue plugins work, how they work in *Typescript* and how Vuetify components and their specialities (like slots) work. But the good examples and documentation for Vuetify and Vue helped a great deal.

Yours
Dennis

*This post is one of five posts from the **tflookup developer diary series***

*Cover Image: **„diary writing" by Fredrik Rubensson***

*Originally published to **dev.to***