SPAs with Vue.js

What is a single-page application (SPA) ?

It’s a web app that fits on a single web page, providing fluid & fast UX by loading all necessary code with a single page load.

Single-page applications allow you to simulate the work of desktop apps. The architecture is arranged in such a way that when you go to a new page, only a portion of the content is updated. Thus, there is no need to re-download the same elements.

Ex: GMail, Twitter, Facebook, Trello

SPA advantages

The 3 Rs:

- **Reach**. Easily develop for multiple devices with the finished code.
- **Rich User Experience**. Pages move fluently with no overhead latency because of client side navigation. Important state persists on the client (in cache in the browser, in the local storage or in cookies).
- **Reduced round tripping**. Less refreshes & postbacks of the entire page, only the required parts. Loads (full or most) needed data and views on the initial load then loads the rest on demand, asynchronously. Progressively download features as required, when the user navigates to other parts of the web app.

SPA disadvantages

- **JavaScript dependence**. SPA are JS heavy and some users disable JS in their browsers.
- **Poor SEO optimization**. SPA downloads information on request from the client part. Search engines can hardly simulate this behavior. Because most of the pages are simply not available for scanning by search bots.
- **Security**. JS has a low level of security, but if you use modern frameworks, like Vue.js **:D**, they can make your web application secure.

Model–view–viewmodel (MVVM)

MVVM was invented in 2005 by Microsoft architects Ken Cooper and Ted Peters specifically to simplify event-driven programming of user interfaces.

It facilitates a separation of development of the graphical user interface from development of the business logic or back-end logic (the data model).

Components of MVVM pattern are **model**, **view**, **view model**, **binder**.

What is Vue.js?

Vue.js is a JavaScript framework for building web interfaces using the model-view-viewmodel (MVVM) architecture pattern.

- Lightweight and incrementally adoptable.
- Easy to learn, pick up and integrate with other libraries or existing projects.
- Although it has a simpler syntax, it can power sophisticated SPAs in combination with modern tooling and supporting libraries.

Reactive vs imperative programming

Reactive programming is a **declarative programming** paradigm concerned with data streams and the propagation of change. For example:

- In an imperative programming setting, **a=b+c** would mean that **a** is being assigned the result of **b+c** in the instant the expression is evaluated, and later, the values of **b** and **c** can be changed with no effect on the value of **a.** (Ex: JQuery)
- In reactive programming, the value of **a** is automatically updated whenever the values of **b** or **c** change, without the program having to re-execute the statement **a=b+c.** (Ex: Vue.js, React, AngularJS, Angular, Ember, Knockout, Polymer, Riot)

Software components

An individual software component is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data).

Main characteristics of a component:

- reusable
- loosely coupled
- independent