



# Frontend development Class 06, Series 02

## Frontend

**HahuJobs**

## CLASS 06

### 01 Ways of Using Vue

**HahuJobs**

# Ways of Using Vue



Depending on your use case, Vue can be used in different ways to strike the optimal balance between stack complexity, developer experience and end performance.



# Ways of Using Vue

## Standalone Script

Vue can be used as a standalone script file - no build step required! If you have a backend framework already rendering most of the HTML, or your frontend logic isn't complex enough to justify a build step, this is the easiest way to integrate Vue into your stack. You can think of Vue as a more declarative replacement of jQuery in such cases.

Vue also provides an alternative distribution called [petite-vue](#) that is specifically optimized for progressively enhancing existing HTML. It has a smaller feature set, but is extremely lightweight and uses an implementation that is more efficient in no-build-step scenarios.



# Ways of Using Vue

## Embedded Web Components

You can use Vue to build standard **Web Components** that can be embedded in any HTML page, regardless of how they are rendered. This option allows you to leverage Vue in a completely consumer-agnostic fashion: the resulting web components can be embedded in legacy applications, static HTML, or even applications built with other frameworks.



# Ways of Using Vue

## Single-Page Application (SPA)

Some applications require rich interactivity and non-trivial stateful logic on the frontend. The best way to build such applications is to use an architecture where Vue not only controls the entire page, but also handles data updates and navigation without having to reload the page. This type of application is typically referred to as a Single-Page Application (SPA).



# Ways of Using Vue

## Fullstack / SSR

Pure client-side SPAs are problematic when the app is sensitive to SEO and time-to-content. This is because the browser will receive a largely empty HTML page, and has to wait until the JavaScript is loaded before rendering anything.

Vue provides first-class APIs to "render" a Vue app into HTML strings on the server. This allows the server to send back already-rendered HTML, allowing end users to see the content immediately while the JavaScript is being downloaded. Vue will then "hydrate" the application on the client side to make it interactive. This is called **Server-Side Rendering (SSR)** and it greatly improves Core Web Vital metrics such as Largest Contentful Paint (LCP).



# Ways of Using Vue

## JAMStack / SSG

Server-side rendering can be done ahead of time if the required data is static. This means we can pre-render an entire application into HTML and serve them as static files. This improves site performance and makes deployment a lot simpler since we no longer need to dynamically render pages on each request.

Vue can still hydrate such applications to provide rich interactivity on the client. This technique is commonly referred to as Static-Site Generation (SSG), also known as **JAMStack**.





# Ways of Using Vue

## Beyond the Web

Although Vue is primarily designed for building web applications, it is by no means limited to just the browser. You can:

- Build desktop apps with [Electron](#) or [Tauri](#)
- Build mobile apps with [Ionic Vue](#)
- Build desktop and mobile apps from the same codebase with [Quasar](#)
- Use Vue's [Custom Renderer API](#) to build custom renderers targeting [WebGL](#) or even [the terminal!](#)



# Ways of Using Vue

## Beyond the Web

Although Vue is primarily designed for building web applications, it is by no means limited to just the browser. You can:

- Build desktop apps with [Electron](#) or [Tauri](#)
- Build mobile apps with [Ionic Vue](#)
- Build desktop and mobile apps from the same codebase with [Quasar](#)
- Use Vue's [Custom Renderer API](#) to build custom renderers targeting [WebGL](#) or even [the terminal!](#)



# Class Exercise

## Build countries list

- show list of countries
  - displayed information should include
    - name
    - flag
    - continent
    - country code
    - phone code
    - capital
    - currency
    - languages spoken
    - it should include list of states
- design with figma
- GraphQL Server:

<https://countries.trevorblades.com/>

# Thank you!

## HahuJobs

ለህገር ልጅ በህገር ልጅ !

Michael Sahlu  
[michael.sahlu@hahu.jobs](mailto:michael.sahlu@hahu.jobs)

