

More of Panda CSS, Routing, Client vs Server components, Server-side rendering

Darjan Baričević

01

Routing

02

Server-side rendering

03

Client vs Server components



01

Routing

Routing

- URL addresses specific resource on the Internet (page, response, ...)
- Makes user navigation easier (refresh, browser back)
- Process of navigating to the specific resource on the web

SEO (Search Engine Optimization)

- Process of making a website more visible in search results
- Search engines crawl web, index content from pages, points to relevant pages in search results
- Ranking algorithm is secret, known only by search engine companies.

SEO (Search Engine Optimization)

- More traffic -> More 💰



Server side routing

- Route transition is handled on the server
 - When the URL changes, a new HTML document is retrieved from the server
- Browser has to communicate with the server
- Good:
 - Minimal data for each page
 - Search engine friendly
- Bad:
 - Slower interaction between pages
 - Full refresh of a page -> Context is lost
- Does not necessarily imply server-side rendering

Client side routing

- Route transition is handled on the client
 - When URL changes, new HTML is not needed, current one is changed
- Good:
 - Faster page transitions
 - Preserves context
 - Page transitions can be animated
- Bad:
 - Larger first load
 - Possible extra data
 - Not so friendly to search engines - they have to render the page like a true browser
- First load can be server-side rendered, but it isn't default in React

02

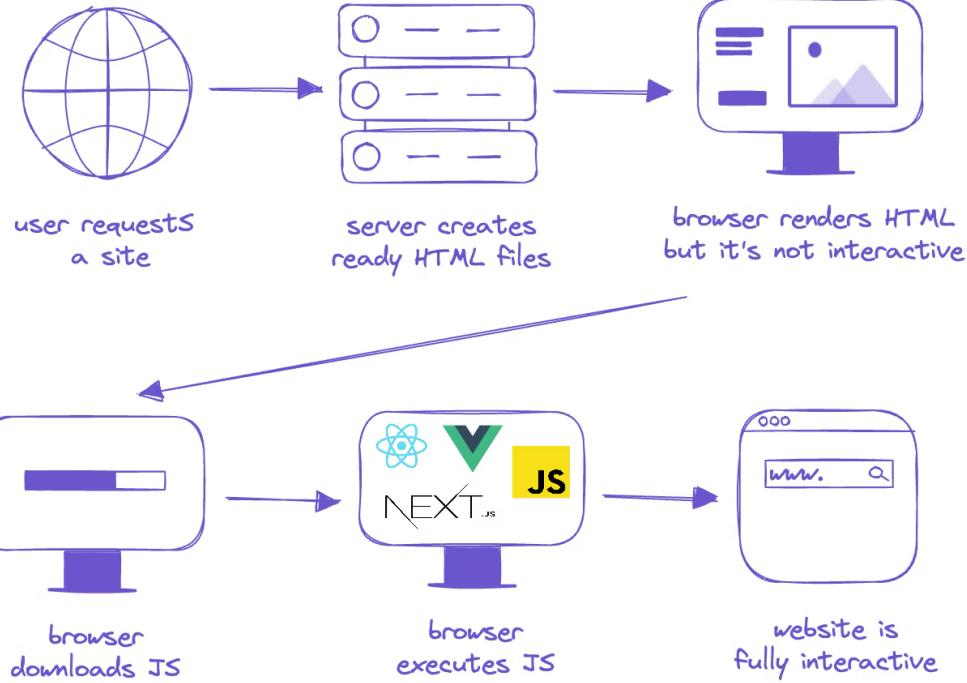


Server-side rendering

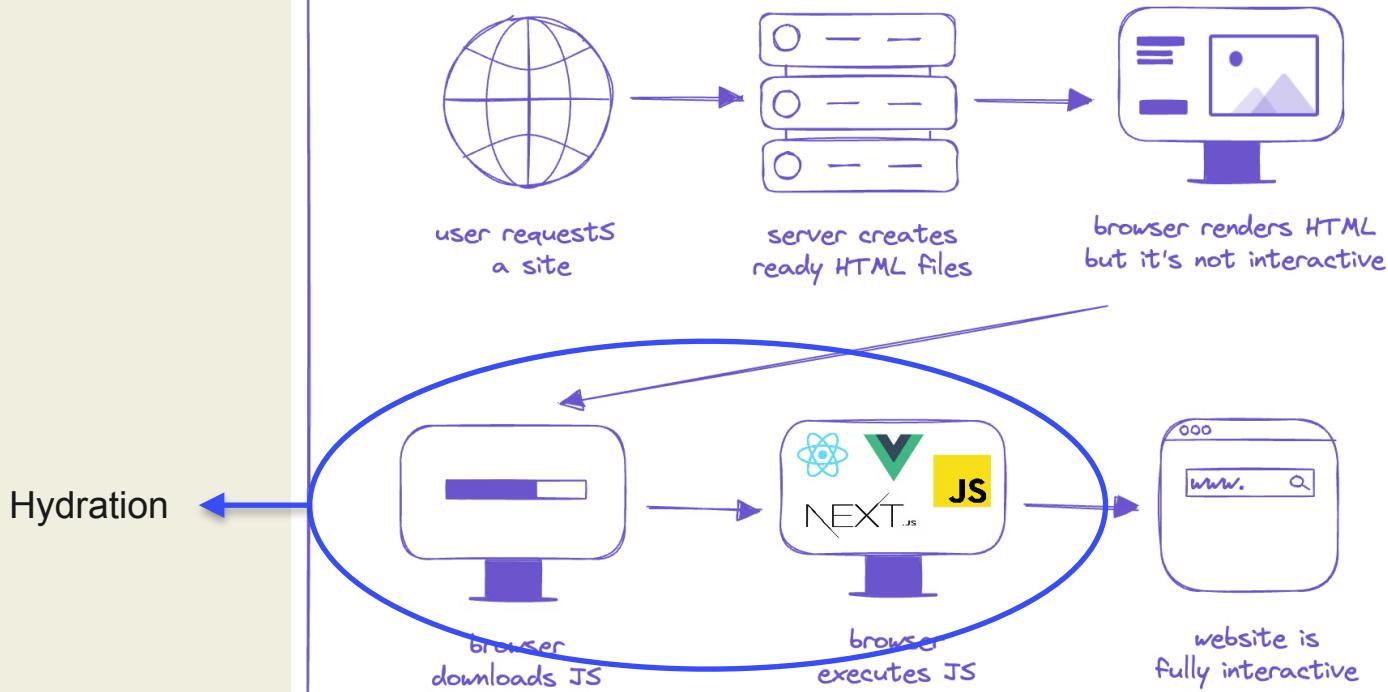
Introduction

- technique that allows us to generate HTML on the server and send it back to client
- opposite of client-side rendering (**CSR**) where everything is rendered on the client
- why do we want that? better **performance** and **SEO**, of course
- nothing new, this is basically the default mode of every MPA framework/application (Symfony, ASP.NET, JSP etc.)
- why is it trending now? because now we can leverage new technologies to develop rich and SEO-optimised applications **without losing SPA experience**
- Most meta frameworks offer this feature by default (Next, Nuxt, SolidStart, SvelteKit etc.)

Server-side Rendering (SSR)



Server-side Rendering (SSR)



Hydration

- process of transforming a server-rendered HTML document into a full interactive web application
- involves attaching event handlers to HTML elements and restoring application state from the backend
- if application state can't be restored -> [hydration errors](#)

Drawbacks

- increased expenses - to deploy an SSR application, we need a backend
- more configuration - configure rendering strategy (SSG, ISG) based on your use case
- increased complexity - now you have to handle both client and server
- compatibility issues - lots of third-party packages are not compatible with SSR

Client vs server

- Server
 - does not have access to browser APIs (document, localStorage etc.)
 - usually Node runtime environment
 - initialises application state
- Client
 - has access to browser APIs
 - restores (continues) application state in process of hydration

**Thank you for your
attention!**

