SPA

Single Page Application

What is SPA

- Single Page Application
- simulate behaviour of desktop applications we remain in a single page and content is loaded dynamically
- **initial load -** The first load of the page we grab the HTML and other resources referenced from the HTML
- after the **initial load** all other load from the server is based on AJAX
- URL can still change using history api in HTML5
- reloading the page should lead to the same state

Advantages of SPA

- better UX
- less server requests to download a new full page
- server works less

Disadvantages of SPA

- requires JS
- initial load can be slow
- SEO
- using SPA frameworks makes larger JS files
- usually more memory cpu consuming sites

- Initial load and SEO can be improved by using Server Side Rendering

Server Side Rendering

Usually in SPA the initial HTML we get from the server looks similar to this:

```
<html>
...
<body>
        <script src=”my-spa-app.js”></script>
</body>
</html>
```

- the body is empty and contains a download of a script
- the script is running and in charge of presenting the page to the user
- initial load can be slow - we need to grab the html and js and only after that we need to run the js to render the page
- Unfriendly to search engines

Server Side Rendering

- with server side rendering the initial HTML we get from the server looks like this:

```
<html>
    ...
    <body>
        <h1>The full page is rendered by the server</h1>
        <p>we get a full html page like it was when there was no SPA</p>
        <p>We can also use server caching so the html will be sent faster</p>
        ....
        <script src=”my-spa-app.js”></script>
    </body>
</html>
```

- the first html is sent by the server, after that the spa takes control and everything is loaded dynamically
- modern frameworks/libraries like angular2/react support SSR - we have to make sure that the code we write is universal

History timeline

- HTTP Protocol was invented in 1989
- The first web browser was released in 1991, The first graphic one Mosaic was released in 1993
- First version of HTML was released in 1992
- HTML5 published in 2014
- JS was invented in 1995 and was combined with Netscape browser
- In 1996 iFrame was introduced by IE
- In 1998 first version of Ajax which started as XMLHttp by IE
- 2003 First SPA concept and 2004-2005 first spa applications start to emerge
- jQuery released in 2006
- Angular1 released 2009
- ReactJs was put in production in 2011 and was open sourced in 2013
- Angular2 first beta was 2015 first release in end of 2016

Challenges of SPA

- Templates
- Routing
- SSR
- binding from inputs to JS
- Taking care of forms
- rerender page when needed

SPA Framework

- It was challenging to create SPA so as a result frameworks were released to help us create SPA
- Frameworks direct us to their way of developing SPA
- Frameworks usually have a very large code base which may have a substantial effect on memory and cpu usage

Angular5

- SPA Framework
- built with TypeScript
- has it’s own templating language based on HTML
- you extend that templating language by adding tags or classes that creates UI components
- take care of re rendering the components when needed
- works for node server side as well

Angular5 Advantages

- typescript
- reusable components
- easy testing
- much faster then angular 1

Disadvantages of Angular5

- High memory usage
- Large JS files
- High CPU
- Performance is slower than React

React - SPA library

- React is open source js library maintained by facebook
- Library helps us create UI components
- React manage the state of component and re render the component when needed
- React is doing changes to the DOM very fast using VirtualDOM
- React is a library and not a framework and does not constrain you to a certain way of developing things
- React is fast with a minimal footpring on resources
- Not possible to create large SPA with just react and not using some other libraries like Redux or Flux
- React has a large open source community that on top of the library built frameworks and other libraries to help us create SPA application

React VS Angular

- The question should change to: When React and When Angular
- using npm you can split your frontend application to multiple packages
- you can use in your project Angular and React without mixing the framework/libraries and keep separation of frameworks in each project