**React JS**

**Day 1st**

**Pre-Requisites for ReactJS (HTML, CSS, JavaScript, DOM)**

**What is SPA??**

Single Page Application is a type of Application and it does not have any page refresh or reload and it is more faster compared to MPA. It is having separation like frontend is separate and backend is separate. You can cache data, responses in client side and based on that one you can make old req and new req.

Here, you can avoid HTTP requests with a fewer HTTP request which makes your application faster by reducing the server efforts. So that you can make a better application if you have more HTTP requests then your application will be slow.

**How to optimize your application??**

You can reduce the number of requests. Here easily you can categorize the old and new requests. If it is old request then you can get the data from the cache. This way we can say it is optimized.

**Disadvantages??**

SEO (how)

**AJAX (Asynchronous JavaScript & XML)**

**Single Page Application??**

A Single Page Application is an app that works inside a browser and does not require page reloading during use. You are using this type of applications everyday. These are, for instance: Gmail, Google Maps, Facebook, or Github. SPAs are all about serving an outstanding UX by trying to imitate a natural environment in the browser – no page reloads, no extra wait time. It is just one web page that you visit which then loads all other content using JavaScript – which they heavily depend on.

SPA requests the markup and data independently and renders page straight in the browser. We can do this thanks to the advanced JavaScript frameworks like AngularJS, Ember JS, Meteor JS, Knockout JS.

**Pros of the single Page Application**

* SPA is fast, as most resource (HTML, CSS, Scripts) are only loaded once throughout the lifespan of application. Only data is transmitted back and forth. It stores data in the cache.
* The development is simplified and streamlined. There is no need to write code to render pages on the server. It is much easier to get started because you can usually kick off the development from a <file://URI>, without using any server at all.
* SPAs are easy to debug with chrome, as you can monitor network operations, investigate page elements and data associated with it.
* It’s easier to make a mobile application because the developer can reuse the same backend code for web application and native mobile.
* SPA can cache any local storage effectively. An application sends only one request, store all data, tehn it can use this data and works even offline.

**Cons of the Single-Page-Application**

* It is very tricky and not an easy task to make SEO optimization of a single page application. Its content is loaded by AJAX – a method of exchanging data and updating in the application without refreshing the page.
* It is slow to download because heavy client frameworks are required to be loaded to the client.
* It requires JavaScript to be present and enabled. If any user disables JavaScript in his or her browser, it won’t be possible to present application and its actions in a correct way.
* Compared to the traditional application, SPA is less secure. Due to Cross-Site-Scripting (XSS), it enables attackers to inject client side scripts into web application by other users.
* Memory leak in JavaScript can even cause powerful system to slow down.

**Multi Page Application**

Multiple-Page-Applications work in a traditional way. Every change display the data or submit data back to the server requests rendering a new page from the server in the browser. These applications are large, bigger than SPAs because they need to be. Due to the amount of content, these applications have many levels of UI. Luckily it’s not a problem anymore. Thanks to AJAX, we don’t have ot worry that big and complex applications have to transfer a lot of data between server and browser. That solution improves and it allows to refresh only particular parts of the application. On the other hand, it adds more complexity and it is more difficult to develop than a single page application.

**XHR (XML HTTP request) MDN**

XML HTTP request objects are used to interact with servers. You can retrieve data from a URL without having to do a full page refresh. This enables a web page to update just part of a page without disrupting what the user is doing.

XML HTTP request is used heavily in AJAX programming.

**How to create AJAX requests**

let ajaxRequest = new XMLHttpRequest();

Use XMLHttpRequest objects to interact with servers. You can retrieve data from a URL without having to do a full page refresh. This enables a web page to update just part of a page without disrupting what the user is doing.

console.log(ajaxRequest)

It returns you object with a lots of properties.

Let’s take an example connecting an API

ajaxRequest.Open(‘get’,’https://api.github.com/users’);

ajaxRequest.onload = function(info){

console.log(ajaxRequest.response) // it will return you progress even object

}

ajaxRequest.send()

It means we are sending a request to github request to give the particular data.

**Pros of the MPAs:**

* It’s the perfect approach for users who need a visual man of where to go