**Introduction**

NuxtJs replacing the Vue SSR (server side rendering) with a cool features and make it easier and highly optimized

**Commands**

Npm install -g create-nuxt-app

To get started:

npm run dev

To build & start for production:

npm run build

npm run start

~/ means root folder (in JS)

~asset means asset in root folder (use it only in <style> or <Template> )

<template>

<img src="~/assets/your\_image.png" />

<img :src="require(`~/assets/img/${image}.jpg`)" />

</template>

background: url('~assets/banner.svg');

**Adding general CSS**

Go to nuxt.config.js file and in css line add its location (in asset/css for example)

**Adding general link**

Go to nuxt.config.js file and in links line add the link

**Routes**

Nuxt create automatic routes in pages folder

/ => index.vue page

/name => pageName.vue

**Nuxt dynamic routes**

Under score (\_) in nuxt refer to params in Vue

So if create \_id folder and add index it we will be like /params

Ex: test/\_id/index.vue

Go to URL test/\_id/4

$route.params.id will be = 4

**asyncData() vs data()**

asyncData is more better with the server rendering   
with it use return before getting API

Ex: asyncData() {

return axios.get('/api/v1/products').then(response => {

// Note that you can't access the `this` instance inside asyncData

let products = response.data

})

}

**Control title and meta for every page**

head() {

      return {

         title: "About Page",

         mete: [{

            hid: "hid description",

            name: "description ",

            content: " best website for blablabla",

         }]

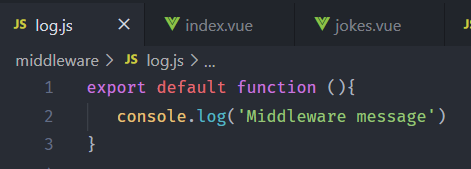
      }

   }

**Middleware**

Middleware uses to execute a function before route

Set a middleware: Just create a js file in middleware folder



Use it in a page

export default {

  middleware: ["log"],

  data() {

    return {};

  }

}

To use it in every router open nuxt.config.js and add this:

  router:{

    middleware:['log']

  }

**Validate**

Is a special Nuxt method used to excute before route

export default {

  validate(data) {

    console.log('validate => ', data)

    return true *//if false => page or component will not load*

  }

}

**Nested Routes**

Create vue file with name of folder but parallel not not inside the folder

<template>

<section class="container">

   <h5>Header for (Folder Name) Components and Pages</h5> *<!-- this will be in all traget folder components-->*

   <nuxt-child /> *<!-- <== HEREEEEEEE  -->*

</section>

</template>

**Layout**

In Layouts Folder Nuxt app uses the default layout

We can make a new layout and use it for some pages

First Create new layout in Layout folder then call it in any page you like by:

export default {

  layout: "users"

};

There is a special layout page in Nuxt called error.vue it’s like 404 page

Create it to show it while error happens