

Pages/\_app.tsx

import { AppProps } from 'next/app';

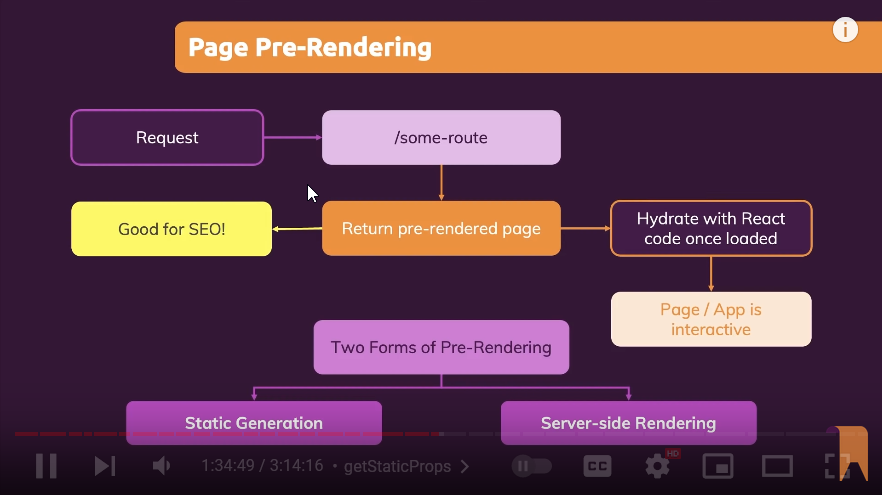
import '../styles/globals.css';

const MyApp = ({ Component, pageProps }: AppProps) => {

  return <Component {...pageProps} />

 };

export default MyApp;



Static Generation

When the application is built for production by the developer, the pages are created. By default NextJS for static generation.

If a component has the getStaticProps , the NextJS first executes the function before creation of the component,

getStaticProps 🡪 can be async and is used to keep the props ready for the page. It must return an object with props key

const NewsPage: NextPage = (props) => {

  return <Fragment>

    <h1>The News Page</h1>

    <ul>

      <li> something else</li>

    </ul>

  </Fragment>;

};

export async function getStaticProps() {

  //fetch data from an API

  return {

    props: {

      meetups:DUMMY\_MEETUPS

    },

    revalidate: 10,

  }

}

One problems may be due to dynamic change, hence we set revalidate which means the number of seconds NextJS waits to revalidate if there are incoming requests

Will always run on server for every requests

On-Fly after deployment

export async function getServerSideProps(context) {

  const req = context.req;

  const res = context.res;

  return {

    props:{

    }

  }

}

GetStaticPaths

export async function getStaticPaths() {

  return {

    fallback:true, // somepath instead of all paths

    paths: [{

      params: {

        meetupId: 'm1',

      }

    },

      {

        params: {

        meetupId:'m2'

        }

      },

    ]

  }

}

For API Routes

/pages/api/new-meetup.js:

Function handler(req,res){}

const response = await fetch(‘api/meetup’);