**Data Fetching**

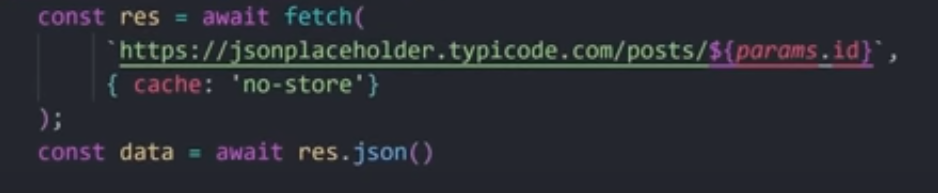
Next.js provides 3 choices for selecting how to fetch data.

1. Server Side Rendering (SSR)
2. Static Site Generation (SSG)
3. Incremental Static Generation (ISR)

SSR:

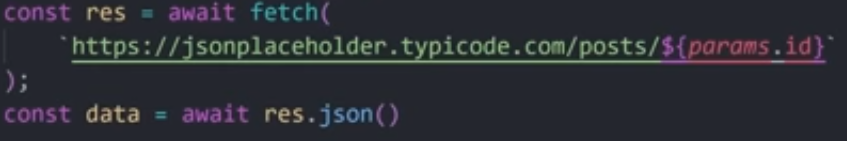
It means Dynamic server rendered data. It is fetched fresh on each request. With SSR each request to the server triggers a new rendering cycle and data fetch. Ensuring that the content is always up to date.

So, to implement this we use  
{cache: ‘no-store’} object, which means it tells server don’t store it just simply call it.  
This is going to ensure that it refetches it every single time which means that its server side rendered.



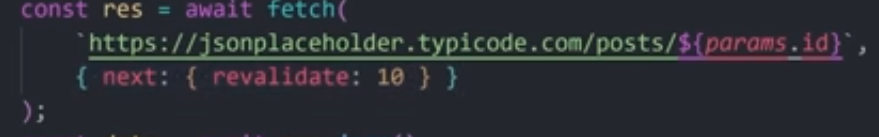
SSG:-

The only this we’ve to do is to remove that {cache: ‘no-store’} object. That means that by default, Next.js uses Static Site Generation (SSG). It will automatically fetch the data but it will *also cache it*. This method is ideal for Content that doesn’t change frequently such as blog posts, documentation or marketing pages. First time it’s going to make a fetch and then it’s already going to have the data and it’s simply gonna display it.



Static Site Generation

**ISR:**Instead of messing with the cache we can provide an additional parameter here called {next: { revalidate: 10 }}. It combines the benefits of SSR and SSG for dynamic content in static sites with incremental static generation you can specify certain data to be statically fetched at build time while defining a revalidation time interval. This means that the data will be cached but after a specific time frame it’s then going to refresh it and you’re always going to have new data making this The Best of Both Worlds for the Dynamic content.

****