Routing and Navigation

Terms

Client cache
Dynamic routes
Layout
Prefetching

Summary

- The new App router in Next.js uses convention over configuration to define routes. It looks for special files such as page.tsx, layout.tsx, loading.tsx, route.tsx, etc.

- With the App router, we can colocate our pages and their building blocks (eg components, services, etc). This helps us better organize our projects as we can keep highly related files next to each other. No need to dump all the components in a centralized components directory.

- A *dynamic route* is one that takes one or more parameters. To add parameters to our routes, we wrap directory names with square brackets (eg [id]).

- In standard React applications, we use the state hook for managing component state. In server-rendered applications, however, we use query string parameters to keep state. This also allows us to bookmark our pages in specific state. For example, we can bookmark a filtered and sorted list of products.

- We use *layout* files (layout.tsx) to create UI that is shared between multiple pages. The root layout (/app/layout.tsx) defines the common UI for all our pages. We can create additional layouts for specific areas of our application (eg /app/admin/layout.tsx).

- To provide smooth navigation between pages, the Link component prefetches the links that are in the viewport.

- As the user moves around our application, Next.js stores the page content in a cache on the client. So, if they revisit a page that already exists in the cache, Next.js simply grabs it from the cache instead of making a new request to the server. The client cache exists in the browser's memory and lasts for an entire session. It gets reset when we do a full refresh.

File Conventions

```
page.tsx
layout.tsx
loading.tsx
not-found.tsx
erorr.tsx
```

Accessing Route and Query String Parameters

```tsx
interface Props {
  params: { id: string },
  searchParams: { sortOrder: string }
}

export default function Home({ params, searchParams }: Props) {
}
```

Programmatic Navigation

```tsx
const router = useRouter();
router.push('/')
```