Routing & Page Rendering

File-based Routing, Component Types & More

- Understanding **Routing** In NextJS Applications
- **File Name** Conventions & Project **Structure**
- **Server Components** vs **Client Components**

A Challenge For You

Add **two new pages**

/news → **Show a list of news item links**

(for this exercise, it's enough to just output some dummy text)

/news/<id> → **Show a detail page for a news item**

(for this exercise, it's enough to just output some dummy text)

Also add a <MainHeader> component that contains links to the (already existing) "Home" page and the newly added "News" page

Standard Routes

/about

Folder name = route path

app/about $\longrightarrow$ my-website.com/about

Dynamic Routes

/posts/<dynamic>

Define dynamic segments by wrapping the folder name with []

app/posts/[slug] $\longrightarrow$ my-website.com/posts/next-is-awesome

Layouts & Pages

Pages

Created via **page.js** file

Define page content (JSX) for a route

Layouts

Created via **layout.js** file

Define wrapping layout for one or more pages

Server vs Client Components

React Server Components (RSC)

Components that are **only** rendered on the server

By default, all React components (in NextJS apps) are RSCs

Advantage: Less client-side JS, great for SEO

Client Components

Components that are **pre-rendered** on the server but then also **potentially on the client**

Opt-in via “use client” directive

Advantage: Client-side interactivity

Not Found & Errors

Not Found Pages

Created via **not-found.js** file

Shown if a “Not Found” (404) error occurred

Error Fallback Pages

Created via **error.js** file (or **global-error.js**)

Shown if an error gets thrown by a child page or layout

Loading UI

Loading Fallback Page

Created via **loading.js** file

Shown if the child page is still loading data

Route Handlers (API Routes)

/

Non-Page Routes

Created via **route.js** file

Manually handle request & send (any) response

Caching

/

Non-Page Routes

Created via **route.js** file

Manually handle request & send (any) response

Parallel Routes

/dashboard

Load multiple pages in one page by prefixing folder names with @

$$\begin{array}{l} \text{app/dashboard/@analytics} \\ \text{app/dashboard/@expenses} \end{array} \longrightarrow \text{my-website.com/dashboard}$$

Intercepting Routes

/dashboard

Intercept navigation requests by prefixing folder names with (.), (..) etc

app/expenses/(.)new $\longrightarrow$ my-website.com/expenses

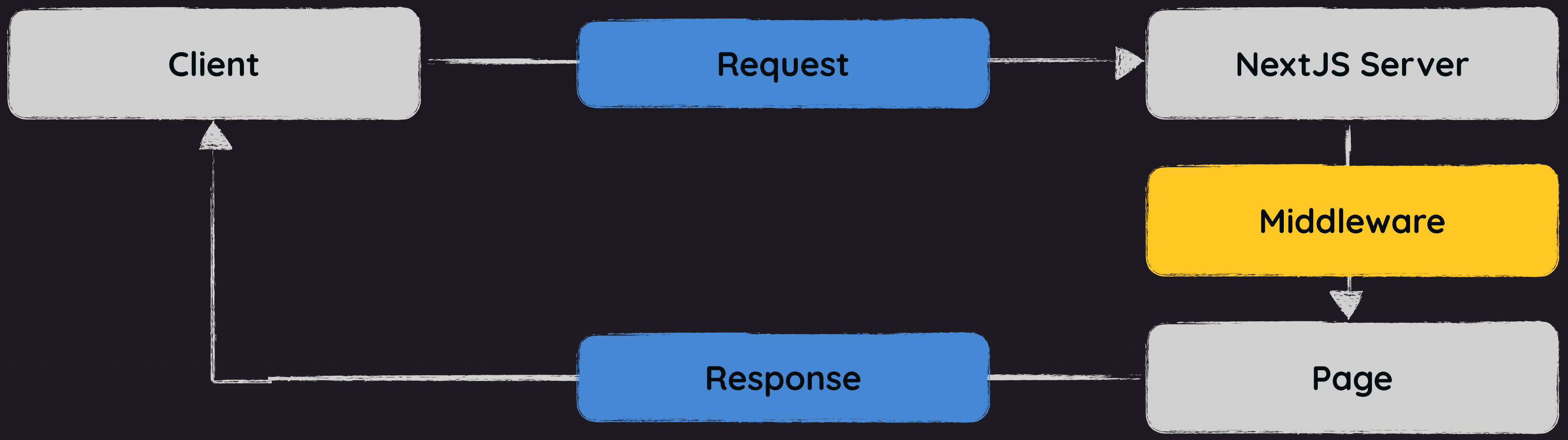
Route Groups

/posts vs /welcome

Define groups by wrapping folders with ()

$$\text{app/(marketing)/welcome} \longrightarrow \text{my-website.com/welcome}$$

Middleware


Client
Request
NextJS Server
Middleware
Page
Response