**NextGen UI-frameworks**

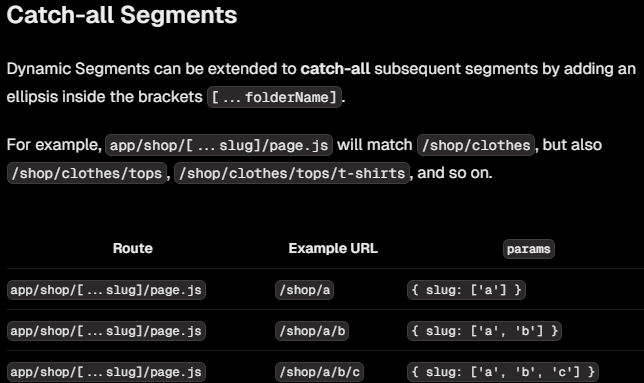
Введение…………………………………………………………………………

1. Next.js………………………………………………………………………...
   1. Область проблем, различия с SPA…………………………………………
   2. App router…………………………………………………………………….
      1. Файловая структура роутинга…………………………………………..
      2. Зарезервированные компоненты……………………………………….
      3. Разделение компонентов на роли (client/server) ……………………….
      4. Формирование страницы………………………………………………..
      5. Серверные компоненты и стратегии рендеринга………………………
   3. Гидратация и регидратация………………………………………………...

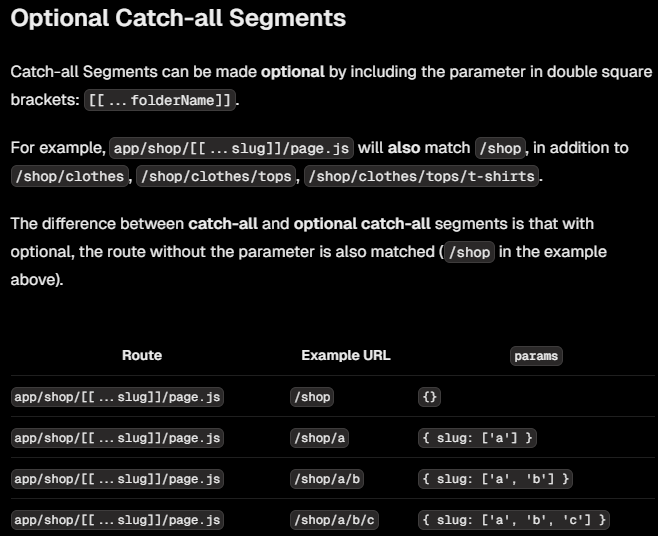
Введение

1. Next.js
   1. Область проблем, различия с SPA
   2. **App router**
      1. **Файловая структура роутинга**

• Catch all segments



• Optional catch all segments



* + 1. **Зарезервированные компоненты**

• layout – это компонент, у которого рендеринг происходит всего один раз за весь жизненный цикл приложения и «кэшируется». Особенности:

Рутовый layout можно использовать как место инициализации глобального состояния Redux, Effector.

Часто используется для включения на каждую страницу приложения хэдэра и футтера.

• template – схож с layout с той лишь разницей, что рендеринг такого компонента осуществляется **при каждом** запросе страницы сегмента.

• error

• loading

• not-found

• page

* + 1. **Разделение компонентов на роли (client/server)**

• Дефолтно все компоненты являются серверными, используя SSR и SSG.

• Определить роль компонента – директива вначале скрипта «use client».

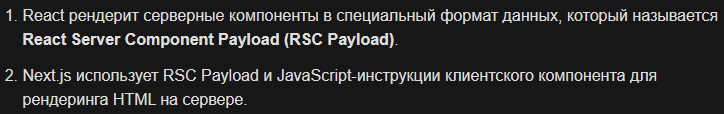
Логика разделения ролей между компонентами:



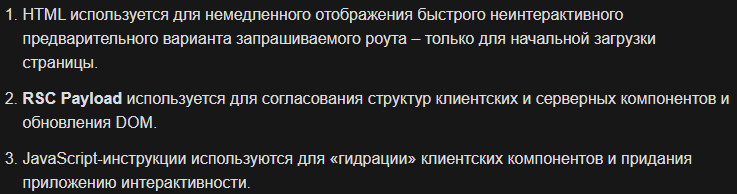
* + 1. **Формирование страницы**

Формирование результирующей страницы происходит в 2 этапа:

1. Формирование страницы на стороне сервера



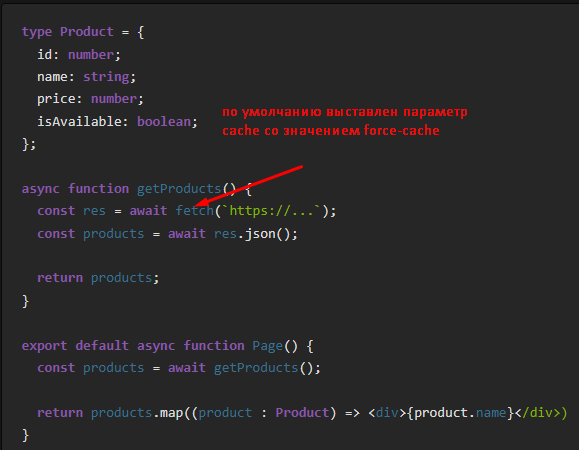
1. Формирование страницы на стороне клиента



* + 1. **Серверные компоненты и стратегии рендеринга**

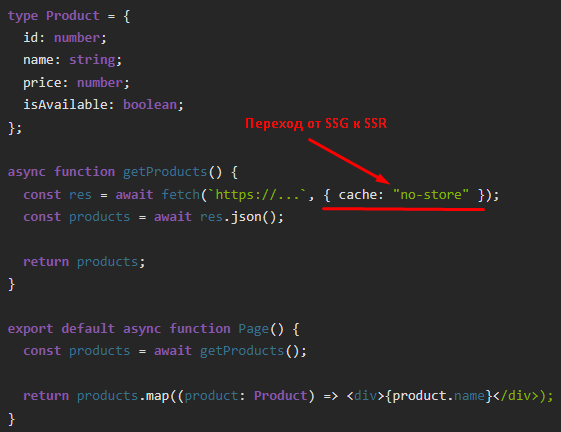
1. Статическая генерация SSG

По умолчанию. Любые результаты запросов на бэкэнд кэшируются (Соответствует getStaticProps у PageRouter).



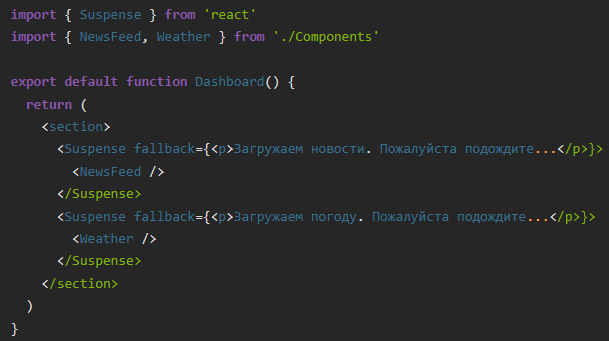
1. Серверный рендеринг SSR

Результаты запросов на бэк не кэшируются. Компонент проходит стадию рендеринга на сервере во время каждого запроса страницы сегмента, где есть этот компонент (Соответствует getServerSideProps у PageRouter).

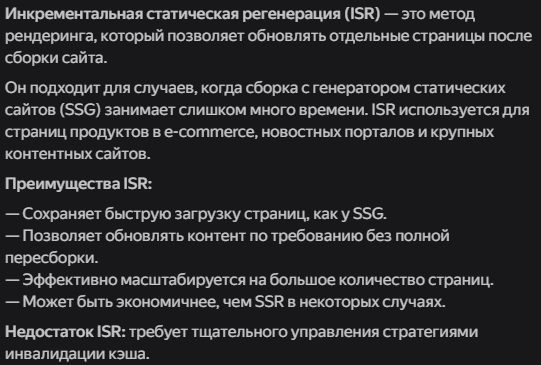


2.1) Стриминг

Специальный вид SSR, который используется для рендеринга серверного компонента, включающего в себя несколько независимых асинхронных блоков или других серверных компонентов. Каждый такой блок или компонент оборачивается в React-Suspense с заданным fallback. Цель – показать временный UI, пока внутренний JSX не завершит рендеринг.



1. Инкрементальная регенерация ISR



4) Клиентский рендеринг CSR

* 1. Гидратация и регидратация