Next.js навсегда?

Год использования в проекте

Владимир Морозов

Владимир Морозов

- Вебмастер;

Владимир Морозов

- Вебмастер;
- Fullstack;

Владимир Морозов

- Вебмастер;
- Fullstack;
- Frontend;

Владимир Морозов

- Вебмастер;
- Fullstack;
- Frontend;
- Frontend lead, команды Портал, Цифромед.

План

- Создадим приложение на Next.js;

План

- Создадим приложение на Next.js ;
- Поговорим, какие были сложности;

План

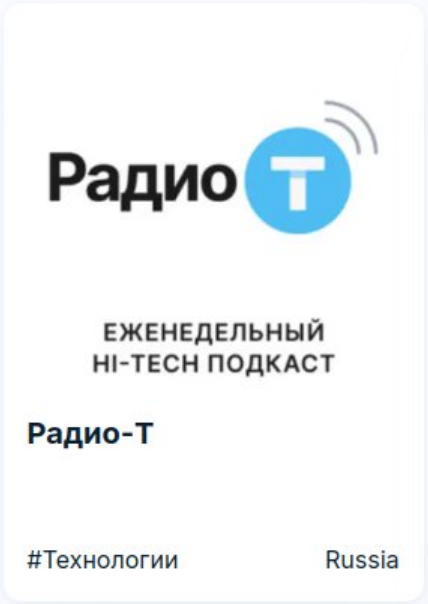
- Создадим приложение на Next.js;
- Поговорим, какие были сложности;
- Подумаем, куда движется индустрия.

Примерно год назад

Требования

Root podcasts Статистика Выйти

Выбор редакции:



Радио-Т

#Технологии Russia

Веб-стандарты

#Технологии Russia

Акулы Пера

#Культура Russia

Три тимлида заходят в бар

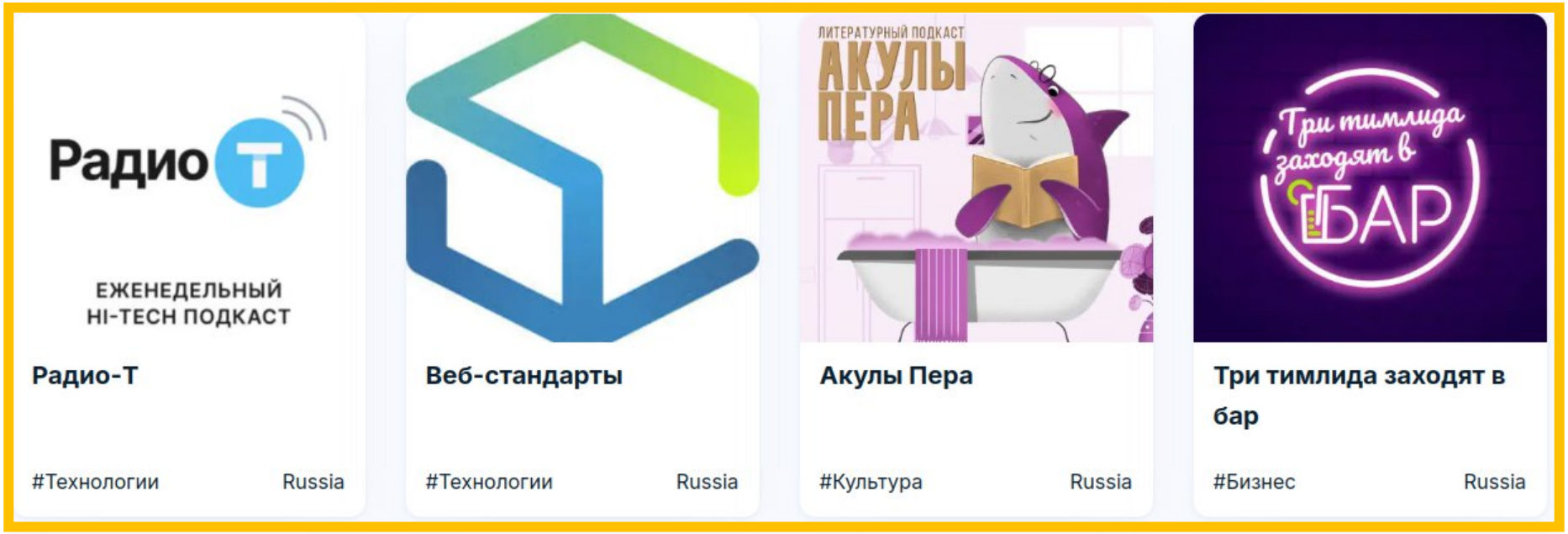
#Бизнес Russia

Тема

Страна

Root podcasts Статистика Выйти

Выбор редакции:

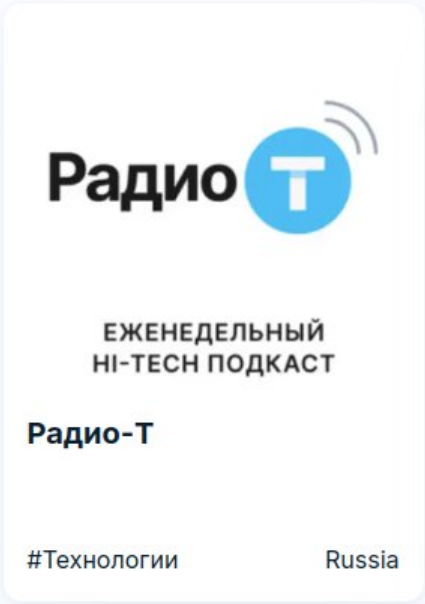


Тема

Страна

Root podcasts Статистика Выйти

Выбор редакции:



Радио-Т

#Технологии Russia

Веб-стандарты

#Технологии Russia

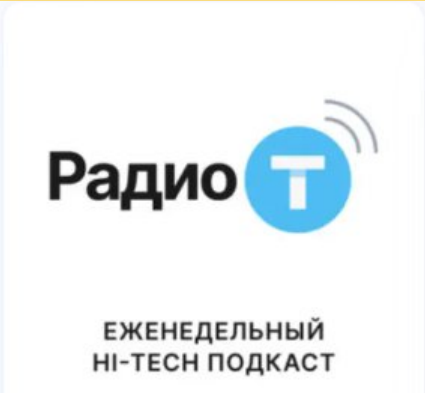
Акулы Пера

#Культура Russia

Три тимлида заходят в бар

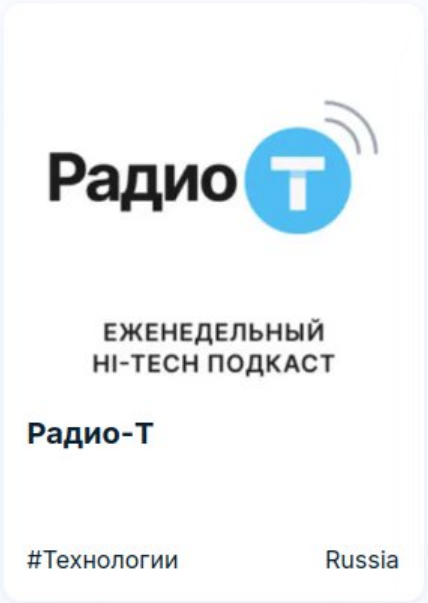
#Бизнес Russia

Тема Страна



Root podcasts Статистика Выйти

Выбор редакции:



Радио-Т

#Технологии Russia

Веб-стандарты

#Технологии Russia

Акулы Пера

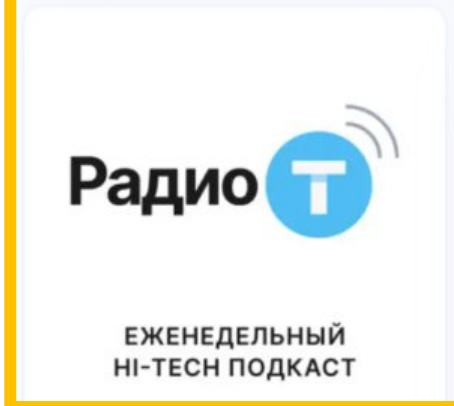
#Культура Russia

Три тимлида заходят в бар

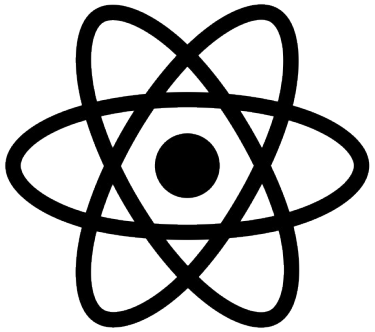
#Бизнес Russia

Тема

Страна

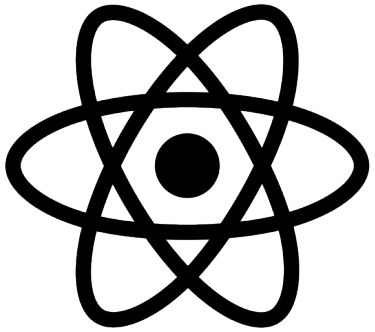


Требования



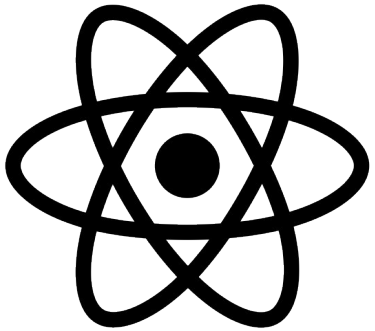
React

Требования



React

Требования



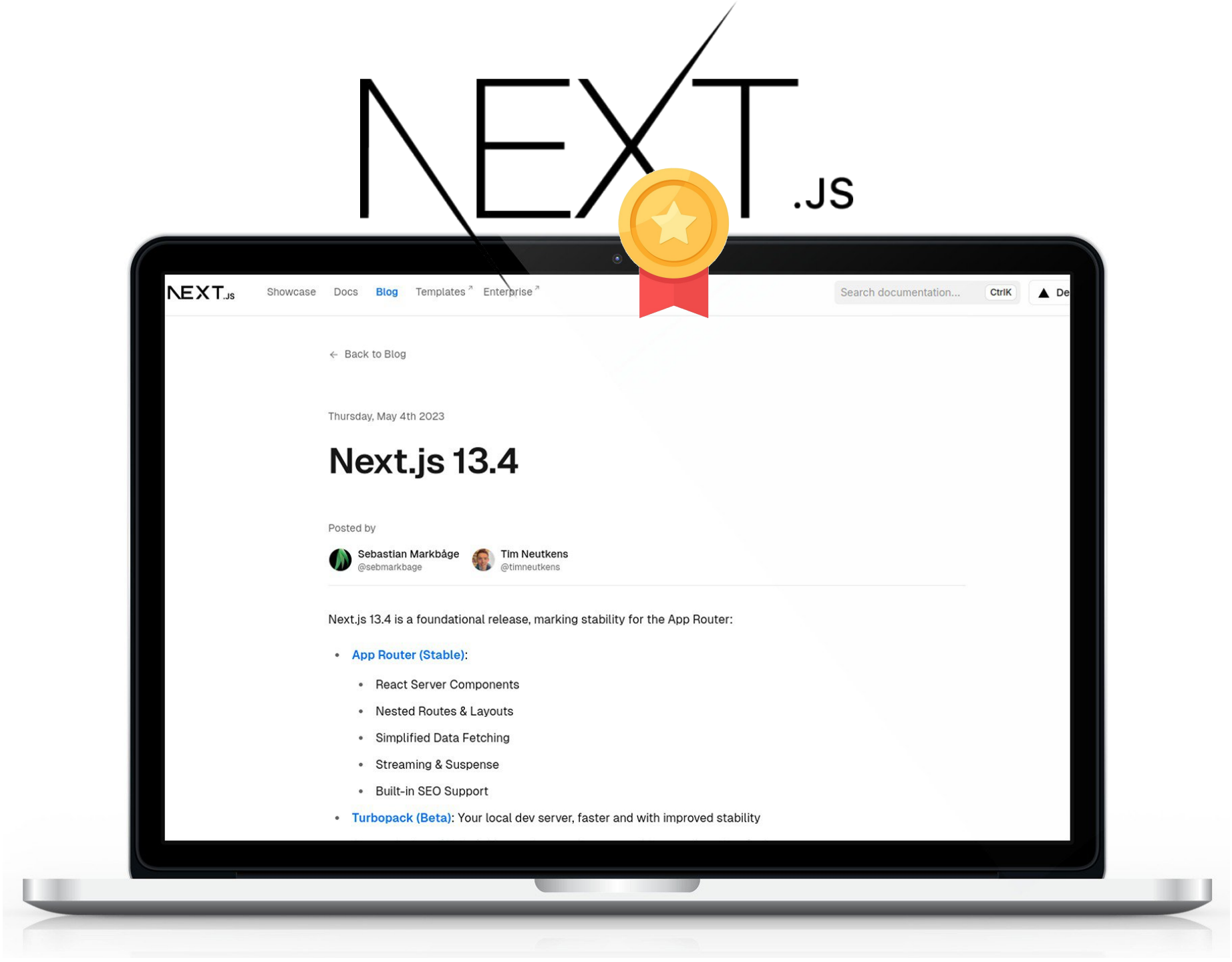
React

Финалисты

Финалисты

Remix

NEXT.JS


NEXT.JS
Showcase
Docs
Blog
Templates
Enterprise
Search documentation...
CtrlK
Back to Blog
Thursday, May 4th 2023
Next.js 13.4
Posted by
Sebastian Markbåge
@sebmarkbage
Tim Neutkens
@timneutkens
Next.js 13.4 is a foundational release, marking stability for the App Router:
App Router (Stable):
React Server Components
Nested Routes & Layouts
Simplified Data Fetching
Streaming & Suspense
Built-in SEO Support
Turbopack (Beta): Your local dev server, faster and with improved stability

Количество пользователей больше, чем у Remix

Next.js

Next.js

Next.js is a React framework for building full-stack web applications

90K	26	Top 2%
Members	Online	Rank by size

Remix.js

remixrun

This Subreddit is about the Meta-Framework Remix for React.

720	Top 23%
Members	Rank by size

Отображение данных

Client-Side Rendering

```
useEffect(() => {
  const load = async () => {
    const res = await fetch(
      `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts?${searchParams}`
    );
    const data = await res.json();
    setPodcasts(data);
  };

  load();

}, [searchParams]);
```

#Технологии Russia

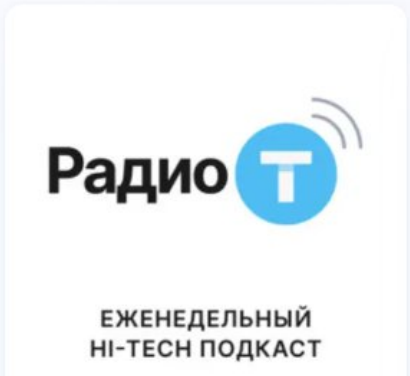
#Технологии Russia

#Культура Russia

#Бизнес Russia

Тема

Страна

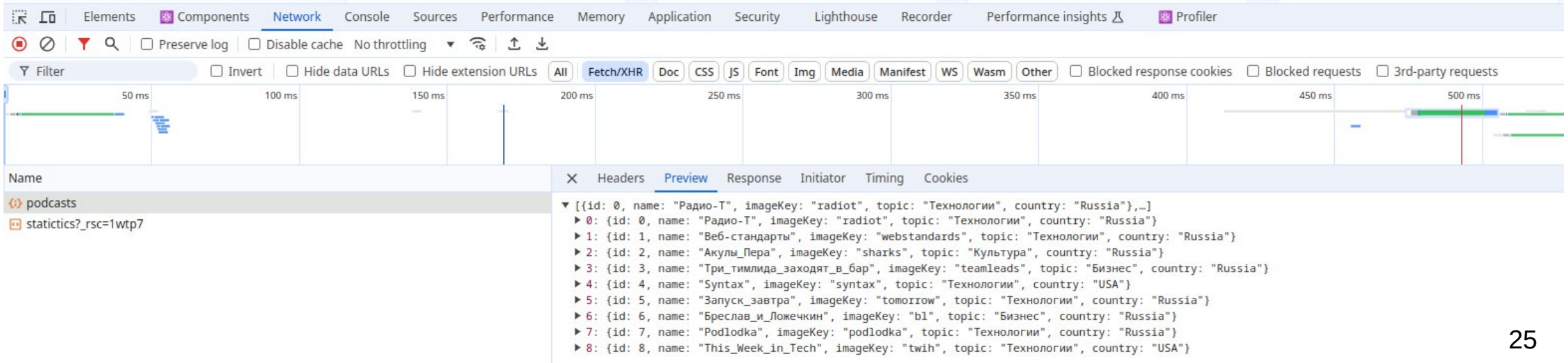


Радио-Т

Веб-стандарты

Акулы Пера

Три тимлида заходят в



Server-Side Rendering

```
async function getFilters() {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts/filters`
  );
  const { countries, topics } = (await res.json()) satisfies FirtersResponse;

  return (
    <div className={styles.filters}>
      <Suspense>
        <Search />
        <div className={styles.selects}>
          <Select searchParam="topic" elements={topics} placeholder="Тема" />
          <Select
            searchParam="country"
            elements={countries}
            placeholder="Страна"
          />
        </div>
      </Suspense>
    </div>
  );
}
```

Server-Side Rendering

```
async function getFilters() {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts/filters`
  );
  const { countries, topics } = (await res.json()) satisfies FirtersResponse;

  return (
    <div className={styles.filters}>
      <Suspense>
        <Search />
        <div className={styles.selects}>
          <Select searchParam="topic" elements={topics} placeholder="Тема" />
          <Select
            searchParam="country"
            elements={countries}
            placeholder="Страна"
          />
        </div>
      </Suspense>
    </div>
  );
}
```

Server-Side Rendering

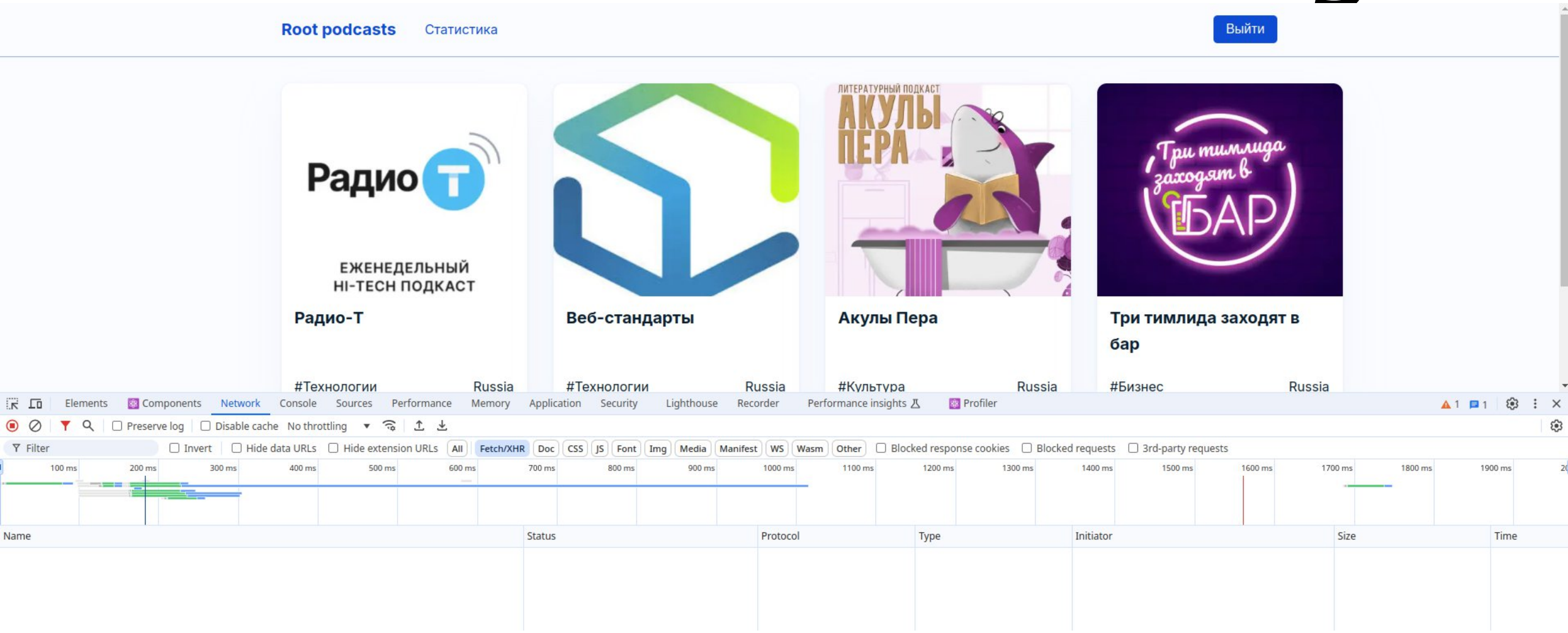
```
async function getFilters() {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts/filters`
  );
  const { countries, topics } = (await res.json()) satisfies FirtersResponse;

  return (
    <div className={styles.filters}>
      <Suspense>
        <Search />
        <div className={styles.selects}>
          <Select searchParam="topic" elements={topics} placeholder="Тема" />
          <Select
            searchParam="country"
            elements={countries}
            placeholder="Страна"
          />
        </div>
      </Suspense>
    </div>
  );
}
```

Server-Side Rendering

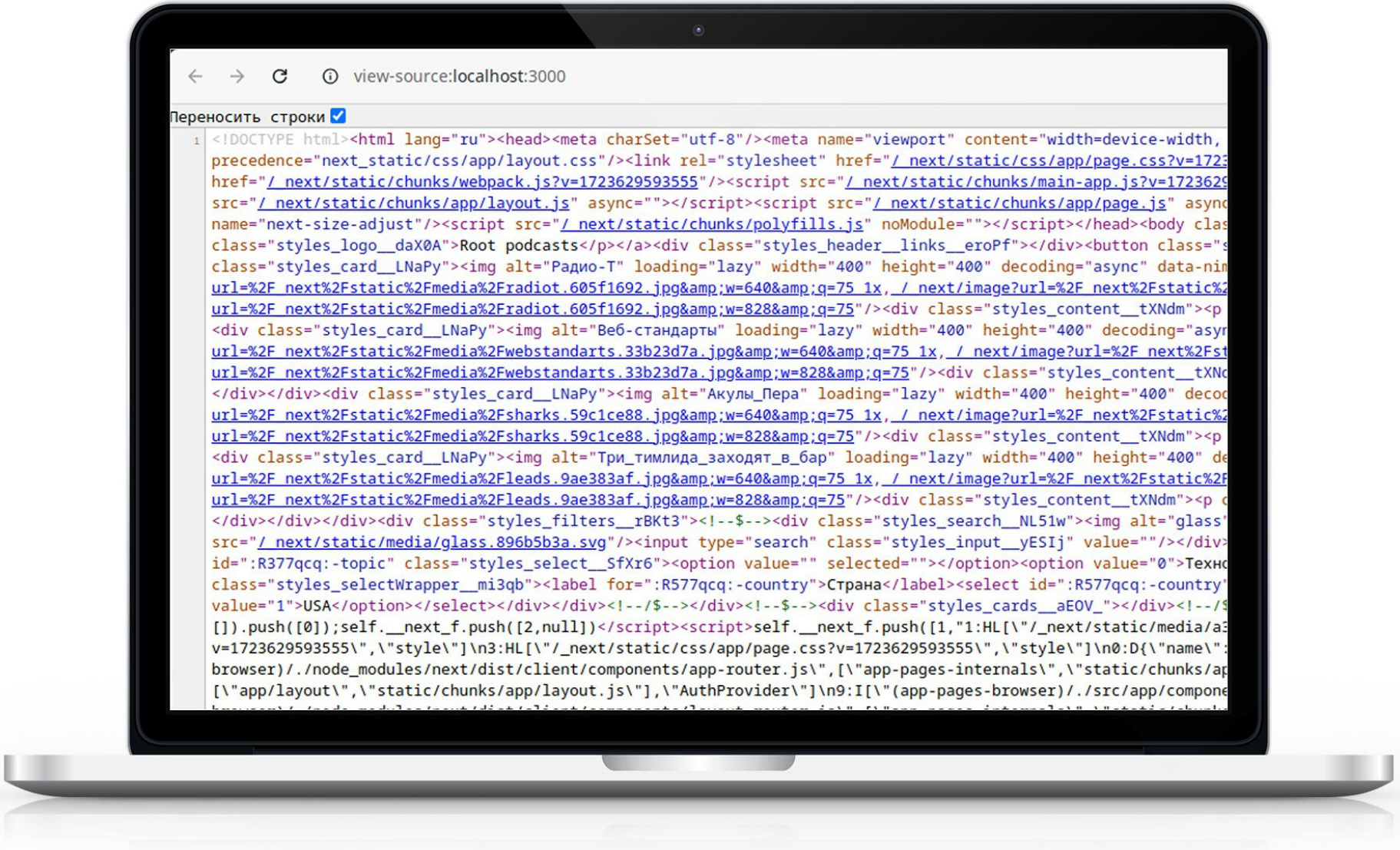
```
async function getFilters() {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts/filters`
  );
  const { countries, topics } = (await res.json()) satisfies FirtersResponse;

  return (
    <div className={styles.filters}>
      <Suspense>
        <Search />
        <div className={styles.selects}>
          <Select searchParam="topic" elements={topics} placeholder="Тема" />
          <Select
            searchParam="country"
            elements={countries}
            placeholder="Страна"
          />
        </div>
      </Suspense>
    </div>
  );
}
```

```
export default async function MainPage() {
  return (
    <main>
      <p>Выбор редакции:</p>
      {await getFilters()}
    </main>
  );
```

Server-Side Rendering



Server-Side Rendering



Сила в объединении

Сила в объединении

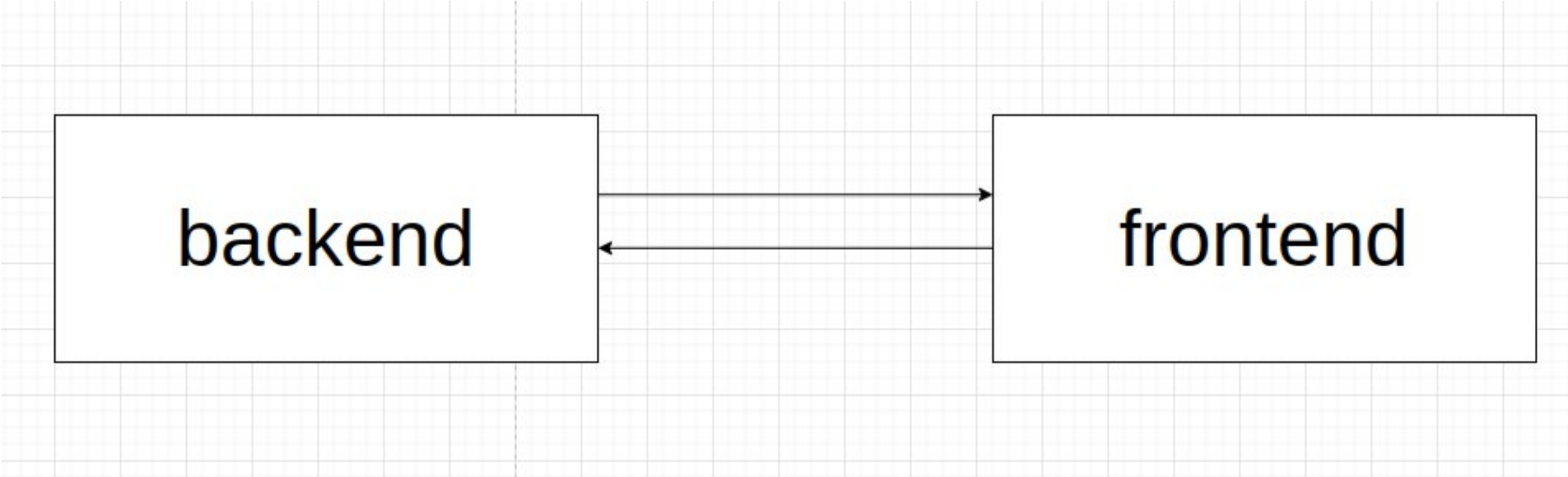
- Динамический контент загружаем на клиенте;

Сила в объединении

- Динамический контент загружаем на клиенте;
- Условно-статический на сервере.

Как получать данные?

Как получать данные?



Как получать данные?

/src/app/page.tsx

```tsx
import { Dictonary } from "@/types";
import styles from "./styles.module.css";

async function getFilters() {
  const [countriesResponse, topicsResponse] = await Promise.all([
    fetch(`${process.env.BASE_BACKEND}/countries`),
    fetch(`${process.env.BASE_BACKEND}/topics`),
  ]);

  const [countries, topics]: [Dictonary[], Dictonary[]] = [
    await countriesResponse.json(),
    await topicsResponse.json(),
  ];

  return <div>{/*отбражаем данные*/}</div>;
}

export default async function MainPage() {
  return <main className={styles.main}>{await getFilters()}</main>;
}

```

Как получать данные?

Как получать данные?


Your tango
Ogres have layers.
Onions have layers. — Shrek

/src/app/api/podcasts/route.ts

```ts
import { Podcast } from "@/types";
import { NextRequest, NextResponse } from "next/server";

export const dynamic = "force-dynamic";

export async function GET(request: NextRequest) {
  const url = new URL(request.url);
  const search = url.searchParams.get("s");

  const baseUrl = `${process.env.BASE_BACKEND}/podcasts`;
  let res;

  if (search) {
    res = await fetch(`${baseUrl}/search?s=${search}`);
  } else {
    res = await fetch(`${baseUrl}?${url.searchParams}`);
  }

  const data = (await res.json()) satisfies Podcast[];

  return NextResponse.json(data, { status: res.status });
}

```

/src/app/api/podcasts/route.ts

```ts
import { Podcast } from "@/types";
import { NextRequest, NextResponse } from "next/server";

export const dynamic = "force-dynamic";

export async function GET(request: NextRequest) {
  const url = new URL(request.url);
  const search = url.searchParams.get("s");

  const baseUrl = `${process.env.BASE_BACKEND}/podcasts`;
  let res;

  if (search) {
    res = await fetch(`${baseUrl}/search?s=${search}`);
  } else {
    res = await fetch(`${baseUrl}?${url.searchParams}`);
  }

  const data = (await res.json()) satisfies Podcast[];

  return NextResponse.json(data, { status: res.status });
}

```

/src/app/api/podcasts/route.ts

```ts
import { Podcast } from "@/types";
import { NextRequest, NextResponse } from "next/server";

export const dynamic = "force-dynamic";

export async function GET(request: NextRequest) {
  const url = new URL(request.url);
  const search = url.searchParams.get("s");

  const baseUrl = `${process.env.BASE_BACKEND}/podcasts`;
  let res;

  if (search) {
    res = await fetch(`${baseUrl}/search?s=${search}`);
  } else {
    res = await fetch(`${baseUrl}?${url.searchParams}`);
  }

  const data = (await res.json()) satisfies Podcast[];

  return NextResponse.json(data, { status: res.status });
}

```

/src/app/page.tsx

```tsx
async function getFilters() {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts/filters`
  );
  const { countries, topics } = (await res.json()) satisfies FirtersResponse;

  return (
    <div className={styles.filters}>
      {/*отображение компонентов*/}
    </div>
  );
}
```

/src/app/page.tsx

```tsx
async function getFilters() {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_APP_BASE_URL}/api/podcasts/filters`
  );
  const { countries, topics } = (await res.json()) satisfies FirtersResponse;

export default async function MainPage() {
  return (
    <main>
      <p>Выбор редакции:</p>
      {await getFilters()}
      <Suspense>
        <CardBlock />
      </Suspense>
    </main>
  );
}
```

backend for frontend

Было

backend for frontend

Было

Стало

backend for frontend

Закрыть доступ к странице

Вот так вот

Page

/src/app/statictics

```tsx
import { cookies } from "next/headers";
import { redirect } from "next/navigation";
import { MAIN_LINK } from "@/constants";

export default function StaticticsPage() {
  const user = cookies().get("user");
  if (!user) {
    redirect(MAIN_LINK);
  }

  return (
    <main>
     //контент страницы
    </main>
  );
}
```

Page

/src/app/statictics

```
1 import { cookies } from "next/headers";
2 import { redirect } from "next/navigation";
3 import { MAIN_LINK } from "@/constants";
4
5 export default function StaticticsPage() {
6   const user = cookies().get("user");
7   if (!user) {
8     redirect(MAIN_LINK);
9   }
10
11   return (
12     <main>
13       //контент страницы
14     </main>
15   );
16 }
```

Page

/src/app/statictics

```tsx
import { cookies } from "next/headers";
import { redirect } from "next/navigation";
import { MAIN_LINK } from "@/constants";

export default function StaticticsPage() {
  const user = cookies().get("user");
  if (!user) {
    redirect(MAIN_LINK);
  }

  return (
    <main>
     //контент страницы
    </main>
  );
}
```

Layout

/src/app/(private)/statictics/layout.tsx

```tsx
import { MAIN_LINK } from "@/constants";
import { cookies } from "next/headers";
import { redirect } from "next/navigation";

export default function Layout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  const user = cookies().get("user");
  if (!user) {
    redirect(MAIN_LINK);
  }

  return children;
}
```

Layout

/src/app/(private)/statictics/layout.tsx

```tsx
import { MAIN_LINK } from "@/constants";
import { cookies } from "next/headers";
import { redirect } from "next/navigation";

export default function Layout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  const user = cookies().get("user");
  if (!user) {
    redirect(MAIN_LINK);
  }

  return children;
}
```

Middleware

/src/middleware.ts

```ts
import { NextRequest, NextResponse } from "next/server";
import { STATICTICS_LINK, MAIN_LINK } from "@/constants";

const isAuth = (request: NextRequest) => request.cookies.has("user");

export async function middleware(request: NextRequest) {
  const pathname = request.nextUrl.pathname;

  if (!isAuth(request) && pathname.startsWith(STATICTICS_LINK)) {
    return NextResponse.redirect(new URL(MAIN_LINK, request.url));
  }
  return NextResponse.next();
}
```

Middleware

/src/middleware.ts

```ts
import { NextRequest, NextResponse } from "next/server";
import { STATICTICS_LINK, MAIN_LINK } from "@/constants";

const isAuth = (request: NextRequest) => request.cookies.has("user");

export async function middleware(request: NextRequest) {
  const pathname = request.nextUrl.pathname;

  if (!isAuth(request) && pathname.startsWith(STATICTICS_LINK)) {
    return NextResponse.redirect(new URL(MAIN_LINK, request.url));
  }
  return NextResponse.next();
}
```

Middleware

/src/middleware.ts

```ts
import { NextRequest, NextResponse } from "next/server";
import { STATICTICS_LINK, MAIN_LINK } from "@/constants";

const isAuth = (request: NextRequest) => request.cookies.has("user");

export async function middleware(request: NextRequest) {
  const pathname = request.nextUrl.pathname;

  if (!isAuth(request) && pathname.startsWith(STATICTICS_LINK)) {
    return NextResponse.redirect(new URL(MAIN_LINK, request.url));
  }
  return NextResponse.next();
}
```

Развитие Next.js

- 13.4 - React Server Components;

Развитие Next.js

- 13.4 - React Server Components;
- 13.5 - Исправление ошибок, увеличение скорости;

Развитие Next.js

- 13.4 - React Server Components;
- 13.5 - Исправление ошибок, увеличение скорости;
- 14.1 - Лучший вывод ошибок, logger;

Развитие Next.js

- 13.4 - React Server Components;
- 13.5 - Исправление ошибок, увеличение скорости;
- 14.1 - Лучший вывод ошибок, logger;
- 14.2 - turbopack In dev, Tree-shaking.

Авторизация

Ресурсы

3 Runtime

3 Runtime

API	Purpose	Where	Status Code
`redirect`	Redirect user after a mutation or event	Server Components, Server Actions, Route Handlers	307 (Temporary) or 303 (Server Action)
`permanentRedirect`	Redirect user after a mutation or event	Server Components, Server Actions, Route Handlers	308 (Permanent)
`useRouter`	Perform a client-side navigation	Event Handlers in Client Components	N/A
`redirects` in `next.config.js`	Redirect an incoming request based on a path	`next.config.js` file	307 (Temporary) or 308 (Permanent)
`NextResponse.redirect`	Redirect an incoming request based on a condition	Middleware	Any

3 Runtime

- document.cookie;

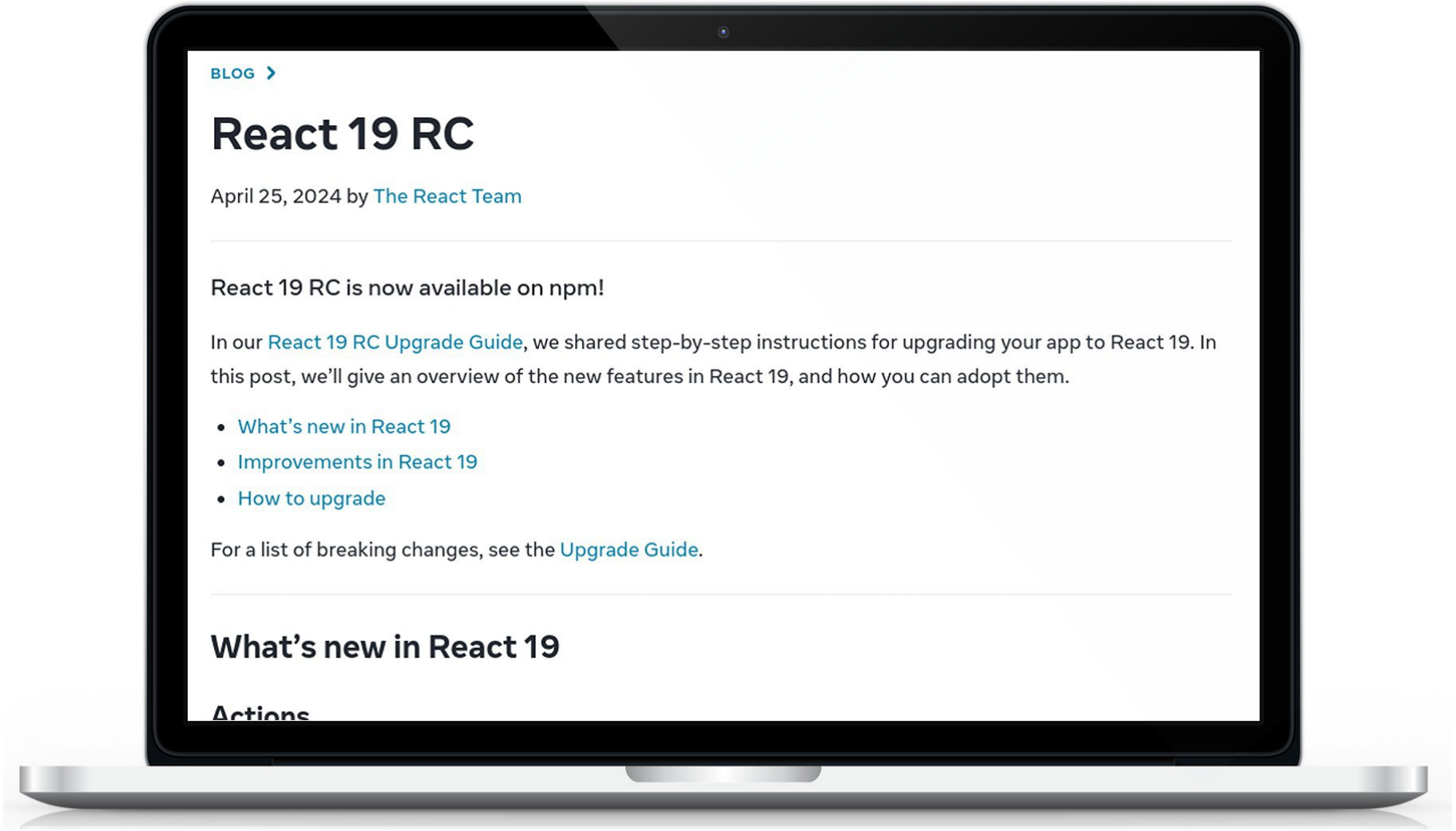
3 Runtime

- document.cookie;
- (res | req).cookies;

3 Runtime

- document.cookie;
- (res | req).cookies;
- cookies() .

Официально React 18



Официально React 18

React 19 RC

April 25, 2024

Компоненты из UI Kit только клиентские

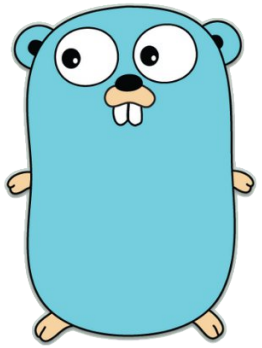

htmx
high power tools for HTML

</>htmx

high power tools for HTML

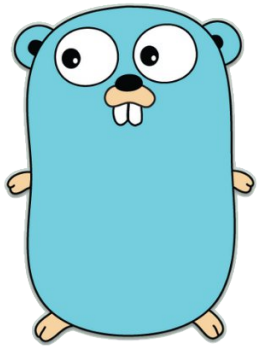
</>htmx

high power tools for HTML



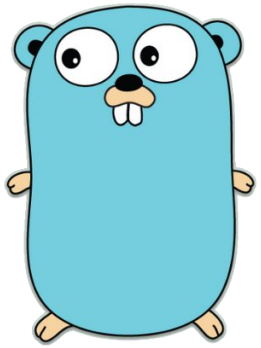
</>htmx

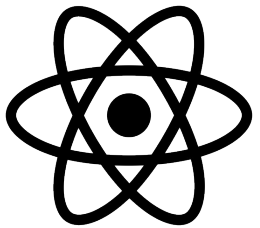
high power tools for HTML



</>htmx

high power tools for HTML

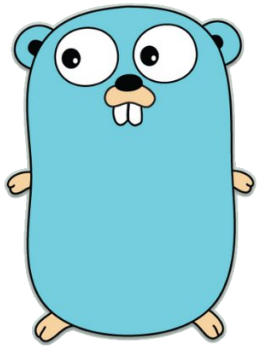




React

</>htmx

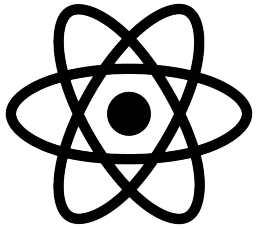
high power tools for HTML



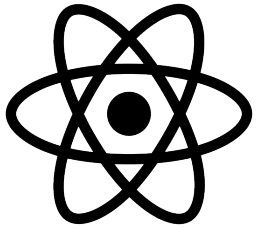
Java

GO

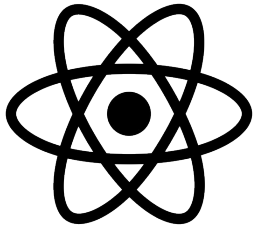
React



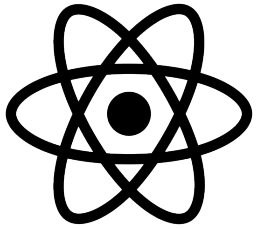
React



React



React

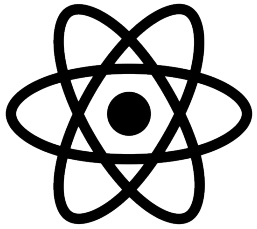


React

NEXT.JS

Waku

Remix

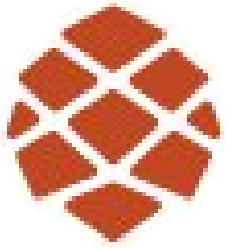


React

NEXT.JS

Waku

Remix



RedwoodJS

Возвращение Fullstack-а

Что сегодня обсудили

- Построили приложение на **Next.js;**

Что сегодня обсудили

- Построили приложение на Next.js;
- Обсудили сложности в работе с Next.js;

Что сегодня обсудили

- Построили приложение на Next.js;
- Обсудили сложности в работе с Next.js;
- Предположили куда будет двигаться индустрия.

Материалы