

React 개발이 이렇게 쉬웠나? (Feat. `Next.js`)

이관형 NAVER 쇼핑검색서비스개발



CONTENTS

1. UI Library React
2. React Framework Next.js
3. Troubleshooting & Tips

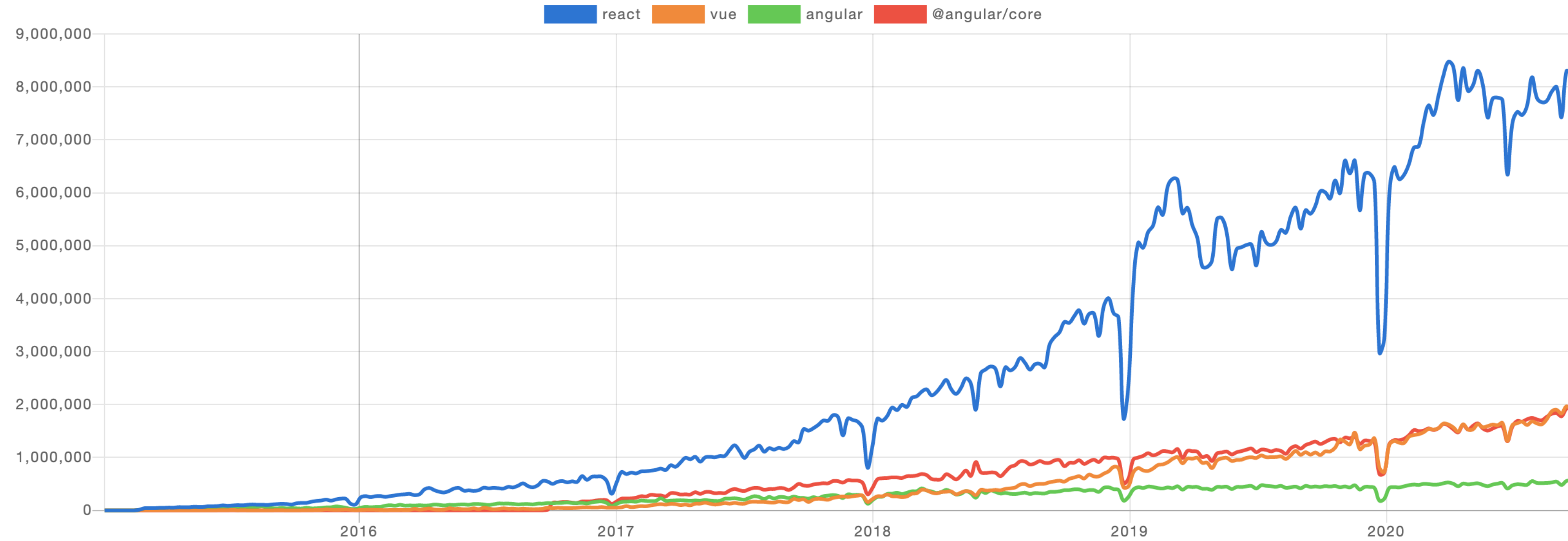
• **01 UI Library React**

02

1.1 커져가는 React 생태계

react x vue x angular x @angular/core x + ember-source

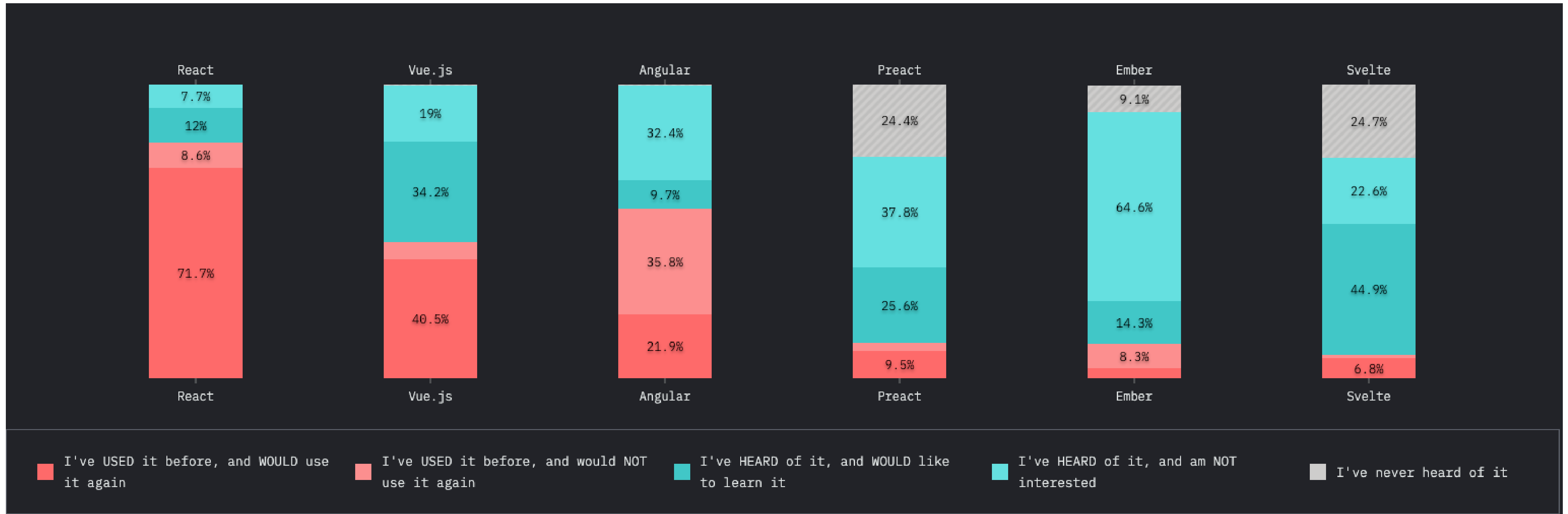
Downloads in past All time v



Stats

	stars 🌟	issues ⚠️	updated 🛠️	created 🗓️	size 📦
react	156,836	582	Oct 4, 2020	May 25, 2013	minzipped size 2.6 KB
vue	173,343	521	Oct 4, 2020	Jul 29, 2013	minzipped size 22.9 KB
angular	59,496	464	Oct 2, 2020	Jan 6, 2010	minzipped size 62.2 KB
@angular/core	66,478	3,243	Oct 4, 2020	Sep 19, 2014	minzipped size 87.2 KB

1.1 커져가는 React 생태계



1.2 React의 한계

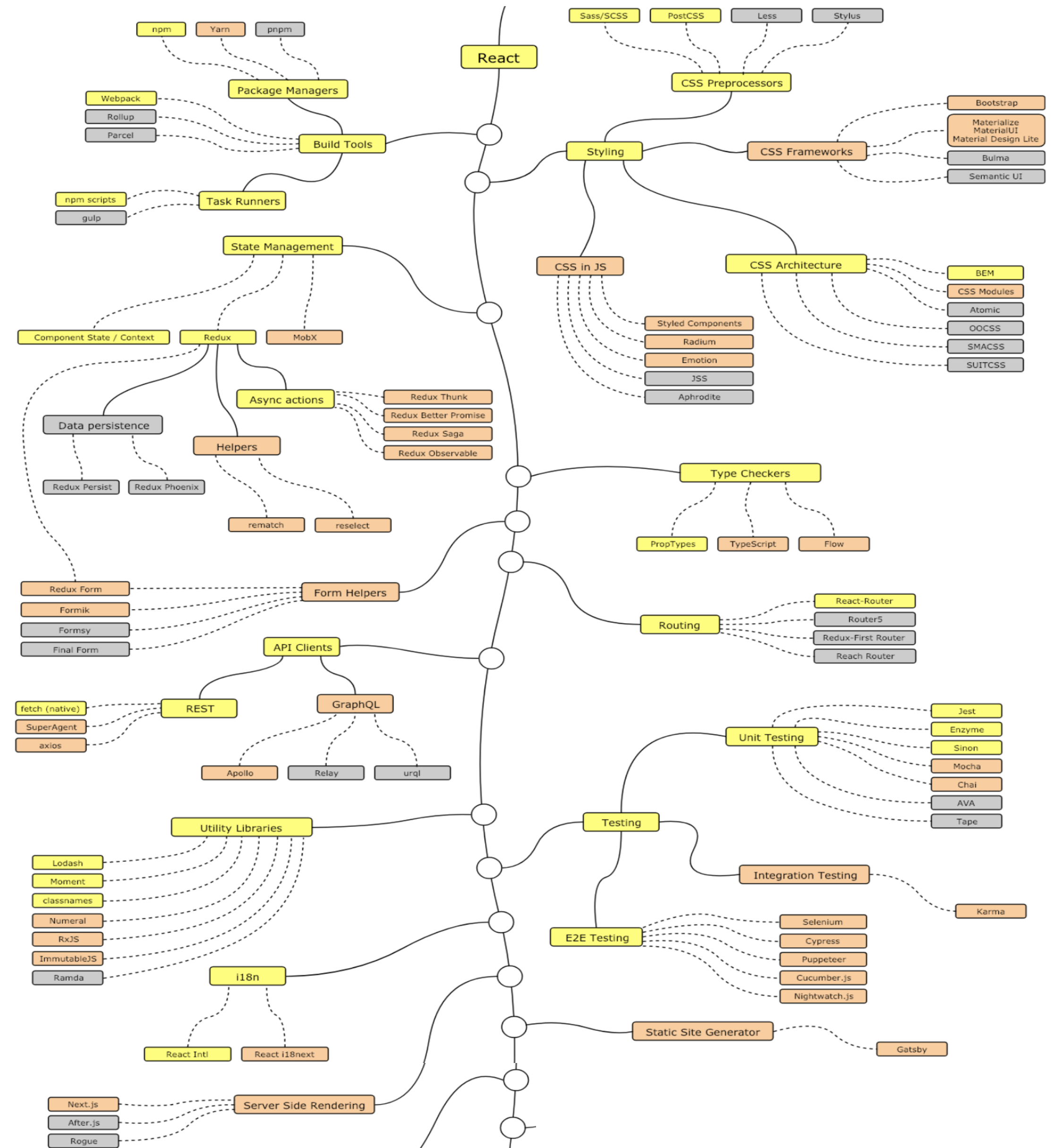
FOCUSES ON ONE THING (The view layer)

Feature	Angular	React	Vue
UI / DOM Manipulation	✓	✓	✓
State Management	✓	✓	✓
Routing	✓	✗	✓
Form Validation & Handling	✓	✗	✗
Http Client	✓	✗	✗

1.2 React의 한계

프로젝트 구성을 위한 기능들

- SSR/SSG
- Type checker
- Routing
- State management
- Styling
- Bundling
- Lazy loading
- Code splitting
- Static file serve
- Scroll Restoration



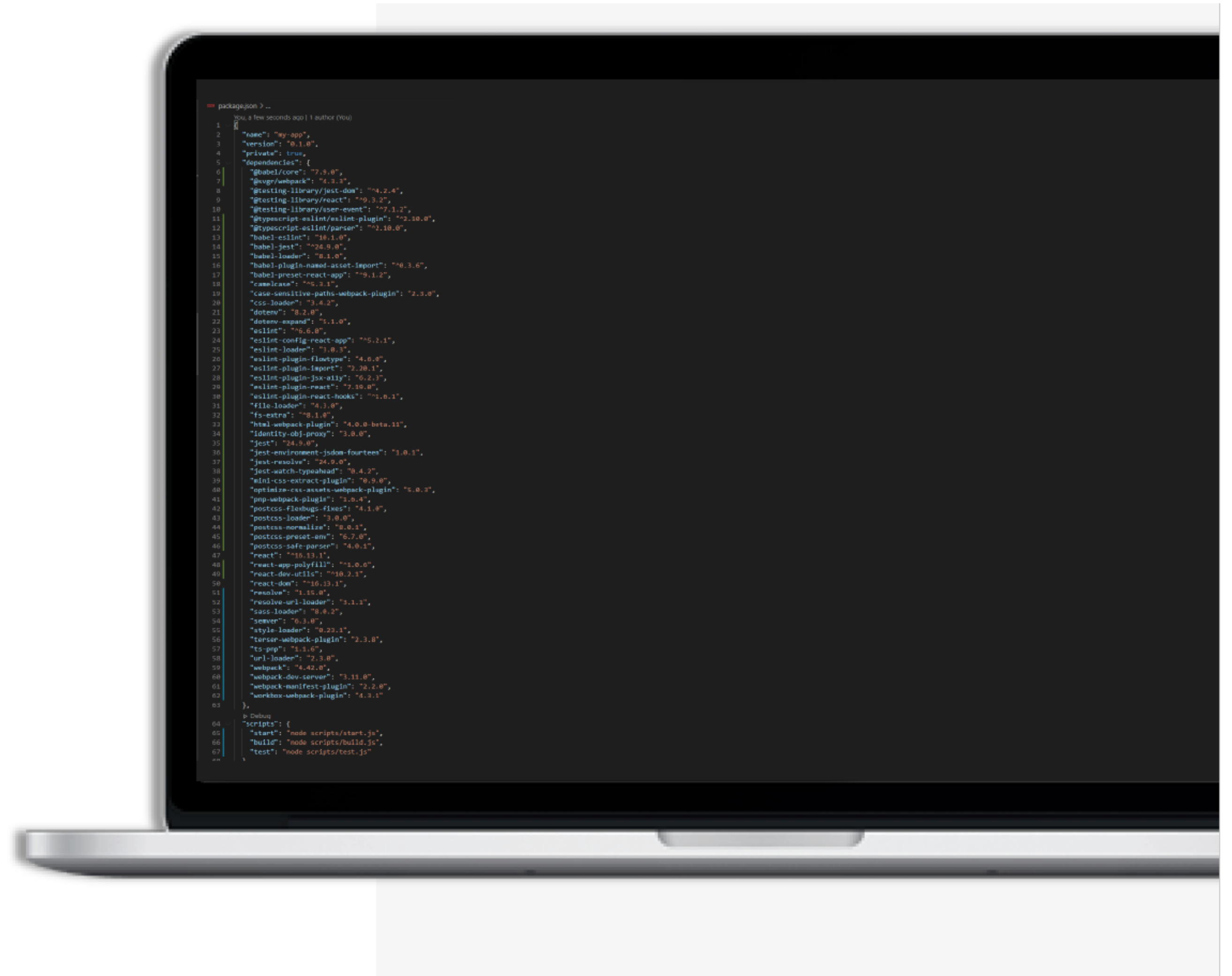
1.2 React의 한계

리액트 빌드 툴체인 마스터

2020/09 기준, 플러그인

Webpack

- plugin : 5924개
- loader : 4799개
- Babel : 4931개



1.2 React의 한계

리액트 빌드 툴체인 마스터

2020/09 기준, 플러그인

Webpack

- plugin : 5924개
- loader : 4799개
- Babel : 4931개

지속적복잡도증가





• **02** **React Framework**
Next.js

2.1 Web Framework가 필요한 이유

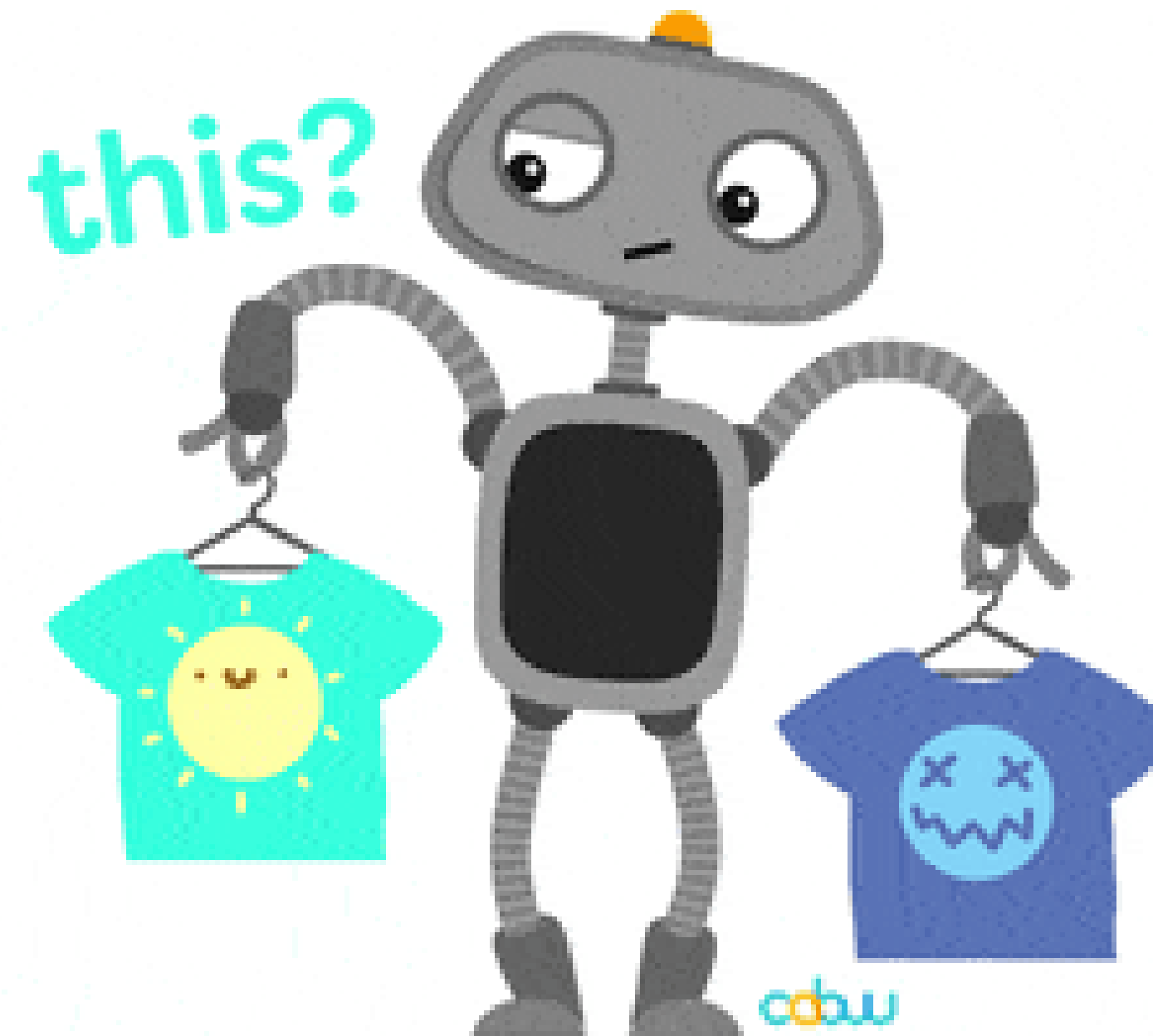
User Experience

What are sites struggling with?

vs.

Developer Experience

What are developers struggling with?



2.1 Web Framework가 필요한 이유

사용자 경험 > 개발자 경험 > 구현 용이성

2.1 Web Framework가 필요한 이유

사용자 경험 > 개발자 경험 > 구현 용이성

개발자 경험 ↑

ex) 빌드 시간 개선, HMR 제공



사용자 경험 ↑

ex) 버그 감소, 개발시간 증가

2.1 Web Framework가 필요한 이유

사용자 경험 > 개발자 경험 > 구현 용이성

개발자 경험 ↓

ex) SSR, lazy loading,
code splitting



사용자 경험 ↑

ex) 성능 향상

2.1 Web Framework가 필요한 이유

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

2.2 Next.js vs Gatsby vs Create React App



Next.js



Gatsby



Create React App



- 페이지마다 data-fetch, pre rendering 전략을 다르게 가능
- 증분 정적 페이지 생성
- 다양한 상태관리, 스타일링 라이브러리 지원
- 파일 시스템 라우팅
- API route
- Webpack 5 지원

- GraphQL
- 플러그인 생태계
- 이미지 최적화
- 정적 페이지 성능이 next.js 보다 약간 좋음

- 가장 배우기 쉬움 (추가 프레임 워크 없음)
- React 팀에서 생성 및 유지 관리



- 프레임 워크 지식 필요
- (_app.js, _document.js, getServerSideProps)
- 플러그인 시스템 없음(todo)
- 이미지 최적화(todo)

- GraphQL
- 빌드시간이 오래걸림 (이미지 최적화, 정적 페이지)

- 서버 측 렌더링 없음(라우팅 라이브러리 필요)
- Code Splitting과 같은 기능 직접 구현
- eject 시 복잡성이 증가함

2.2 Next.js vs Gatsby vs Create React App

NEXT.js

Next.js



- 페이지마다 data-fetch, pre rendering 전략을 다르게 가능
- 증분 정적 페이지 생성
- 다양한 상태관리, 스타일링 라이브러리 지원
- 파일 시스템 라우팅
- API route
- Webpack 5 지원



- 프레임 워크 지식 필요
- (_app.js, _document.js, getServerSideProps)
- 플러그인 시스템 없음(todo)
- 이미지 최적화(todo)



Gatsby

- GraphQL
- 플러그인 생태계
- 이미지 최적화
- 정적 페이지 성능이 next.js 보다 약간 좋음

- GraphQL
- 빌드시간이 오래걸림 (이미지 최적화, 정적 페이지)



Create React App

- 가장 배우기 쉬움 (추가 프레임 워크 없음)
- React 팀에서 생성 및 유지 관리

- 서버 측 렌더링 없음(라우팅 라이브러리 필요)
- Code Splitting과 같은 기능 직접 구현
- eject 시 복잡성이 증가함

2.2 Next.js vs Gatsby vs Create React App



Next.js



- 페이지마다 data-fetch, pre rendering 전략을 다르게 가능
- 증분 정적 페이지 생성
- 다양한 상태관리, 스타일링 라이브러리 지원
- 파일 시스템 라우팅
- API route
- Webpack 5 지원



- 프레임 워크 지식 필요
- (_app.js, _document.js, getServerSideProps)
- 플러그인 시스템 없음(todo)
- 이미지 최적화(todo)



Gatsby

- GraphQL
- 플러그인 생태계
- 이미지 최적화
- 정적 페이지 성능이 next.js 보다 약간 좋음

- GraphQL
- 빌드시간이 오래걸림 (이미지 최적화, 정적 페이지)



Create React App

- 가장 배우기 쉬움 (추가 프레임 워크 없음)
- React 팀에서 생성 및 유지 관리

- 서버 측 렌더링 없음(라우팅 라이브러리 필요)
- Code Splitting과 같은 기능 직접 구현
- eject 시 복잡성이 증가함

2.2 Next.js vs Gatsby vs Create React App



Next.js



- 페이지마다 data-fetch, pre rendering 전략을 다르게 가능
- 증분 정적 페이지 생성
- 다양한 상태관리, 스타일링 라이브러리 지원
- 파일 시스템 라우팅
- API route
- Webpack 5 지원



- 프레임 워크 지식 필요
- (_app.js, _document.js, getServerSideProps)
- 플러그인 시스템 없음(todo)
- 이미지 최적화(todo)



Gatsby

- GraphQL
- 플러그인 생태계
- 이미지 최적화
- 정적 페이지 성능이 next.js 보다 약간 좋음

- GraphQL
- 빌드시간이 오래걸림 (이미지 최적화, 정적 페이지)

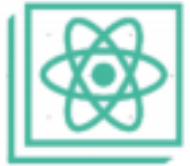


Create React App

- 가장 배우기 쉬움 (추가 프레임 워크 없음)
- React 팀에서 생성 및 유지 관리

- 서버 측 렌더링 없음(라우팅 라이브러리 필요)
- Code Splitting과 같은 기능 직접 구현
- eject 시 복잡성이 증가함

2.2 Next.js vs Gatsby vs Create React App



Create React App

Create React App is a new officially supported way to create **single-page** React applications. It offers a modern build setup with no configuration.



Gatsby.js Documentation

Gatsby is a blazing fast modern **site generator** for React.

The logo for Next.js, featuring the text 'NEXT.js' in a stylized font with a diagonal slash through the 'X' and a small '.js' suffix.

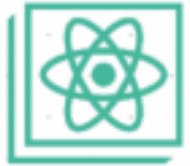
NEXT.js

Next.js

Next.js gives you the best developer experience with all the features you need for production :

hybrid static & server rendering, TypeScript support, smart bundling, route pre-fetching, and more. No config needed.

2.2 Next.js vs Gatsby vs Create React App



Create React App

Create React App is a new officially supported way to create **single-page** React applications. It offers a modern build setup with no configuration.



Gatsby.js Documentation

Gatsby is a blazing fast modern **site generator** for React.



Next.js

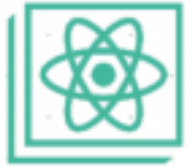
Next.js gives you the best developer experience with all the features you need for production :
hybrid static & server rendering, TypeScript support, smart bundling, route pre-fetching, and more. No config needed.

🔗 추천 틀체인

React 팀의 추천 방법은 아래와 같습니다

- **React**를 배우고 있거나 아니면 새로운 **싱글 페이지** 앱을 만들고 싶다면 **Create React App**.
- 서버 렌더링 **Node.js** 웹사이트를 만들고 있다면 **Next.js**을 시도해보세요..
- 고정적인 콘텐츠 지향적 웹사이트를 만들고 있다면 **Gatsby**를 시도해보세요..
- 컴포넌트 라이브러리 혹은 이미 있는 코드 베이스에 통합을 한다면 **더 유연한 틀체인**.

2.2 Next.js vs Gatsby vs Create React App



Create React App

Create React App is a new officially supported way to create **single-page** React applications. It offers a modern build setup with no configuration.



Gatsby.js Documentation

Gatsby is a blazing fast modern **site generator** for React.



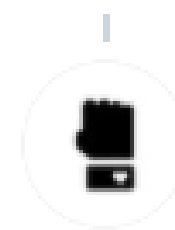
Next.js

Next.js gives you the best developer experience with all the features you need for production :
hybrid static & server rendering, TypeScript support, smart bundling, route pre-fetching, and more. No config needed.

추천 틀체인

React 팀의 추천 방법은 아래와 같습니다

- **React**를 배우고 있거나 아니면 새로운 **싱글 페이지** 앱을 만들고 싶다면 [Create React App](#).
- 서버 렌더링 **Node.js** 웹사이트를 만들고 있다면 [Next.js](#)을 시도해보세요..
- 고정적인 콘텐츠 지향적 웹사이트를 만들고 있다면 [Gatsby](#)를 시도해보세요..
- 컴포넌트 라이브러리 혹은 이미 있는 코드 베이스에 통합을 한다면 [더 유연한 틀체인](#).



Dan Abramov @dan_abramov · 4월 20일

The progression between "static" and "server rendered" and "client rendered" is gradual. You usually want a mix of all three, depending on how fast the data changes and what latency you can afford. But our tools don't treat it as continuum. This is what we need to fix.

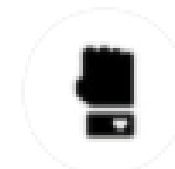
2

1

11



답글



Dan Abramov @dan_abramov · 4월 20일

[@dan_abramov](#) 님과 [@sitnikcode](#) 님에게 보내는 답글

Next.js is moving in the right direction, blending all three modes so that you can switch between them seamlessly.

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

Pre-rendering HTML

How do I choose how these pages are rendered?

	Server	Browser			
	Server Rendering	"Static SSR"	SSR with (Re)hydration	CSR with Prerendering	Full CSR
Overview:	An application where input is navigation requests and the output is HTML in response to them.	Built as a Single Page App, but all pages prerendered to static HTML as a build step, and the JS is removed .	Built as a Single Page App. The server prerenders pages, but the full app is also booted on the client.	A Single Page App, where the initial shell/skeleton is prerendered to static HTML at build time.	A Single Page App. All logic, rendering and booting is done on the client. HTML is essentially just script & style tags.
Authoring:	Entirely server-side (request, response, HTML)	Built as if client-side (components, DOM*, fetch)	Built as client-side	Client-side	Client-side
Rendering:	Dynamic HTML	Static HTML	Dynamic HTML and JS/DOM	Partial static HTML, then JS/DOM	Entirely JS/DOM
Server role:	Controls all aspects. (this client)	Delivers static HTML	Renders pages (navigation requests)	Delivers static HTML	Delivers static HTML
Pros:	👍 TTI = FCP 👍 Fully streaming	👍 Fast TTFB 👍 TTI = FCP 👍 Fully streaming	👍 Flexible	👍 Flexible 👍 Fast TTFB	👍 Flexible 👍 Fast TTFB
Cons:	👎 Slow TTFB 👎 Inflexible	👎 Inflexible 👎 Leads to hydration	👎 Slow TTFB 👎 TTI >>> FCP 👎 Usually buffered	👎 TTI > FCP 👎 Limited streaming	👎 TTI >>> FCP 👎 No streaming
Scales via:	Infra size / cost	build/deploy size	Infra size & JS size	JS size	JS size
Examples:	Gmail HTML, Hacker News	DocuSaurus, Netflix*	Next.js , Razzle , etc	Gatsby, Vuepress, etc	Most apps

developers.google.com/web/updates/2019/02/rendering-on-the-web

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

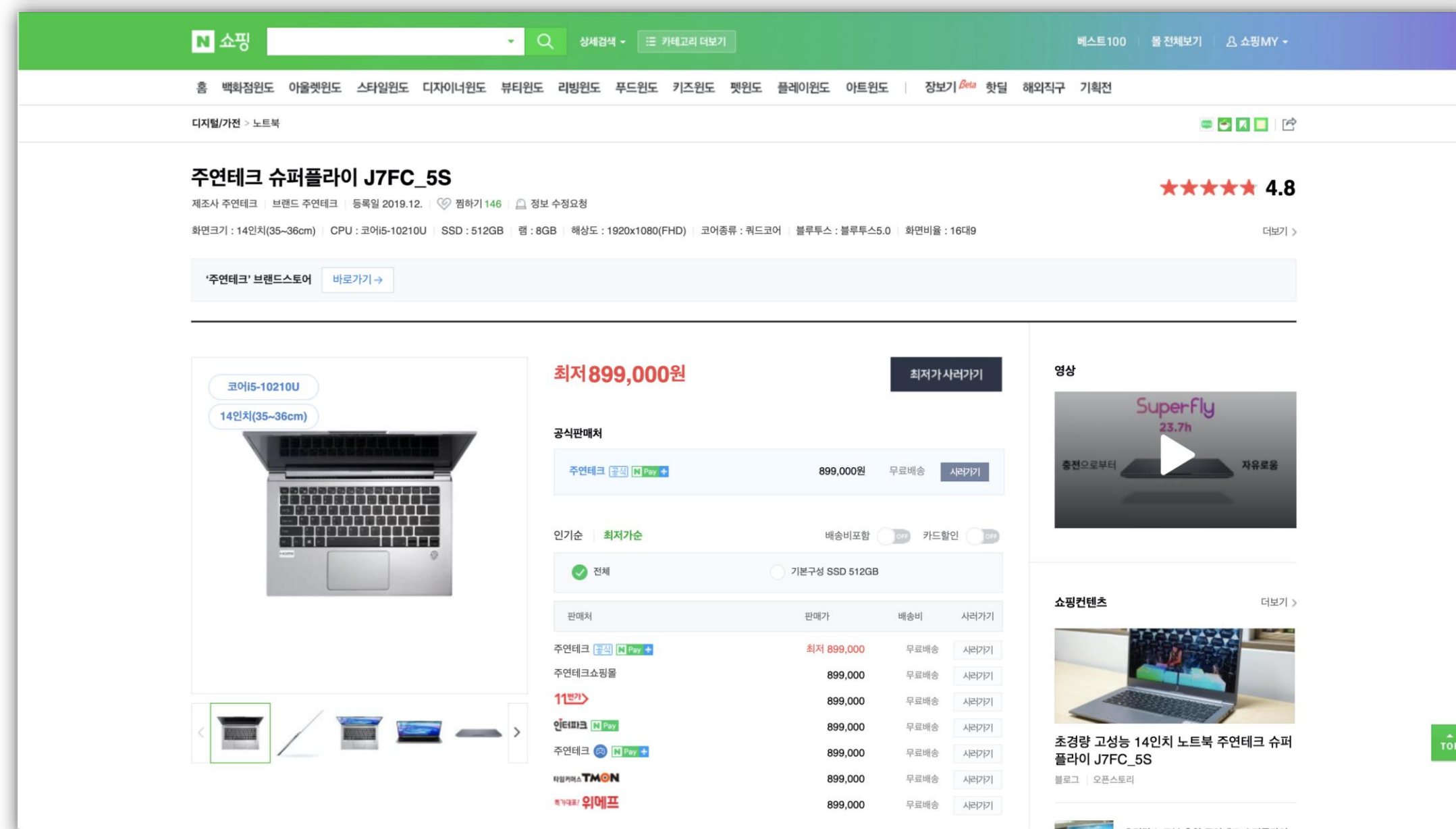
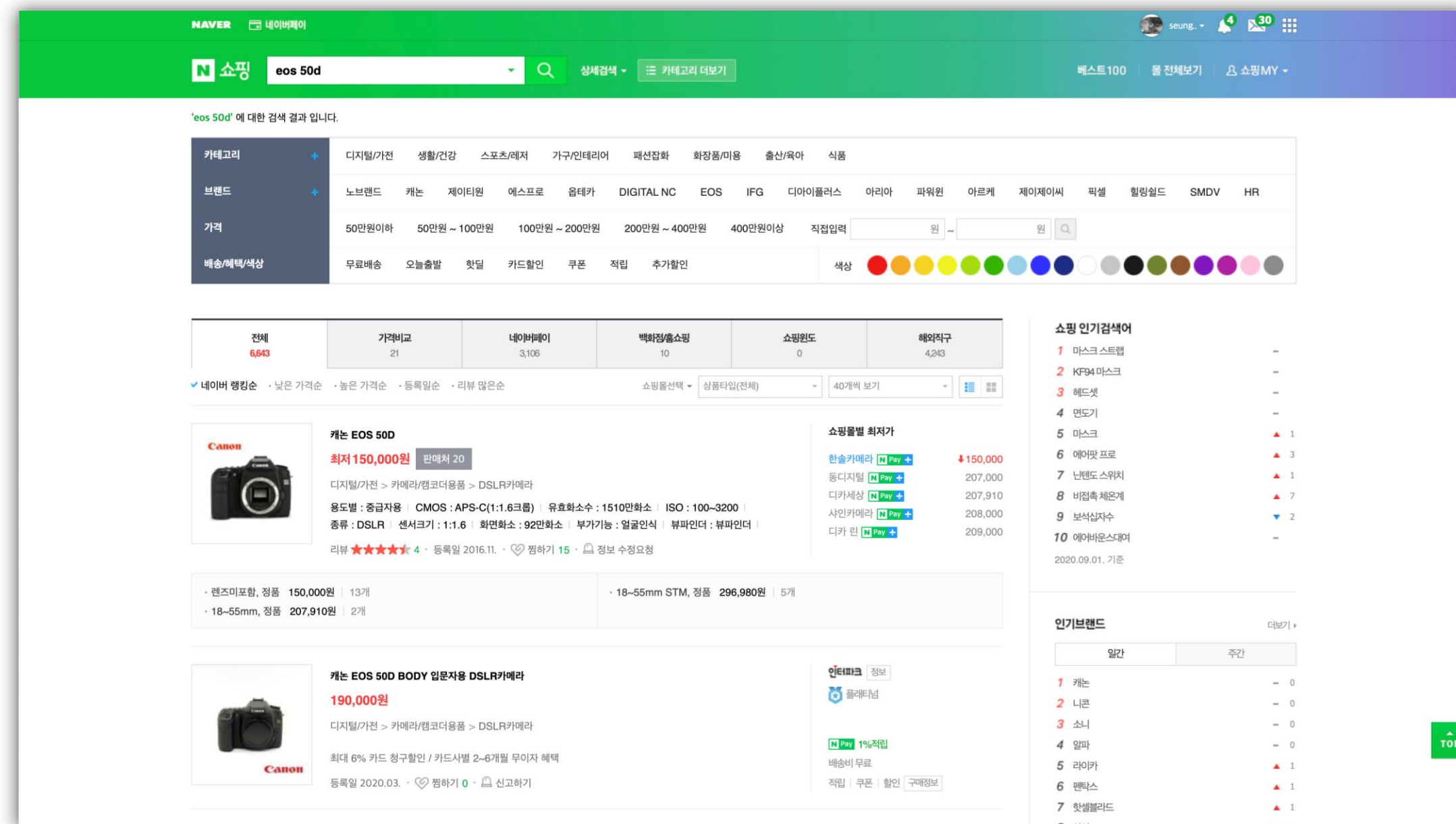
Compilation

Security

bundling

Backend / Server logic

And more



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

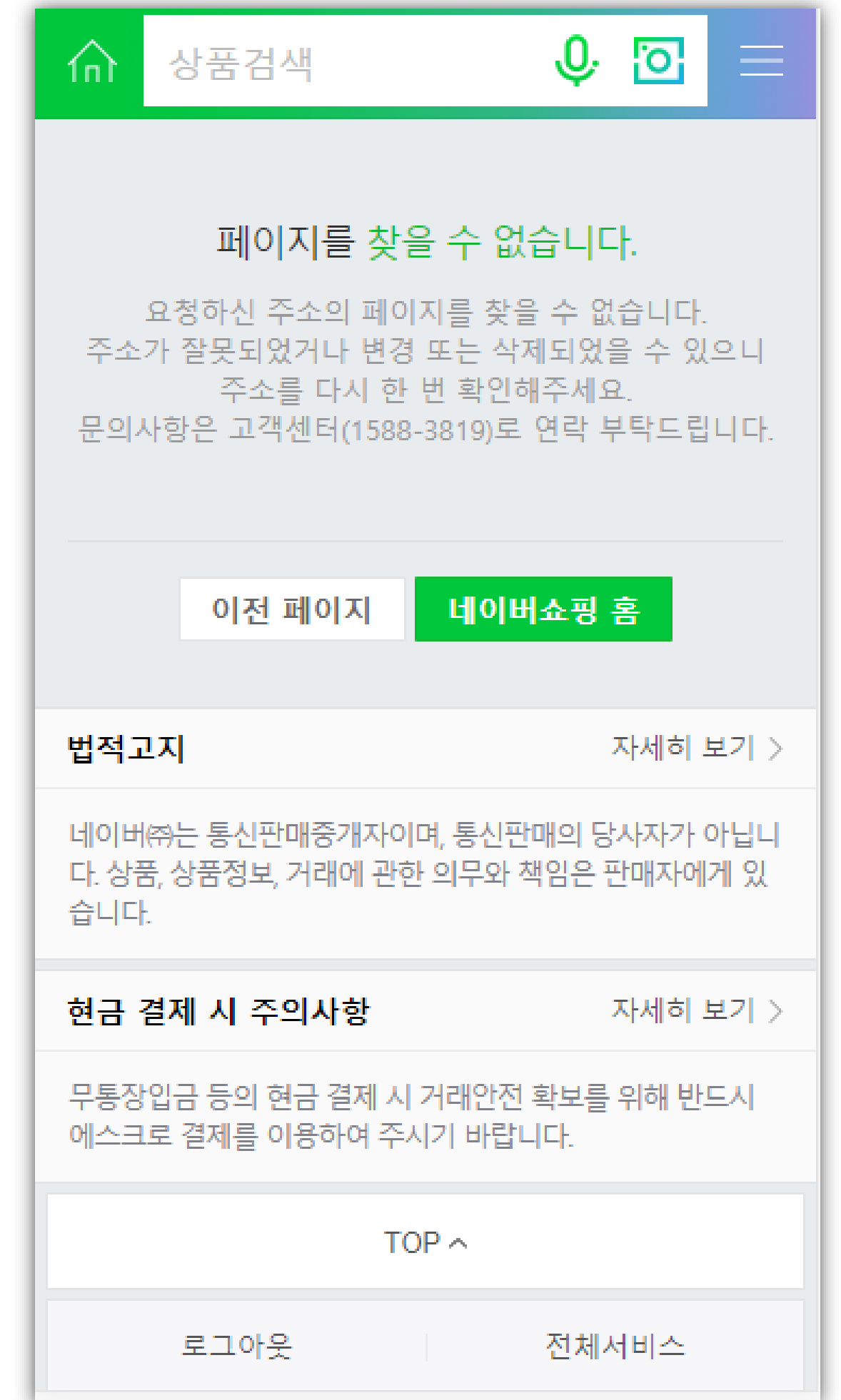
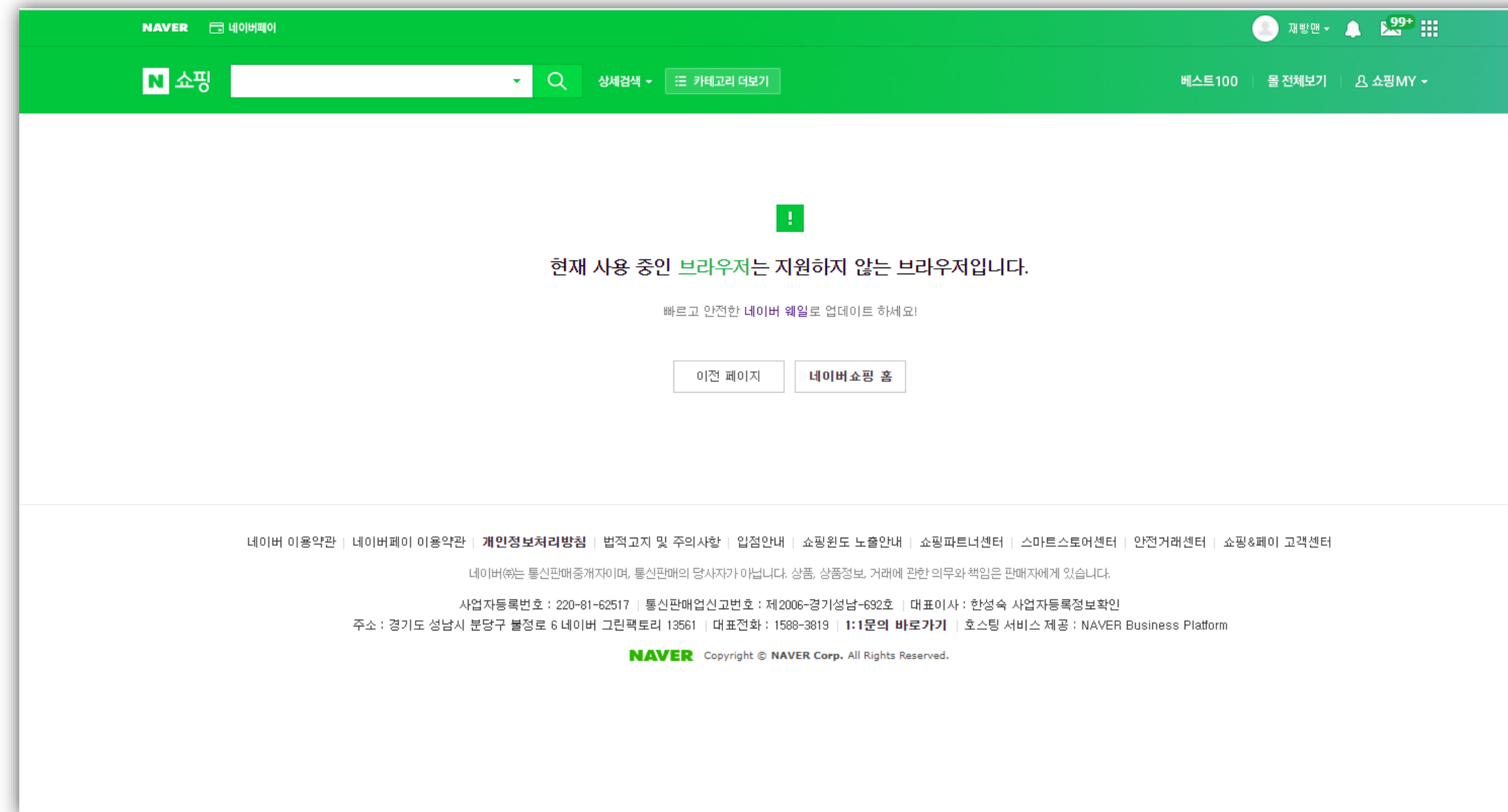
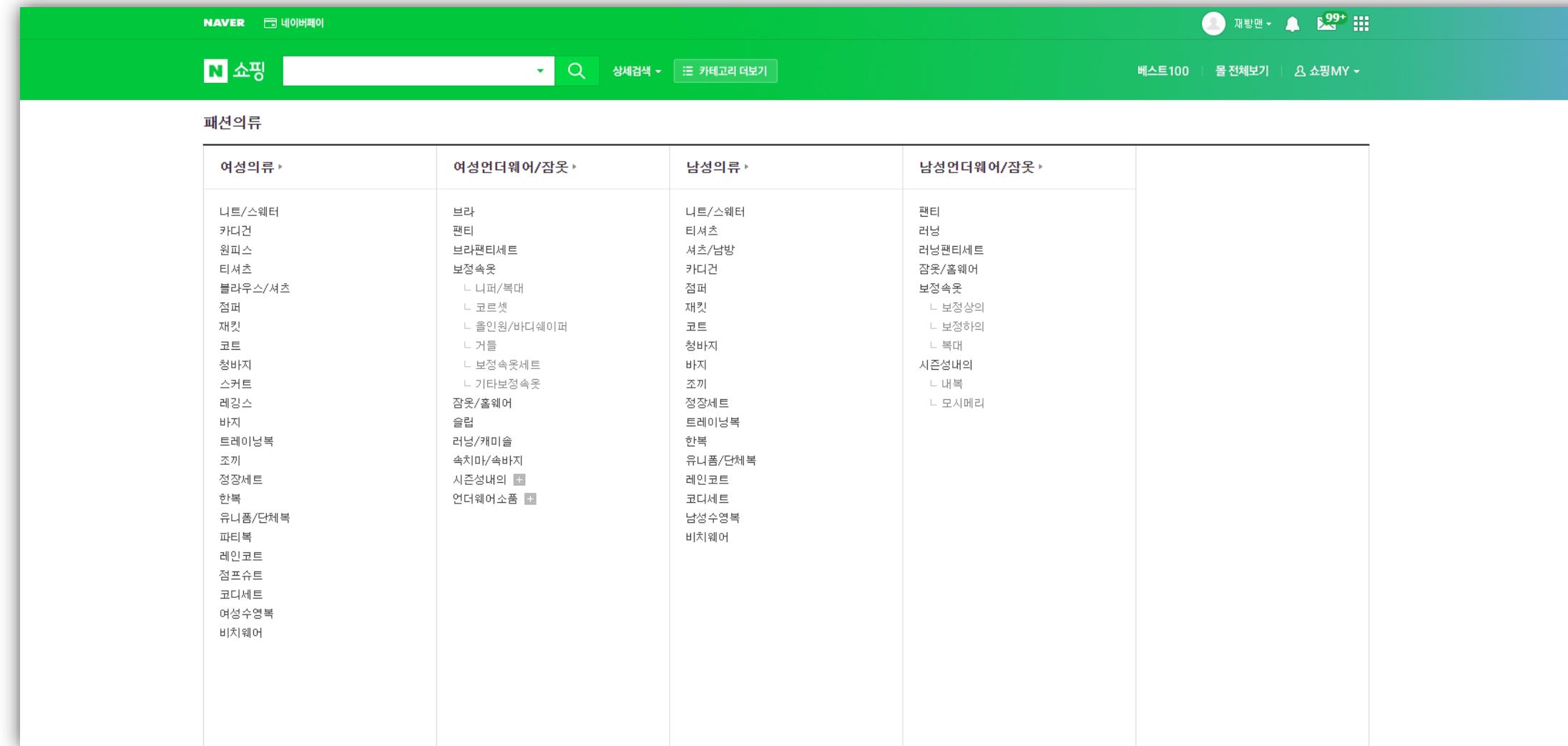
Compilation

Security

bundling

Backend / Server logic

And more



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

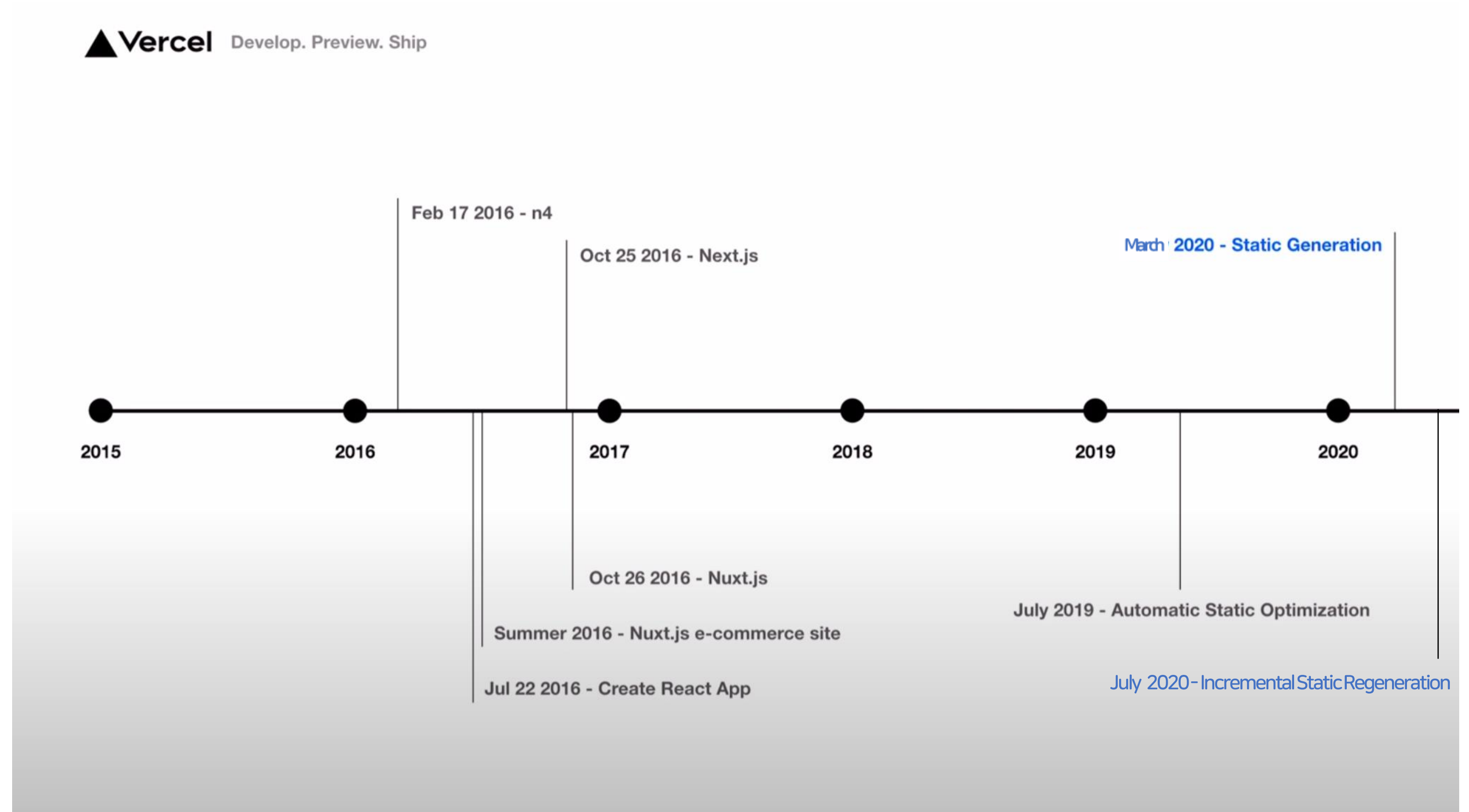
Security

bundling

Backend / Server logic

And more

Pre-rendering HTML



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

Pre-rendering HTML

SSR

server-side rendering

On-demand
Requires runtime execution
JS, CSS is prefetched

vs

SSG

static site generation

At build Time
HTML Can be served from a CDN
JS, CSS and Data is prefetched

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

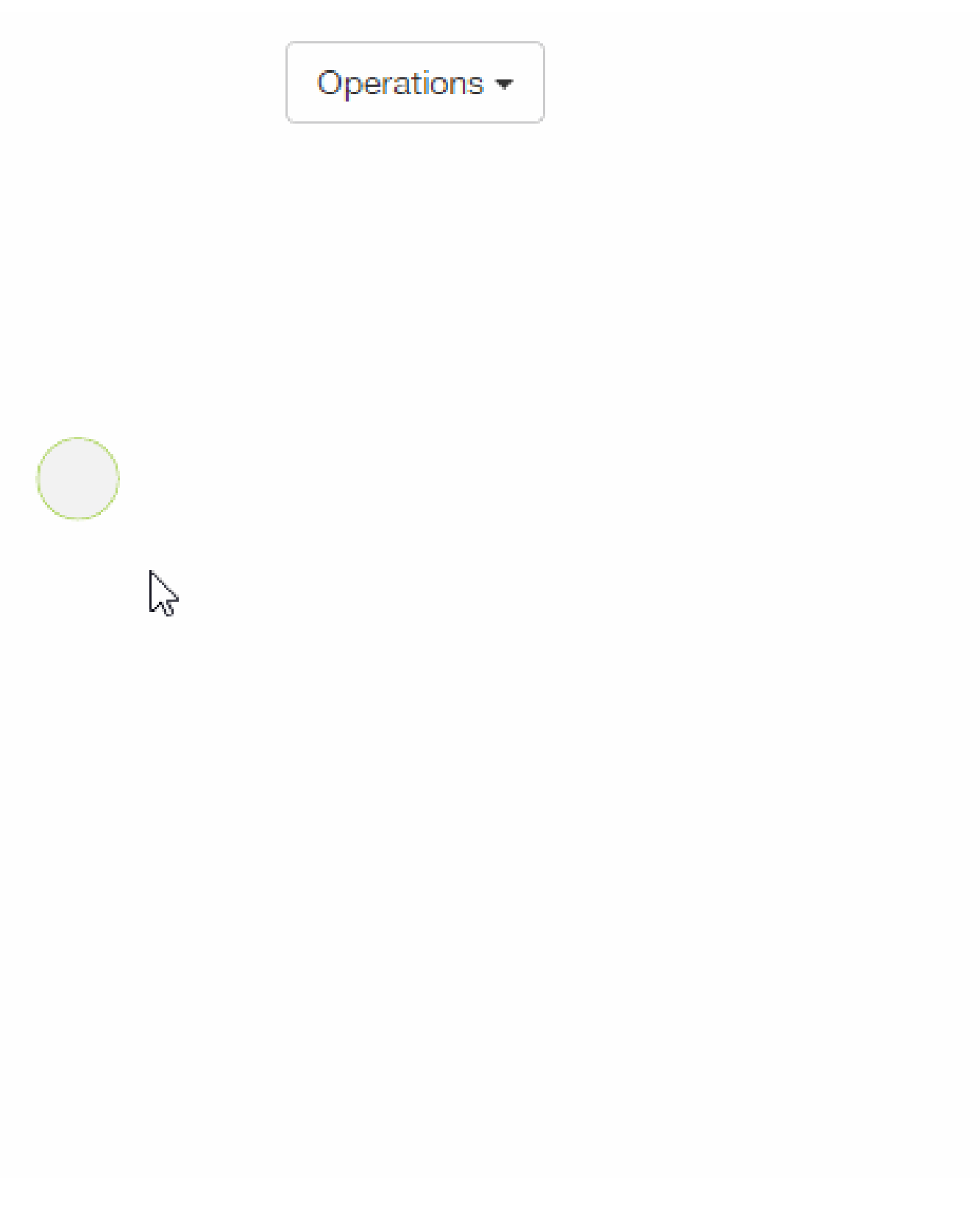
Prefetch

- viewport에 나타나는 링크만 prefetch (Intersection Observer API)
- <link rel="preload">를 동적으로 주입
- 네트워크 속도가 느리거나 Save-data가 on일 경우 disable
- ex) Link prefetch `default : true`

```
<Link href="/search/category?catId=50000167" prefetch>  
  <a>Blog Post</a>  
</Link>
```

- ex) router method

```
const router = useRouter()  
router.prefetch(`/catalog/[nvMid]`, `/catalog/${modelProduct.adId || modelProduct.id}`)
```



jQuery plugin that predicts the mouse trajectory

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

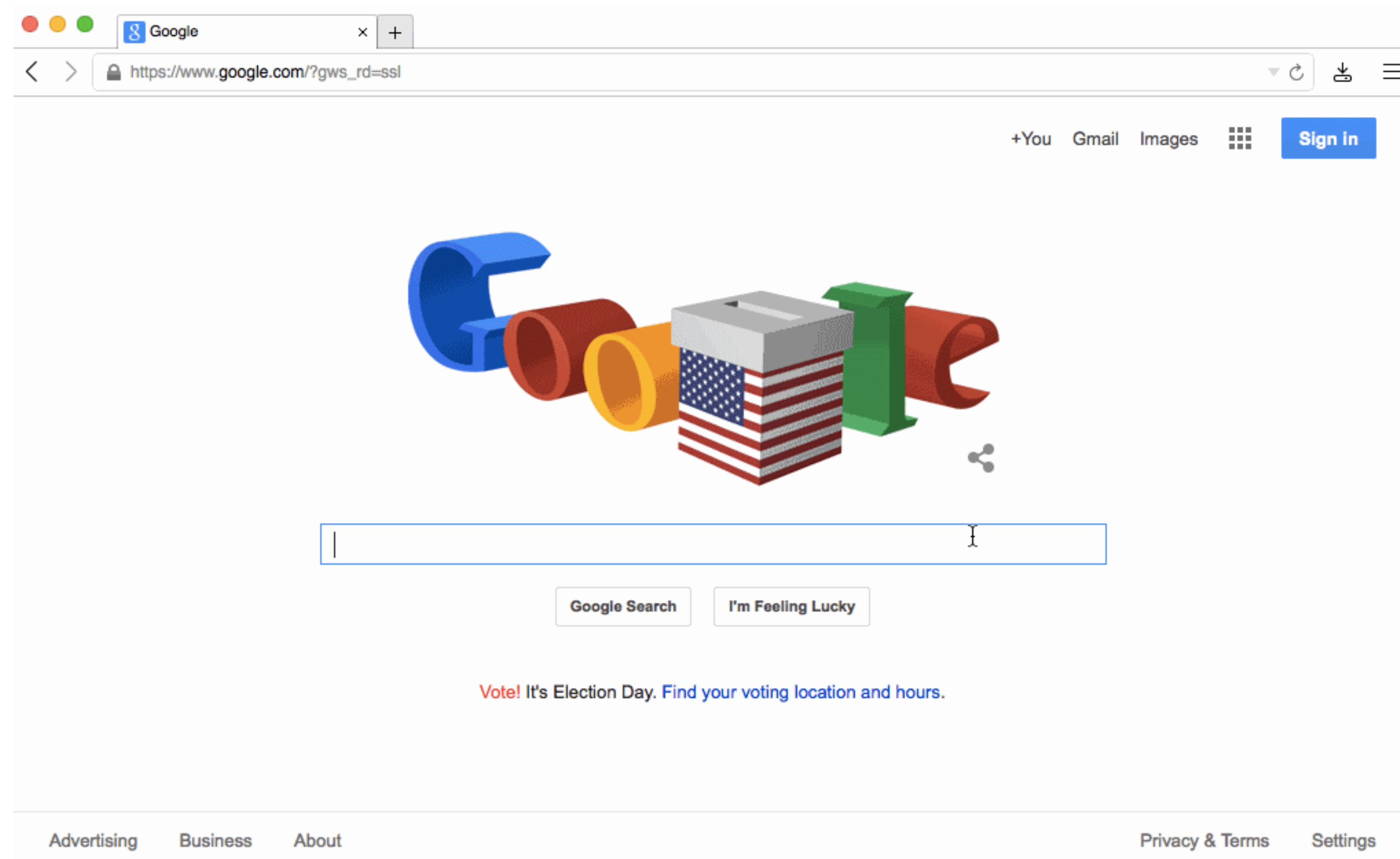
Security

bundling

Backend / Server logic

And more

Prefetch example



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

getStaticProps, getStaticPaths (SSG)

- CDN에 캐시할수있는 HTML, JSON 생성
- 파일시스템에서 읽기 가능 process.cwd()
- 빌드 시간에 실행
- 서버 측 번들

`getStaticProps` (Static Generation)

If you export an `async` function called `getStaticProps` from a page, Next.js will pre-render this page at build time using the props returned by `getStaticProps`.

```
export async function getStaticProps(context) {  
  return {  
    props: {}, // will be passed to the page component as props  
  }  
}
```

The `context` parameter is an object containing the following keys:

- `params` contains the route parameters for pages using dynamic routes. For example, if the page name is `[id].js`, then `params` will look like `{ id: ... }`. To learn more, take a look at the [Dynamic Routing documentation](#). You should use this together with `getStaticPaths`, which we'll explain later.
- `preview` is `true` if the page is in the preview mode and `undefined` otherwise. See the [Preview Mode documentation](#).
- `previewData` contains the preview data set by `setPreviewData`. See the [Preview Mode documentation](#).

`getStaticProps` should return an object with:

- `props` - A **required** object with the props that will be received by the page component. It should be a [serializable object](#)
- `revalidate` - An **optional** amount in seconds after which a page re-generation can occur. More on [Incremental Static Regeneration](#)

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

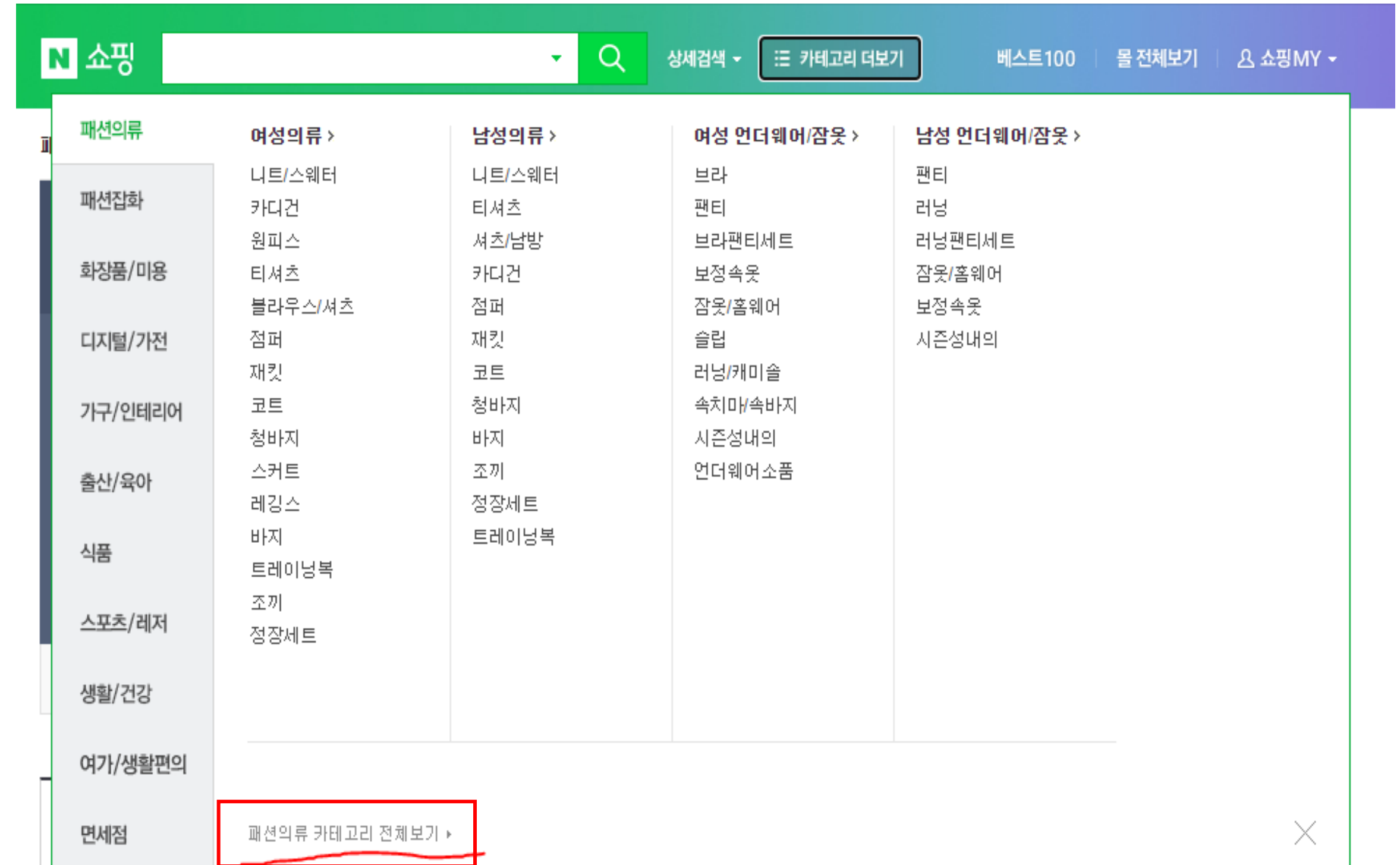
Security

bundling

Backend / Server logic

And more

getStaticProps (SSG)



Name	Status	Type	Initiator	Size	Time	Waterfall
669e31a8db87a5774a8c.css	200	stylesheet	webpack-bf64fff...js:1	1.8 kB	7 ms	
42.64558d73777470a660b4.js	200	script	webpack-bf64fff...js:1	13.4 kB	7 ms	
a	200	text/plain	57e6633...684e101...js:1	348 B	10 ms	
cc7a=SNB.category&bw=1064&px=654&py=62&sx=654&sy=6...EC%96%B4%ED%8C%9F&...	200	gif	57e6633...684e101...js:1	410 B	17 ms	
click7a=SNB.category&&ssc=svc.shopping.v3&&ts=1602...=50000209&&c3=50001406&&...	200	octet-stream	57e6633...684e101...js:1	158 B	6 ms	
9707965e2d331161c888.css	200	text/css	main-33aecbf...js:1	1.4 kB	5 ms	
50000000-b0f27b680b5e36f9c766.js	200	javascript	main-33aecbf...js:1	4.4 kB	5 ms	
50000000.json	200	json	main-33aecbf...js:1	996 B	53 ms	
9707965e2d331161c888.css	200	stylesheet	main-33aecbf...js:1	(prefetch cac...	1 ms	
50000000-b0f27b680b5e36f9c766.js	200	script	main-33aecbf...js:1	(prefetch cac...	1 ms	
50000000.json	200	fetch	b1542cc...f376b59...js:1	(prefetch cac...	1 ms	

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

Incremental Static Regeneration

- 트래픽이 들어올 때 백그라운드에서 다시 렌더링하여 기존 페이지를 업데이트
- 요청 바로 이전 데이터를 렌더링
- `fallback: true` 를 주게 되면 로딩 후 실시간 렌더링
- 👍 빠른 빌드 시간
- 👍 빠른 초기 로딩

```
// This function gets called at build time on server-side.
// It may be called again, on a serverless function, if
// revalidation is enabled and a new request comes in
export async function getStaticProps() {
  const res = await fetch('https://.../posts')
  const posts = await res.json()

  return {
    props: {
      posts,
    },
    // Next.js will attempt to re-generate the page:
    // - When a request comes in
    // - At most once every second
    revalidate: 1, // In seconds
  }
}

export default Blog
```

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

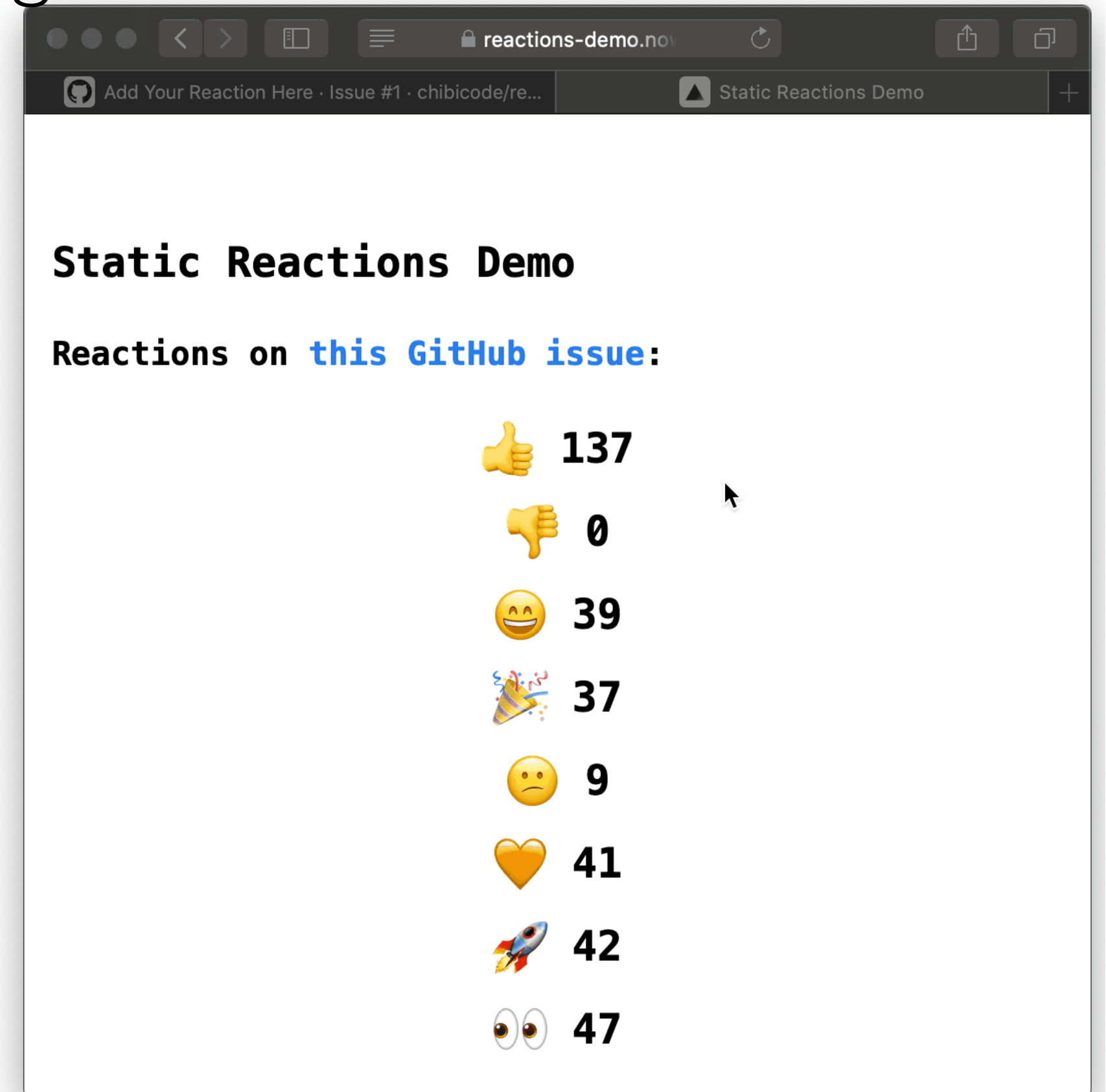
bundling

Backend / Server logic

And more

Incremental Static Regeneration

- 트래픽이 들어올 때 백그라운드에서 다시 렌더링하여 기존 페이지를 업데이트
- 요청 바로 이전 데이터를 렌더링
- `fallback: true` 를 주게 되면 로딩 후 실시간 렌더링
- 👍 빠른 빌드 시간
- 👍 빠른 초기 로딩



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

getServerSideProps (SSR)

- 런타임에 실행
- 서버 측 번들
- Next/link나 next/router로 페이지 전환시 JSON반환

`getServerSideProps` (Server-side Rendering)

If you export an `async` function called `getServerSideProps` from a page, Next.js will pre-render this page on each request using the data returned by `getServerSideProps`.

```
export async function getServerSideProps(context) {  
  return {  
    props: {}, // will be passed to the page component as props  
  }  
}
```

The `context` parameter is an object containing the following keys:

- `params`: If this page uses a dynamic route, `params` contains the route parameters. If the page name is `[id].js`, then `params` will look like `{ id: ... }`. To learn more, take a look at the [Dynamic Routing documentation](#).
- `req`: The HTTP IncomingMessage object.
- `res`: The HTTP response object.
- `query`: The query string.
- `preview`: `preview` is `true` if the page is in the preview mode and `false` otherwise. See the [Preview Mode documentation](#).
- `previewData`: The preview data set by `setPreviewData`. See the [Preview Mode documentation](#).
- `resolvedUrl`: A normalized version of the request URL that strips the `_next/data` prefix for client transitions and includes original query values.

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

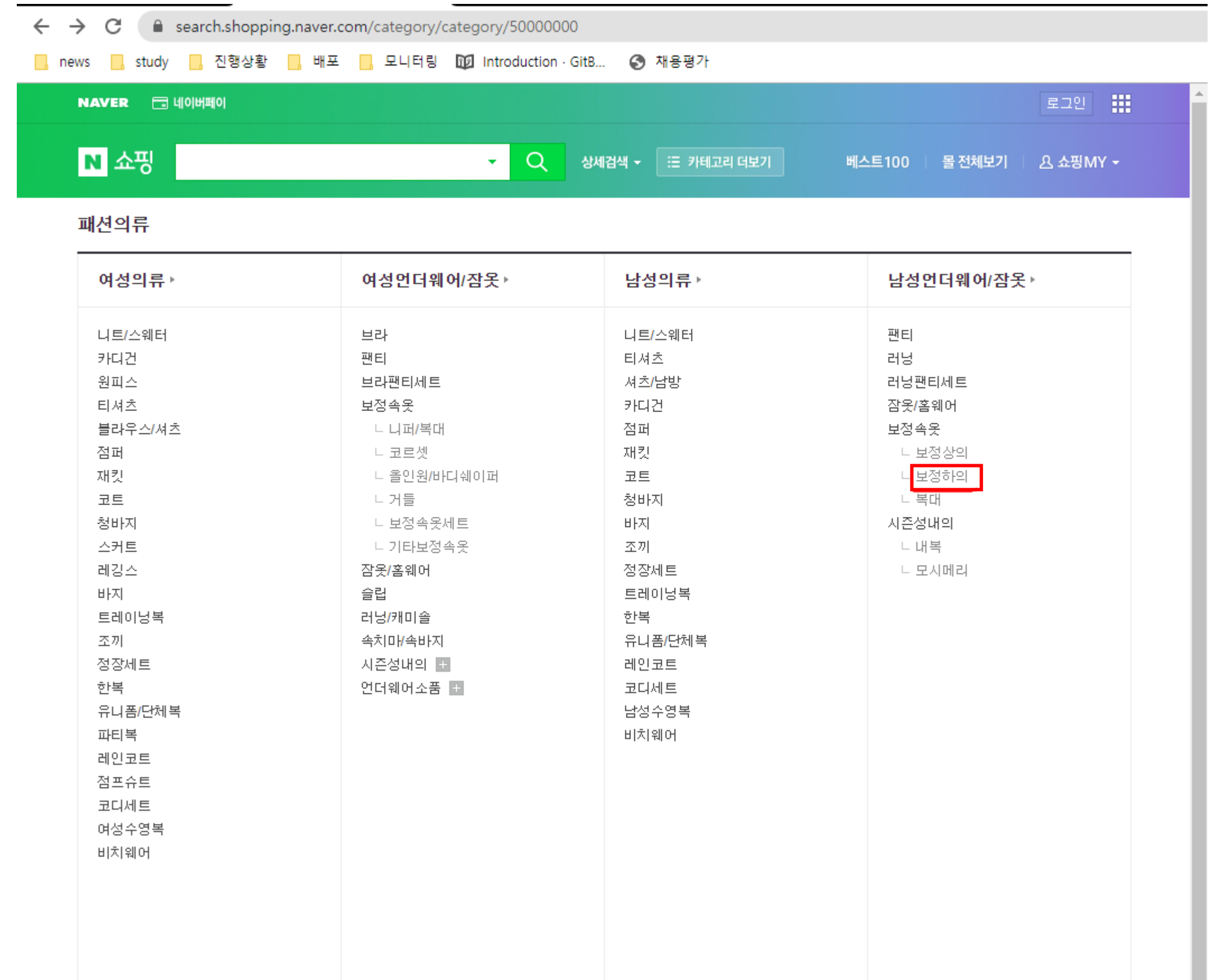
Security

bundling

Backend / Server logic

And more

getServerSideProps (SSR)



Name	Status	Type	Initiator	Size	Time	Waterfall
90eb75da3452a4235b81920abffe199565ad77ca.42e424687e83ded594f7.js	200	script	main-33aecbf...js:1	(prefetch cac...	2 ms	
8a4f3ea1f26be0519701fec8620f5ff71fb7a681.701553a1ed45b47628ec.js	200	script	main-33aecbf...js:1	(prefetch cac...	2 ms	
d1e51fa31865284c261824f0de0734689cb68d00.8476bfe3d304d0e25d06.js	200	script	main-33aecbf...js:1	(prefetch cac...	4 ms	
8c174c48e2c1409ba269.css	200	stylesheet	main-33aecbf...js:1	(prefetch cac...	3 ms	
d1e51fa31865284c261824f0de0734689cb68d00_CSS.17b25e9369f588ea6698.js	200	script	main-33aecbf...js:1	(prefetch cac...	3 ms	
category-fbd80945a571230ebca5.js	200	script	main-33aecbf...js:1	(prefetch cac...	3 ms	

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - SSR

- 서버 코드

```
export default {  
  path: '/report',  
  action: async ({ next, headers, ctx }: ActionParam) => {  
    if (isServer) {
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - SSR

- 서버 코드

```
export default {
  path: '/report',
  action: async ({ next, headers, ctx }: ActionParam) => {
    if (isServer) {
```

```
function createThunkMiddleware(extraArgument?) {
  return ({ dispatch, getState }) => next => action => {
    if (typeof action === 'function') {
      if (getState().serverRendered) {
        return Promise.resolve(null)
      }
      return action(dispatch, getState, extraArgument)
    }
    return next(action)
  }
}
```

```
const routeResult = await universalRouter.resolve(context)
const { redirect, component, title = DEFAULT_TITLE, ogTag } = routeResult
if (metas.length === 0) setOgTagMeta(setMeta, ogTag || makeOgTag(store.getState().searchParam, ctx.path))

if (redirect) {
  ctx.status = HttpStatus.MOVED_TEMPORARILY
  ctx.redirect(redirect)
} else {
  const ReactApp = renderWithContext(component, store)
  store.dispatch(setServerRendered(true))

  if (!ctx.status) {
    ctx.status = HttpStatus.OK
  }
  ctx.body = renderToString(ReactApp, {
    cssToBeInserted,
    title,
    metas,
    userAgent,
    state: store.getState(),
  })
} catch (error) {
  logger.error('[reactApp] error', error)
  const store = configureStore()
  const state = store.getState()
  const title = '네이버쇼핑'
  ctx.status = HttpStatus.INTERNAL_SERVER_ERROR
  const Layout = require(`${process.env.PROJECT_NAME}/components/Common/Layout`).default

  ctx.body = renderToString(
    renderWithContext(
      <Layout pathname={ctx.path}>
        <Error message={MessageTypes.CONNECT_FAILURE} />
      </Layout>,
      store,
    ),
    {
      state,
      cssToBeInserted,
      title,
      userAgent: ctx.req.headers['user-agent'],
    },
  )
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - SSR

- 서버 코드

```
export default {
  path: '/report',
  action: async ({ next, headers, ctx }: ActionParam) => {
    if (isServer) {
```

```
function createThunkMiddleware(extraArgument?) {
  return ({ dispatch, getState }) => next => action => {
    if (typeof action === 'function') {
      if (getState().serverRendered) {
        return Promise.resolve(null)
      }
      return action(dispatch, getState, extraArgument)
    }
    return next(action)
  }
}
```

```
const routeResult = await universalRouter.resolve(context)
const { redirect, component, title = DEFAULT_TITLE, ogTag } = routeResult
if (metas.length === 0) setOgTagMeta(setMeta, ogTag || makeOgTag(store.getState().searchParam, ctx.path))

if (redirect) {
  ctx.status = HttpStatus.MOVED_TEMPORARILY
  ctx.redirect(redirect)
} else {
  const ReactApp = renderWithContext(component, store)
  store.dispatch(setServerRendered(true))

  if (!ctx.status) {
    ctx.status = HttpStatus.OK
  }
  ctx.body = renderToString(ReactApp, {
    cssToBeInserted,
    title,
    metas,
    userAgent,
    state: store.getState(),
    <script id="reduxData" window.__PRELOADED_STATE__=${serialize(state)} />
  })
} catch (error) {
  logger.error('[reactApp] error', error)
  const store = configureStore()
  const state = store.getState()
  const title = '네이버쇼핑'
  ctx.status = HttpStatus.INTERNAL_SERVER_ERROR
  const Layout = require(`${process.env.PROJECT_NAME}/components/Common/Layout`).default

  ctx.body = renderToString(
    renderWithContext(
      <Layout pathname={ctx.path}>
        <Error message={MessageTypes.CONNECT_FAILURE} />
      </Layout>,
      store,
    ),
    {
      state,
      cssToBeInserted,
      title,
      userAgent: ctx.req.headers['user-agent'],
    }
  )
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - SSR

- 서버 코드

```
export default {
  path: '/report',
  action: async ({ next, headers, ctx }: ActionParam) => {
    if (isServer) {
```

```
function createThunkMiddleware(extraArgument?) {
  return ({ dispatch, getState }) => next => action => {
    if (typeof action === 'function') {
      if (getState().serverRendered) {
        return Promise.resolve(null)
      }
      return action(dispatch, getState, extraArgument)
    }
    return next(action)
  }
}
```

```
const routeResult = await universalRouter.resolve(context)
const { redirect, component, title = DEFAULT_TITLE, ogTag } = routeResult
if (metas.length === 0) setOgTagMeta(setMeta, ogTag || makeOgTag(store.getState().searchParam, ctx.path))

if (redirect) {
  ctx.status = HttpStatus.MOVED_TEMPORARILY
  ctx.redirect(redirect)
} else {
  const ReactApp = renderWithContext(component, store)
  store.dispatch(setServerRendered(true))

  if (!ctx.status) {
    ctx.status = HttpStatus.OK
  }
  ctx.body = renderToString(ReactApp, {
    cssToBeInserted,
    title,
    metas,
    userAgent,
    state: store.getState(),
  })
} catch (error) {
  logger.error('[reactApp] error', error)
  const store = configureStore()
  const state = store.getState()
  const title = '네이버쇼핑'
  ctx.status = HttpStatus.INTERNAL_SERVER_ERROR
  const Layout = require(`${process.env.PROJECT_NAME}/components/Common/Layout`).default

  ctx.body = renderToString(
    renderWithContext(
      <Layout pathname={ctx.path}>
        <Error message={MessageTypes.CONNECT_FAILURE} />
      </Layout>,
      store,
    ),
    {
      state,
      cssToBeInserted,
      title,
      userAgent: ctx.req.headers['user-agent'],
    }
  )
}
```


2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - SSR

- 클라이언트 코드

```
export const store = isBrowser && configureStore((window as any).__PRELOADED_STATE__)
```

```
export default function configureStore(initialState?) {  
  const store = createStore<RootState, Action<unknown>, unknown, unknown>(  
    require(`${process.env.PROJECT_NAME}/reducer`).default,  
    initialState,  
    enhancer,  
  )  
  
  if (isBrowser) {  
    delete window['__PRELOADED_STATE__']  
  }  
}
```

```
if (!isInitialRender && store.getState().serverRendered) {  
  store.dispatch(setServerRendered(false))  
}  
  
try {  
  const metas = []  
  
  if (!isInitialRender && action === 'POP') {  
    ReactDOM.render(  
      renderWithContext(  
        <Layout pathname={location.pathname}>  
          <Loading isFull={true} />  
        </Layout>,  
        insertCss,  
        store,  
      ),  
      rootEl,  
    )  
  }  
  
  const context = { ...  
}  
  
  const routeResult = await universalRouter.resolve(context)  
  store.getState().serverRendered && store.dispatch(setServerRendered(false))  
  const { redirect, component, title = DEFAULT_TITLE } = routeResult  
  // Prevent multiple page renders during the routing process  
  if (currentLocation.key !== location.key) {  
    return  
  }  
  
  if (redirect) {  
    history.replace(redirect)  
    store.getState().redirect && store.dispatch(setRedirect(null))  
    return  
  }  
  
  const renderReactApp = module.hot ? ReactDOM.render : ReactDOM.hydrate  
  appInstance = renderReactApp(renderWithContext(component, insertCss, store), rootEl, () => {  
    setMetaTags(metas)  
  
    // not to make unnecessary CSSOM from second client-side rendering  
    if (isInitialRender) {  
      const styleElem = document.getElementById(DOCUMENT_HEAD_STYLE_ID)  
      return styleElem && styleElem.parentNode.removeChild(styleElem)  
    }  
    document.title = title  
  })  
} catch (error) {
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - SSR

- 클라이언트 코드

```
export const store = isBrowser && configureStore((window as any).__PRELOADED_STATE__)
```

```
export default function configureStore(initialState?) {  
  const store = createStore<RootState, Action<unknown>, unknown, unknown>(  
    require(`${process.env.PROJECT_NAME}/reducer`).default,  
    initialState,  
    enhancer,  
  )  
  
  if (isBrowser) {  
    delete window['__PRELOADED_STATE__']  
  }  
}
```

```
if (!isInitialRender && store.getState().serverRendered) {  
  store.dispatch(setServerRendered(false))  
}  
  
try {  
  const metas = []  
  
  if (!isInitialRender && action === 'POP') {  
    ReactDOM.render(  
      renderWithContext(  
        <Layout pathname={location.pathname}>  
          <Loading isFull={true} />  
        </Layout>,  
        insertCss,  
        store,  
      ),  
      rootEl,  
    )  
  }  
  
  const context = { ... }  
}  
  
const routeResult = await universalRouter.resolve(context)  
store.getState().serverRendered && store.dispatch(setServerRendered(false))  
const { redirect, component, title = DEFAULT_TITLE } = routeResult  
// Prevent multiple page renders during the routing process  
if (currentLocation.key !== location.key) {  
  return  
}  
  
if (redirect) {  
  history.replace(redirect)  
  store.getState().redirect && store.dispatch(setRedirect(null))  
  return  
}  
  
const renderReactApp = module.hot ? ReactDOM.render : ReactDOM.hydrate  
appInstance = renderReactApp(renderWithContext(component, insertCss, store), rootEl, () => {  
  setMetaTags(metas)  
  
  // not to make unnecessary CSSOM from second client-side rendering  
  if (isInitialRender) {  
    const styleElem = document.getElementById(DOCUMENT_HEAD_STYLE_ID)  
    return styleElem && styleElem.parentNode.removeChild(styleElem)  
  }  
  document.title = title  
})  
} catch (error) {
```

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

Routing

- 파일 시스템 라우팅

Index routes

The router will automatically route files named `index` to the root of the directory.

- `pages/index.js` → `/`
- `pages/blog/index.js` → `/blog`

Nested routes

The router supports nested files. If you create a nested folder structure files will be automatically routed in the same way still.

- `pages/blog/first-post.js` → `/blog/first-post`
- `pages/dashboard/settings/username.js` → `/dashboard/settings/username`

Dynamic route segments

To match a dynamic segment you can use the bracket syntax. This allows you to match named parameters.

- `pages/blog/[slug].js` → `/blog/:slug` (`/blog/hello-world`)
- `pages/[username]/settings.js` → `/:username/settings` (`/foo/settings`)
- `pages/post/[...all].js` → `/post/*` (`/post/2020/id/title`)

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

Routing

- Shallow Routing

Shallow routing allows you to change the URL without running data fetching methods again, that includes `getServerSideProps`, `getStaticProps`, and `getInitialProps`.

You'll receive the updated `pathname` and the `query` via the `router` object (added by `useRouter` or `withRouter`), without losing state.

To enable shallow routing, set the `shallow` option to `true`. Consider the following example:

```
import { useEffect } from 'react'
import { useRouter } from 'next/router'

// Current URL is '/'
function Page() {
  const router = useRouter()

  useEffect(() => {
    // Always do navigations after the first render
    router.push('/?counter=10', undefined, { shallow: true })
  }, [])

  useEffect(() => {
    // The counter changed!
  }, [router.query.counter])
}

export default Page
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - Routing

- Routes 구성

```
import UniversalRouter from 'universal-router'
import { appendRoute } from '@shopping/common'

const routes = require(`${process.env.PROJECT_NAME}/routes`).default

export default new UniversalRouter(routes, {
  async resolveRoute(context, params) {
    if (typeof context.route.action === 'function') {
      const { route } = context
      if (route.children && route.children.some(child => child['importRoute'] && child['codeSplitPath'])) {
        await appendRoute(context)
      }
      return route.action(context, params)
    }
    return undefined
  },
})
```

```
export const SearchPath = {
  ALL: '/all',
  CATEGORY: '/category',
  SHOPLIST: '/shoplist',
  MALL: '/mall',
  FORYOU: '/foryou',
  SHOPPING_LENS: '/image',
}

export default {
  path: '/search',
  children: [
    {
      codeSplitPath: SearchPath.ALL,
      importRoute: () => import('./all'),
    },
  ],
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - Routing

- Routes 구성
- Code Splitting

```
const routes = (await routeToBeImported.importRoute()).default
/* 주의!
matched 된 {codeSplitPath, importRoute} 을 import된 route object 로 replace 하지 않고, import된 route object 를 route.children 에 append 해야 함
- appendRoute 로직이 언제나 똑같이 동작해야 하기 때문
- appendRoute 결과로 {codeSplitPath, importRoute} 가 제거되면, 다음 appendRoute 호출 시 로직이 다르게 동작할 수 있음
  - 제거된 {codeSplitPath, importRoute} 의 하위 {codeSplitPath, importRoute} 가 childPath 와 match 될 수 있음
*/
route.children = [
  ...route.children.slice(0, routeOrderToBeImported),
  ...(typeChecker.isArray(routes) ? routes : [routes]),
  ...route.children.slice(routeOrderToBeImported),
]
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - Routing

- Routes 구성
- Code Splitting
- Shallow Routing

```
history.listen(onLocationChange)
```

```
const onLocationChange = async (location, action?) => {  
  if (action === 'POP' && getLocationPath(currentLocation) === getLocationPath(location)) {  
    return  
  }  
  
  if (action === 'REPLACE' && getLocationPath(currentLocation) === getLocationPath(location)) {  
    return  
  }  
  
  if (action === 'REPLACE' && location.state && location.state.doNotRender === true) {  
    currentLocation = location  
    return  
  }  
}
```

```
import { createBrowserHistory } from 'history'  
  
const history = process.env.IS_BROWSER && createBrowserHistory()  
  
export default history  
  
export const FIRST_PAGE = 'firstPage'  
  
export const getLocationKey = location => {  
  return location.key || FIRST_PAGE  
}  
  
export const getLocationPath = location => {  
  return location.pathname + location.search + location.hash  
}  
  
export const pushCurrent = (state: object = {}): void => {  
  if (!process.env.IS_BROWSER) {  
    return  
  }  
  
  if (!history.location) {  
    return  
  }  
  
  history.location.state = {  
    ...(history.location && history.location.state),  
    ...state,  
    doNotRender: true,  
  }  
  
  history.push(history.location)  
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

bundling

Backend / Server logic

And more

예전 코드 - Routing

- Routes 구성
- Code Splitting
- Shallow Routing
- CSR (Link 이동)

redux state 가져오기

- component
ex) `pc/components/Store/General/Header/index.tsx`

```
import flipState from 'common/utils/component/flipState'  
import { StateKey } from 'common/utils/flipState'  
  
interface OwnProps {  
  curState: StateKey  
}  
  
export default flipState(  
  connect<StateProps, DispatchProps, OwnProps>( (state: RootState, ownProps: OwnProps) => ({  
    store: state.store[ownProps.curState],  
    view: state.storeView[ownProps.curState],  
  })),  
  )(Header),  
)
```

- route, action creator
ex) `pc/routes/store/general/index.tsx`

```
action: async ({ store, params, baseUrl, pathname, action, curState }: ActionParam) => {  
  const contentRelatedProducts = store.getState().contentRelatedProducts[curState]  
}
```

- `reducer.ts`

```
export default combineReducers({  
  store: flipReducer(store),  
})
```

- `commonState.ts`

```
store: FlipState<StoreChannel>
```

action creator 를 통한 redux state 설정

- redux action creator 정의

ex) `common/ducks/store/general/store.ts`

```
export const fetchStore = (curState: StateKey, id: string) => async (dispatch, getState) => {  
  dispatch({  
    curState,  
    ...  
  })  
}
```

ex) `common/ducks/product/option.ts`

```
export const initializeSelectedOption=(curState: StateKey, selectedOption: SelectedOptions) => {  
  return {  
    curState,  
    ...  
  }  
}
```

- route action 에서 redux action creator 호출

ex) `pc/routes/store/general/index.tsx`

```
action: async ({ store, next, path, baseUrl, params, ctx, setMeta, curState }: ActionParam) => {  
  await store.dispatch(fetchStore(curState, storeId))  
}
```

- component 에서 redux action creator 호출

ex) `pc/components/Store/Designer/Content/New/index.tsx`

```
const New: React.SFC<Props> = ({ curState }) => {  
  const handleClickNews = () => {  
    isBrowser && saveNews(curState, window.location)  
  }  
}
```


2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Dynamic import

- route-based

<https://nextjs.org/blog/next-9-2#improved-code-splitting-strategy>

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Dynamic import

- route-based
- module (ES2020 dynamic `import()`)

```
import { useState } from 'react'

const names = ['Tim', 'Joe', 'Bel', 'Max', 'Lee']

export default function Page() {
  const [results, setResults] = useState()

  return (
    <div>
      <input
        type="text"
        placeholder="Search"
        onChange={async (e) => {
          const { value } = e.currentTarget
          // Dynamically load fuse.js
          const Fuse = (await import('fuse.js')).default
          const fuse = new Fuse(names)

          setResults(fuse.search(value))
        }}
      />
      <pre>Results: {JSON.stringify(results, null, 2)}</pre>
    </div>
  )
}
```

<https://nextjs.org/blog/next-9-2#improved-code-splitting-strategy>

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Dynamic import

- route-based
- module (ES2020 `dynamic `import()``)
- component (SSR이 포함된 `react.lazy`)

```
import dynamic from 'next/dynamic'

const DynamicComponent = dynamic(() => import('../components/hello'))

function Home() {
  return (
    <div>
      <Header />
      <DynamicComponent />
      <p>HOME PAGE is here!</p>
    </div>
  )
}

export default Home
```

<https://nextjs.org/blog/next-9-2#improved-code-splitting-strategy>

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Code Splitting

- 페이지별 청크 분리
- 필요한만큼 common 청크 생성 (최대 25개)

Entrypoints



page1

moduleA
moduleB

page2

moduleB
moduleC

page3

moduleB
moduleC



Bundles



common



moduleB
moduleC

page1

page1
moduleA

page2

page2

page3

page3

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

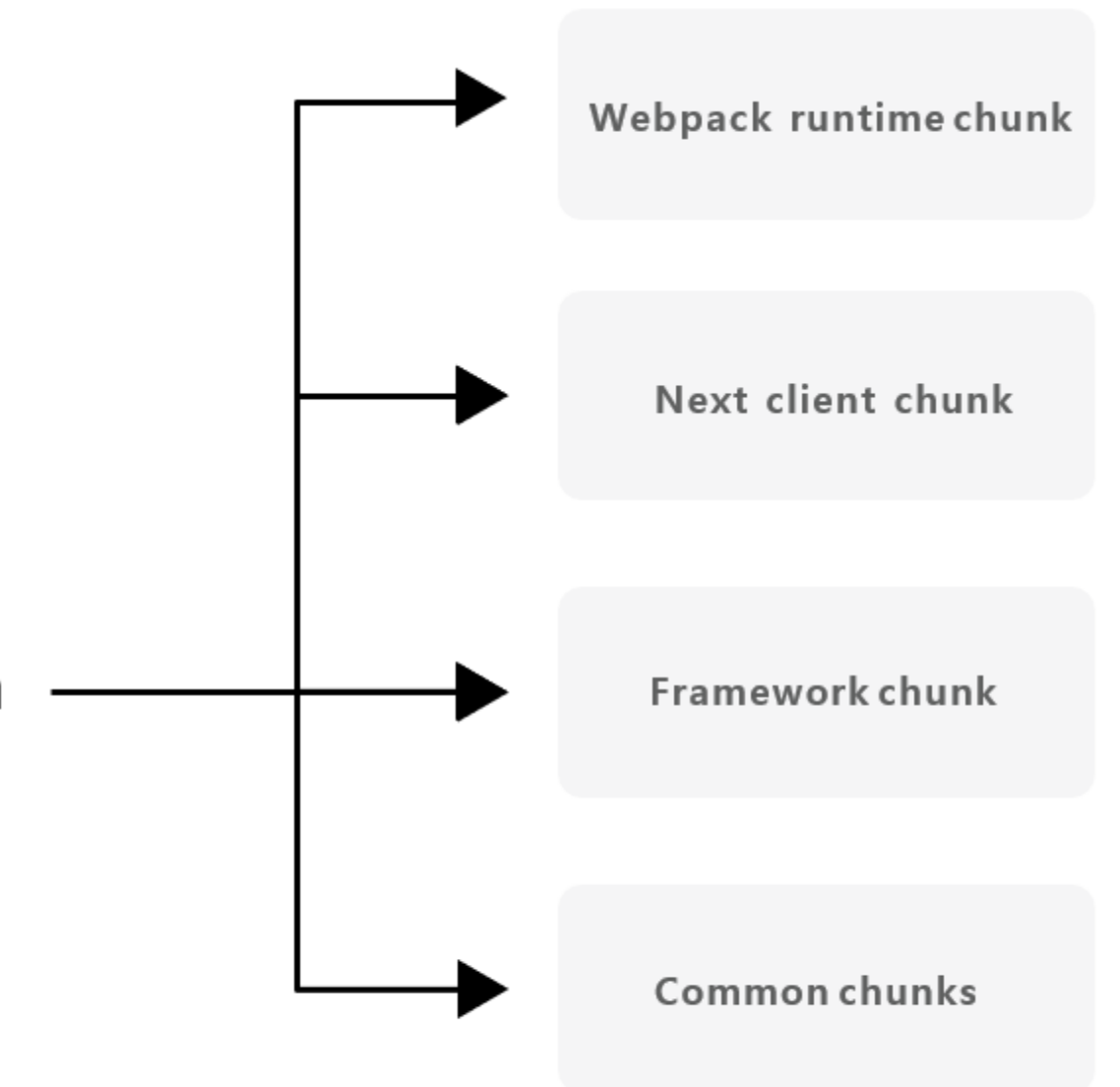
And more

Code Splitting

- 페이지별 청크 분리
- 필요한만큼 common 청크 생성 (최대 25개)
- 프레임워크 청크 (react, react-dom)
- Webpack 런타임 청크
- Next.js 클라이언트 런타임 청크
- 최소 크기 20kb, 최대 160kb



SplitChunksPlugin



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

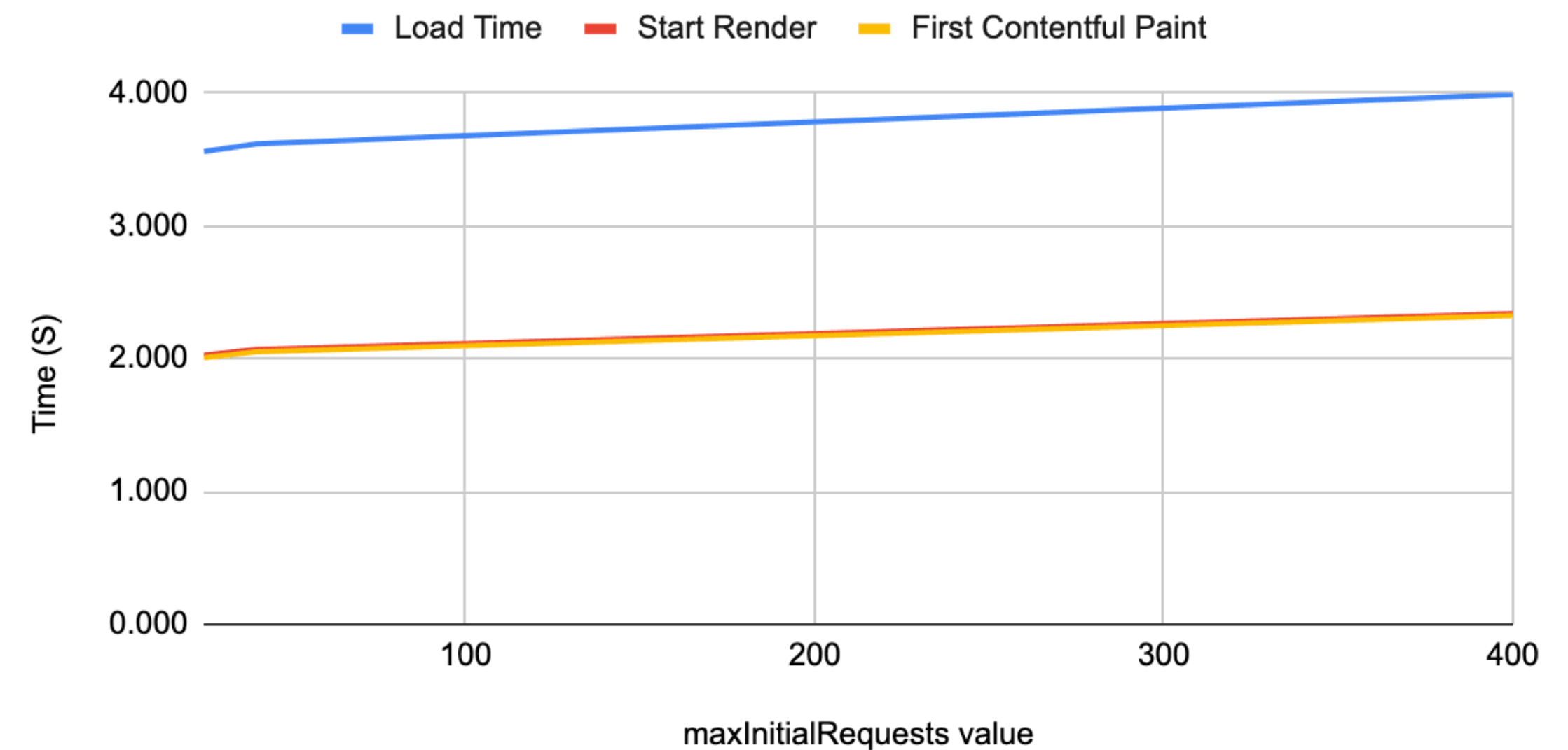
And more

Code Splitting

- 페이지별 청크 분리
- 필요한만큼 common 청크 생성 (최대 25개)
- 프레임워크 청크 (react, react-dom)
- Webpack 런타임 청크
- Next.js 클라이언트 런타임 청크
- 최소 크기 20kb, 최대 160kb
- maxInitialRequests 25

Page Load Performance by maxInitialRequests value

Average of three runs on Zeit.co



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Code Splitting

- 페이지별 청크 분리
- 필요한만큼 common 청크 생성 (최대 25개)
- 프레임워크 청크 (react, react-dom)
- Webpack 런타임 청크
- Next.js 클라이언트 런타임 청크
- 최소 크기 20kb, 최대 160kb
- maxInitialRequests 25

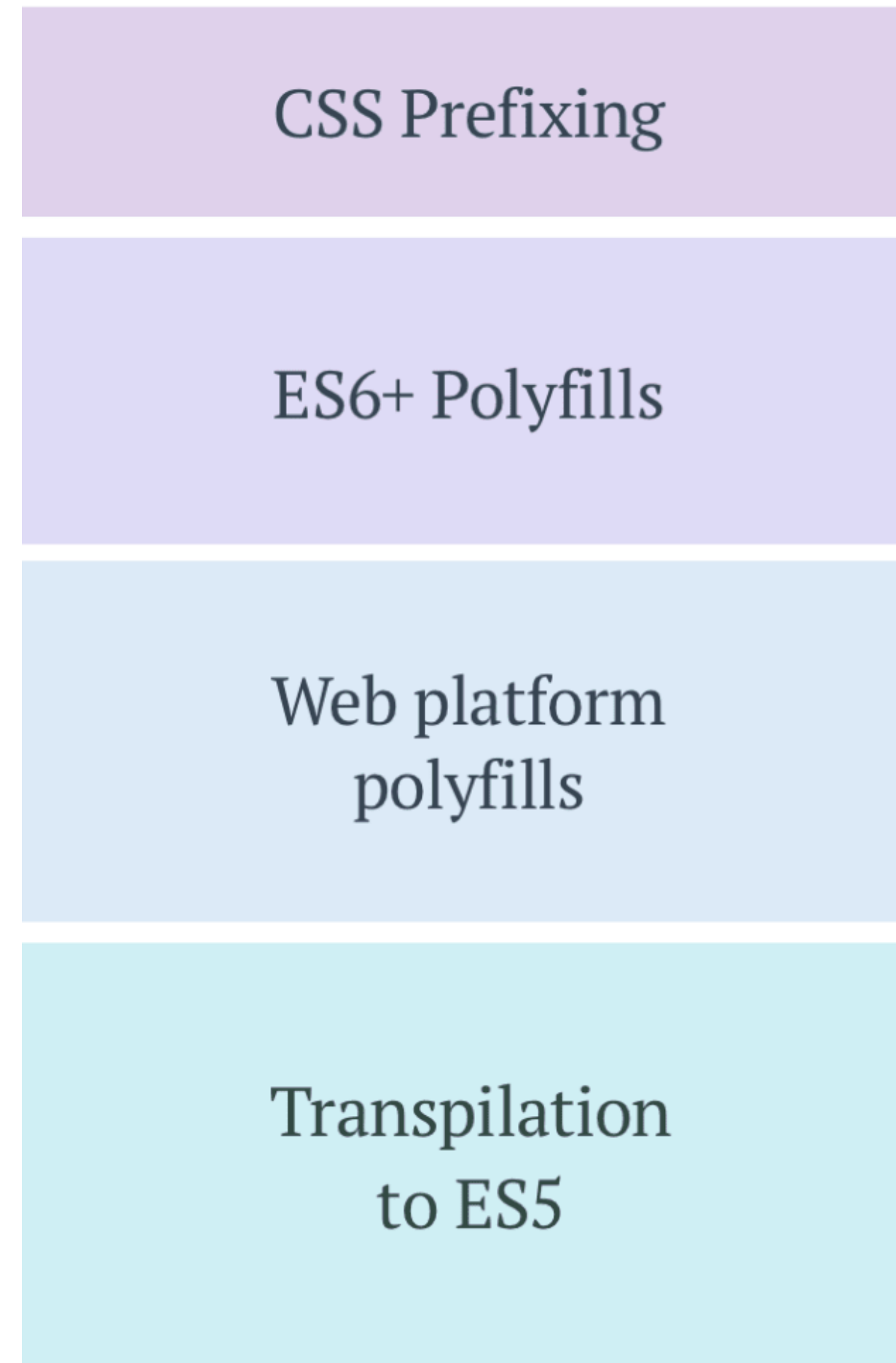
```
prodGranular: {
  chunks: 'all',
  cacheGroups: {
    default: false,
    vendors: false,
    // In webpack 5 vendors was renamed to defaultVendors
    defaultVendors: false,
    framework: {
      chunks: 'all',
      name: 'framework',
      // This regex ignores nested copies of framework libraries so they're
      // bundled with their issuer.
      // https://github.com/vercel/next.js/pull/9012
      test: /(?!(node_modules.*)(node_modules)(react|react-dom|scheduler|prop-types|use-subscription)(\|)/,
      priority: 40,
      // Don't let webpack eliminate this chunk (prevents this chunk from
      // becoming a part of the commons chunk)
      enforce: true,
    },
    lib: {
      test(module: { size: Function; identifier: Function }): boolean {
        return (
          module.size() > 160000 &&
          /node_modules[\/\]/.test(module.identifier())
        )
      },
      name(module: {
        type: string
        libIdent?: Function
        updateHash: (hash: crypto.Hash) => void
      }): string {
        const hash = crypto.createHash('sha1')
        if (isModuleCSS(module)) {
          module.updateHash(hash)
        } else {
          if (!module.libIdent) {
            throw new Error(
              `Encountered unknown module type: ${module.type}. Please open an issue.`
            )
          }
          hash.update(module.libIdent({ context: dir }))
        }
        return hash.digest('hex').substring(0, 8)
      },
      priority: 30,
      minChunks: 1,
      reuseExistingChunk: true,
    },
    commons: {
      name: 'commons',
      minChunks: totalPages,
      priority: 20,
    },
  },
  shared: {
    name(module, chunks) {
      return (
        crypto
          .createHash('sha1')
          .update(
            chunks.reduce(
              (acc: string, chunk: webpack.compilation.Chunk) => {
                return acc + chunk.name
              },
              ''
            )
          )
          .digest('hex') + (isModuleCSS(module) ? '_CSS' : '')
      )
    },
    priority: 10,
    minChunks: 2,
    reuseExistingChunk: true,
  },
  maxInitialRequests: 25,
  minSize: 20000,
}
```

2.3 Web Framework 구성요소

- UI Component Model
- Pre-rendering HTML
- Routing
- CSS Encapsulation
- Data Flow
- Deployment / Releases
- Compilation
- Security
- Bundling
- Backend / Server logic
- And more

Differential Loading

LEGACY BUNDLE



MODERN BUNDLE



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Differential Loading

- Polyfill은 기본
- 트랜스파일 결과물도 분기 처리

```
// next.config.js
module.exports = {
  experimental: { modern: true }
}
```

Modern JavaScript

Modern browsers



async/await, arrows,
classes, ...rest, etc

```
<script
  type=module
  src=modern.js>
</script>
```

Legacy browsers



ES5, helpers,
additional polyfills, etc

```
<script
  nomodule
  src=legacy.js>
</script>
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

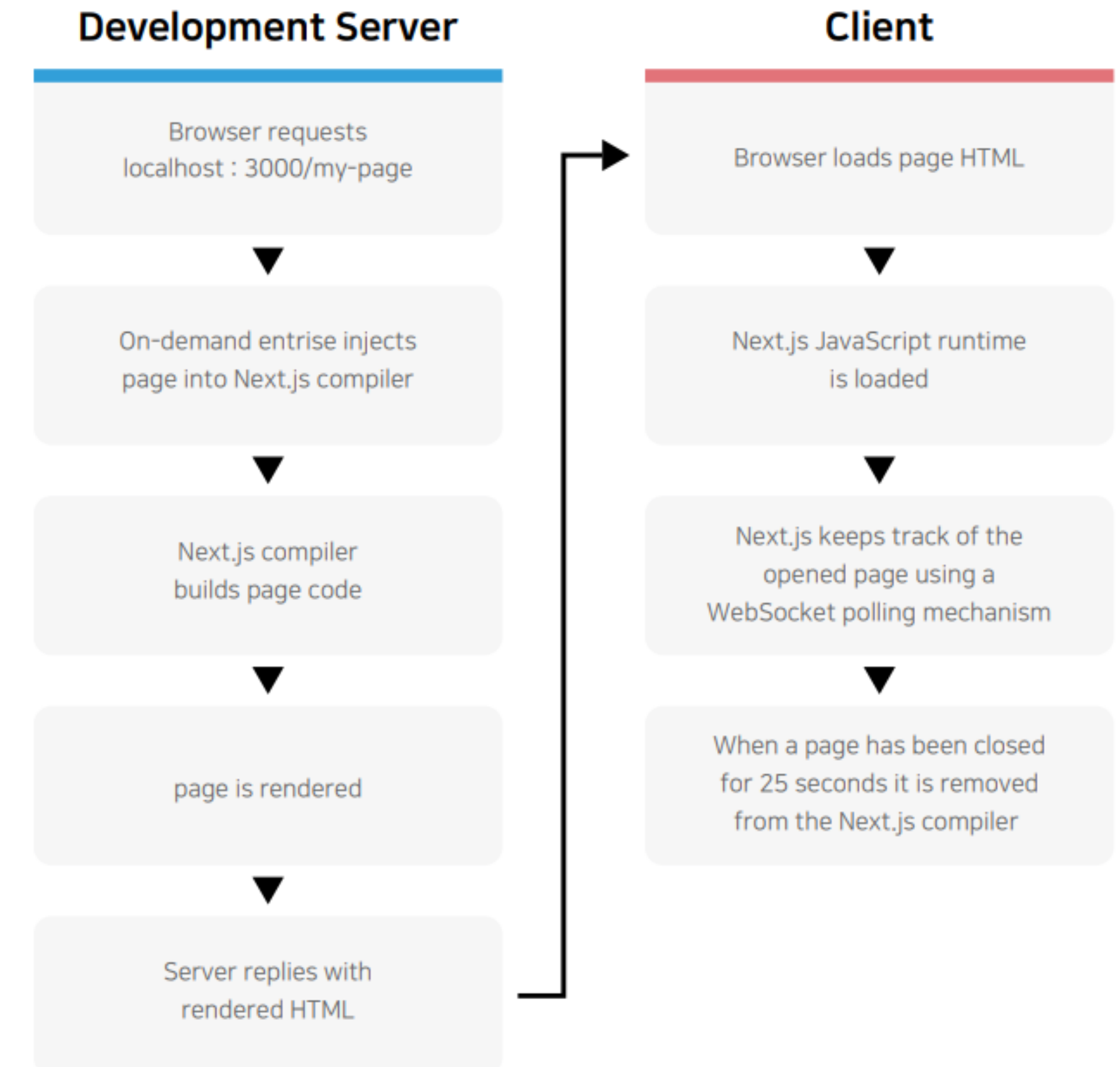
Bundling

Backend / Server logic

And more

개발 서버 최적화

- 모든 페이지가 아닌 페이지 액세스할때
- 웹소켓 기반 5초마다 체크
- 25초 동안 액세스 하지 않으면 캐시 제거



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

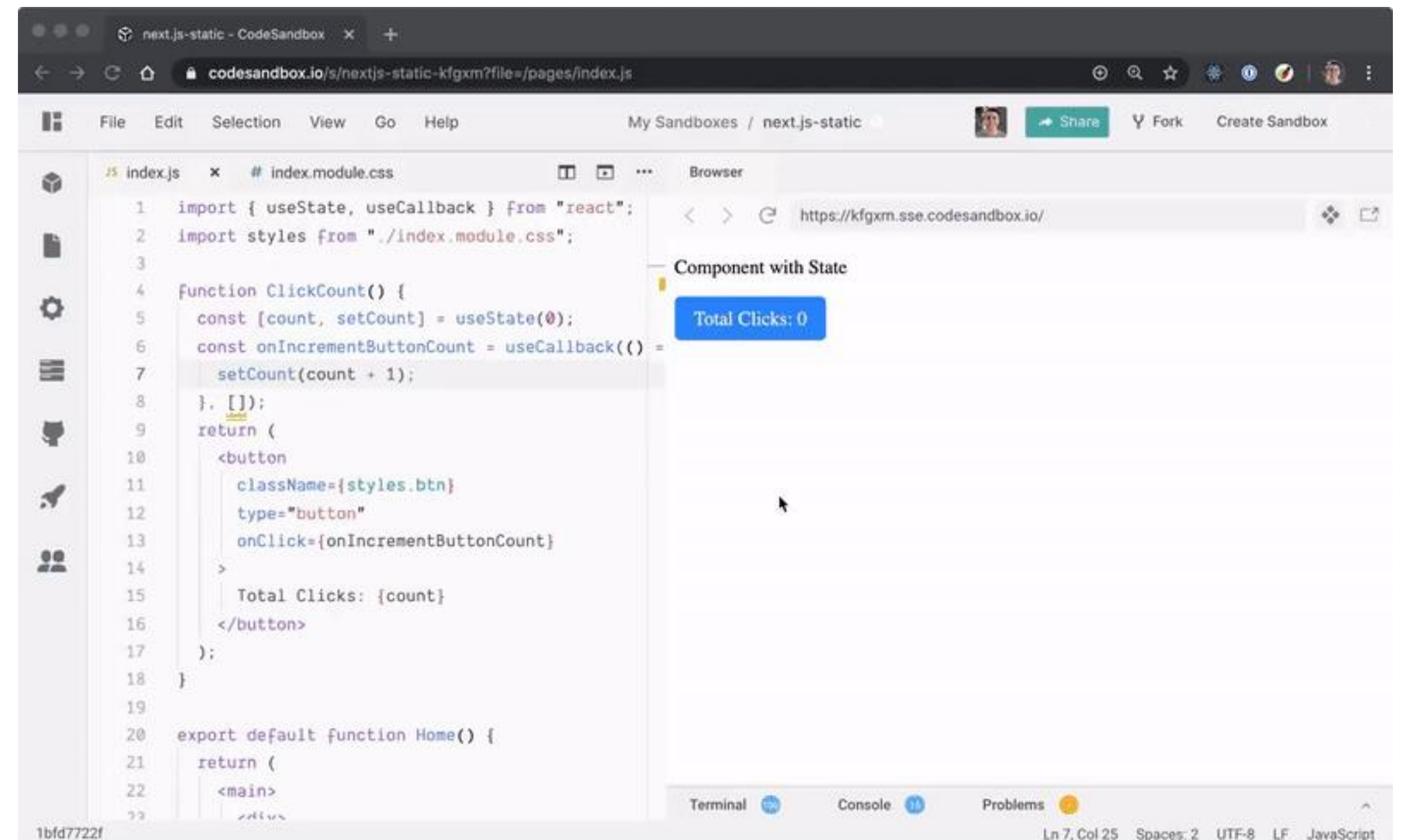
Bundling

Backend / Server logic

And more

Fast Refresh

- next.js, facebook 내부 프로젝트
- compile and runtime errors
- 클래스 컴포넌트, hoc 상태유지X
- 소스맵 자동 적용
- IDE 연동



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - HMR

```
webpackConfig.entry = [
  require.resolve('./polyfills'),
  `webpack-hot-middleware/client?reload=true&path=${config.protocol}://${config.host}:${config.serverForAssetsPort}/_webpack_hmr`,
]
```

```
webpackConfig.output = {
  hotUpdateChunkFilename: 'static/[id].[hash].hot-update.js',
  hotUpdateMainFilename: 'static/[hash].hot-update.json',
}
```

```
webpackConfig.plugins = webpackConfig.plugins.concat([
  // This is necessary to emit hot updates (currently CSS only):
  new webpack.HotModuleReplacementPlugin(),
])
```

```
// Enable Hot Module Replacement (HMR)
const hotModule = (module as any).hot
if (hotModule) {
  hotModule.accept('common/utils/universalRouter', () => {
    if (appInstance && appInstance.updater.isMounted(appInstance)) {
      try {
        // Force-update the whole tree, including components that refuse to update
        deepForceUpdate(appInstance)
      } catch (error) {
        logger.error(`[hotModule] error, ${error}`)
        ReactDOM.render(<>HMR 과정에 오류가 발생했습니다. f5를 눌러 페이지를 새로고침해주세요.</>, rootE1)
        return
      }
    }
  })
  onLocationChange(currentLocation)
}
```

```
if (module.hot) {
  const reducerPath = `${process.env.PROJECT_NAME}/reducer`
  module.hot.accept(reducerPath, () => {
    const nextRootReducer = require(`${process.env.PROJECT_NAME}/reducer`).default
    store.replaceReducer(nextRootReducer)
  })
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - HMR

```
clientCompiler.plugin('compile', () => {
  clientCompiling = true
})

clientCompiler.plugin('done', stats => {
  if (stats.hasErrors()) {
  } else {
    clientCompiling = false
  }
})

const serverForAssetsApp = express()
const webpackDevMiddleware = createWebpackDevMiddleware(clientCompiler, {
  headers: { 'Access-Control-Allow-Origin': serverUrl },
  publicPath: configWebpackClient.output.publicPath,
})

serverForAssetsApp.use(webpackDevMiddleware)
serverForAssetsApp.use(createWebpackHotMiddleware(clientCompiler))

let server = null
let serverWatcher = null

const serverForAssets = serverForAssetsApp.listen(serverForAssetsPort, err => {
  if (err) {
    return console.log(err)
  }
  if (isInteractive) {
    clearConsole()
  }
  console.log(chalk.cyan(`Starting the server for serving assets on port ${serverForAssetsPort}...\n`))

  // Server side

  process.env.REACT_APP_SERVER_PORT = serverPort
  const configWebpackServer = config.webpackServerConfig
  const serverCompiler = webpack(configWebpackServer)
```

```
const wait = () => {
  if (serverCompiling) return
  if (clientCompiling) setTimeout(wait, 50)
  else startServer()
}

serverCompiler.plugin('compile', () => {
  serverCompiling = true
})

serverCompiler.plugin('done', stats => {
  serverCompiling = false

  try {
    if (stats.hasErrors()) {
      console.log(stats.toString())
      return
    }

    wait()
  } catch (err) {
    console.error(err)
  }
})
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Webpack 5 지원

- 이전 버전과 호환성
- [Improved Tree-Shaking](#)
- [Persistent Caching](#)
- [Deterministic chunk and module ids](#)
- [nextjs.org](#), [vercel.com](#) 사용 중

```
{  
  "resolutions": {  
    "webpack": "^5.0.0"  Yo  
  }  
}
```

experimental.unstable_webpack5cache : true or [next@9.5.3](#) ↑

```
// next.config.js  
const withTM = require('next-transpile-modules')(['somemodule', 'and-another'], { unstable_webpack5: true });  
  
module.exports = withTM();
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - Webpack

- CRA 1버전 eject

```
npm run eject
```

Note: this is a one-way operation. Once you eject, you can't go back!

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - Webpack

- CRA 1버전 eject
- Webpack3 -> webpack4 마이그레이션

To v4 from v3

This guide only shows major changes that affect end users. For more details please see [the changelog](#).

Node.js v4

If you are still using Node.js v4 or lower, you need to upgrade your Node.js installation to Node.js v6 or higher.

Instructions for upgrading your Node.js version can be found [here](#).

CLI

The CLI has moved to a separate package: `webpack-cli`. You need to install it before using `webpack`, see [basic setup](#).

The installation guide can be found [here](#).

Update plugins

Many 3rd party plugins need to be updated to the latest versions to be compatible with `webpack 4`. Links to popular plugins can be found [here](#).

mode

Add the new `mode` option to your configuration. Set it to `'production'`, `'development'` or `'none'` depending on your configuration type.

`webpack.config.js`

```
module.exports = {  
  // ...  
  + mode: 'production',  
}
```

The purpose of `'development'` mode and `'production'` mode is different. You can use `webpack-merge` as shown in [production guide](#) to optimize configurations.

Deprecated/Removed plugins

These plugins can be removed from configuration as they are default in production mode:

`webpack.config.js`

```
module.exports = {  
  // ...  
  plugins: [  
    - new NoEmitOnErrorsPlugin(),  
    - new ModuleConcatenationPlugin(),  
    - new DefinePlugin({ "process.env.NODE_ENV": JSON.stringify("production") }),  
    - new UglifyJsPlugin()  
  ],  
}
```


2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - Webpack

- CRA 1버전 eject
- Webpack3 -> webpack4 마이그레이션
- 웹팩 플러그인, 로더 관리

```
"dependencies": {
  "@babel/core": "7.9.0",
  "@svgr/webpack": "4.3.3",
  "babel-jest": "^24.9.0",
  "babel-loader": "8.1.0",
  "babel-plugin-named-asset-import": "^0.3.6",
  "babel-preset-react-app": "^9.1.2",
  "camelcase": "5.3.1",
  "case-sensitive-paths-webpack-plugin": "2.3.0",
  "css-loader": "3.4.2",
  "dotenv": "8.2.0",
  "dotenv-expand": "5.1.0",
  "file-loader": "4.3.0",
  "fs-extra": "^8.1.0",
  "html-webpack-plugin": "4.0.0-beta.11",
  "identity-obj-proxy": "3.0.0",
  "mini-css-extract-plugin": "0.9.0",
  "optimize-css-assets-webpack-plugin": "5.0.3",
  "pnp-webpack-plugin": "1.6.4",
  "postcss-flexbugs-fixes": "4.1.0",
  "postcss-loader": "3.0.0",
  "postcss-normalize": "8.0.1",
  "postcss-preset-env": "6.7.0",
  "postcss-safe-parser": "4.0.1",
  "react": "^16.13.1",
  "react-app-polyfill": "^1.0.6",
  "react-dev-utils": "^10.2.1",
  "react-dom": "^16.13.1",
  "resolve": "1.15.0",
  "resolve-url-loader": "3.1.1",
  "sass-loader": "8.0.2",
  "semver": "6.3.0",
  "style-loader": "0.23.1",
  "terser-webpack-plugin": "2.3.8",
  "ts-pnp": "1.1.6",
  "url-loader": "2.3.0",
  "webpack": "4.42.0",
  "webpack-dev-server": "3.11.0",
  "webpack-manifest-plugin": "2.2.0",
  "workbox-webpack-plugin": "4.3.1"
},
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - Webpack

- CRA 1버전 eject
- Webpack3 -> webpack4 마이그레이션
- 웹팩 플러그인, 로더 관리
- 바벨 폴리필, 플러그인 관리

```
module.exports = api => {  
  api.cache(true)  
  
  return {  
    presets: [['@babel/preset-env', { modules: false, useBuiltIns: 'entry', corejs: 3 }], '@babel/preset-react'],  
    plugins: [  
      '@babel/plugin-proposal-class-properties',  
      '@babel/plugin-syntax-dynamic-import',  
      '@babel/plugin-proposal-optional-chaining',  
    ],  
  }  
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - Webpack

- CRA 1버전 eject
- Webpack3 -> webpack4 마이그레이션
- 웹팩 플러그인, 로더 관리
- 바벨 폴리필, 플러그인 관리
- 타입스크립트 로더

```
// Process TypeScript.
{
  test: /\.tsx?$/,
  exclude: /node_modules/,
  use: [
    {
      loader: require.resolve('cache-loader'),
      options: {
        cacheDirectory: path.resolve('.cache/cache-loader'),
      },
    },
    {
      loader: require.resolve('thread-loader'),
      options: {
        workers: os.cpus().length,
      },
    },
    {
      loader: require.resolve('babel-loader'),
      options: {
        cacheDirectory: true,
        rootMode: 'upward',
      },
    },
    {
      loader: require.resolve('ts-loader'),
      options: {
        configFile: config.tsConfigPath,
        happyPackMode: true,
      },
    },
  ],
},
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

예전 코드 - Webpack

- CRA 1버전 eject
- Webpack3 -> webpack4 마이그레이션
- 웹팩 플러그인, 로더 관리
- 바벨 폴리필, 플러그인 관리
- 타입스크립트 로더
- 스타일 로더

```
{
  test: /\.css$/,
  include: config.cssPaths,
  exclude: /node_modules/,
  use: [
    require.resolve('isomorphic-style-loader'),
    {
      loader: require.resolve('css-loader'),
      options: {
        importLoaders: 1,
        module: true,
        localIdentName: '[path][name]__[local]__[hash:base64:5]',
      },
    },
    {
      loader: require.resolve('postcss-loader'),
      options: {
        ident: 'postcss', // https://webpack.js.org/guides/migrating/#complex-options
        plugins: () => [require('postcss-flexbugs-fixes'), autoprefixer()],
      },
    },
  ],
},
{
  test: /\.scss$/,
  include: config.cssPaths,
  exclude: /node_modules/,
  use: [
    require.resolve('isomorphic-style-loader'),
    {
      loader: require.resolve('css-loader'),
      options: {
        importLoaders: 2,
        module: true,
        localIdentName:
          process.env.NODE_ENV === 'production' ? '[hash:base64:10]' : '[path][name]__[local]__[hash:base64:5]',
      },
    },
    {
      loader: require.resolve('postcss-loader'),
      options: {
        ident: 'postcss', // https://webpack.js.org/guides/migrating/#complex-options
        plugins: () => [require('postcss-flexbugs-fixes'), autoprefixer()],
      },
    },
    require.resolve('sass-loader'),
  ],
},
},
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

etc

- 내장된 Styling(css module, css in js, sass, less, stylus) 지원
- Data flow(redux, mobx, swr, apollo, recoil) 연동 등 210여개 예제
- Preview, Production, Serverless 배포 플랫폼
- 내장된 Typescript 지원
- 정적 파일 서빙
- API Routes
- 절대 경로와 aliases
- Persistent Caching for Page Bundles
- Rewrite, Redirect, Header
- AMP 지원

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Security & community

- Typescript 코드베이스
- 2606 개 통합 테스트, 191개 테스트 스위트
- Google Chrome Team
- React
- Webpack

Security at Vercel

Individuals and companies from every industry place their trust in Vercel.
Our experts ensure that security is top-priority in all of our work.
Not only do we deliver the industry standard, our active OSS efforts help improve it.

Learn More

Report an Incident



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

Security & community

▲ Vercel Develop. Preview. Ship

Next.js in production



2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation


















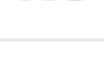

Security

Bundling

Backend / Server logic

And more

RFC

💡 RFC: Incremental Static Regeneration	 6d	🗨️ 248
💡 [RFC] Custom Routes	 33d	🗨️ 206
💡 [RFC] Image Component for Next.js	 9d	🗨️ 77
💡 RFC: i18n Routing	 23h	🗨️ 70
💡 [RFC] Plugins	 14d	🗨️ 71
💡 RFC: Returning redirects from <code>getServerSideProps</code> / <code>getStaticProps</code>	 2d	🗨️ 55
💡 RFC: Image Optimization	 13d	🗨️ 25
🎓 i18n for Serverless Next.js app	 10d	🗨️ 94
🎓 [RFC] <code>.env</code> environment support	 64d	🗨️ 81 
🎓 <code>getStaticProps</code> on <code>_app</code>	 2d	🗨️ 70 
💡 [RFC] - Module/nomodule for Next.js	 83d	🗨️ 23
🎓 How to use <code>getServerSideProps</code> in a HOC for authentication	 17d	🗨️ 50
🎓 Return redirect from <code>getServerSideProps</code>	 43d	🗨️ 42 
💡 [RFC] Deprecate <code>'next/head'</code>	 29d	🗨️ 14
💡 [RFC] ESLint out of the box	 3d	🗨️ 10
💡 RFC: Importing CSS from Third Party React Components	 26d	🗨️ 37

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

RFC

RFC: Image Optimization #17141

 timneutkens  Ideas  26d ago  25 replies

- 현재는 외부 플러그인을 통해서 가능
- WebP, AVIF 같은 최신 형식으로 자동 변환
- 이미지 사이즈 자동 조절
- 로컬호스팅, CDN 이미지 경로 지원
- Preconnect, Preload (experimental)
- 적은 설치 시간

(Imagemin, svg2png와 같은 종속성이 복잡한 라이브러리를 사용하지 않고 WASM로 컴파일된 라이브러리 사용)

[RFC] Image Component for Next.js #16832

 atcastle  Ideas  39d ago  78 replies

We can serve an optimized image like so:

```
import Image from 'next/image'

function Home() {
  return (
    <>
      <h1>My Homepage</h1>
      <Image src="/me.png" width={200} height={200} />
      <p>Welcome to my homepage!</p>
    </>
  )
}

export default Home
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

RFC

RFC: Returning redirects from getServerSideProps / getStaticProps #14890

timneutkens Ideas 98d ago 56 replies

```
res.writeHead(301, {
  Location: 'new/url/destination/here'
});
res.end();
```

```
module.exports = {
  async redirects() {
    return [
      {
        source: '/about',
        destination: '/',
        permanent: true,
      },
    ],
  },
}
```

- An example of how this would look:

```
export async function getServerSideProps({params}) {
  return Hello World
}

export async function getServerSideProps({params}) {
  const item = await getData(params.id)
  if(!item) {
    return {
      redirect: {
        permanent: false,
        destination: '/somewhere-else'
      }
    }
  }
}

return {
  props: {
    item
  }
}
```

- When using static generation with the `fallback: true` behavior you might also want to redirect to another page when the page does not exists, hence why redirect might be useful there also:

```
export async function getStaticProps({params}) {
  return Hello World
}

export async function getStaticProps({params}) {
  const item = await getData(params.id)
  if(!item) {
    return {
      redirect: {
        permanent: false,
        destination: '/somewhere-else'
      }
    }
  }
}

return {
  props: {
    item
  }
}
```

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

RFC

[RFC] Plugins #9133

 timneutkens  Ideas  361d ago  71 replies



timneutkens Maintainer

Authored together with @ijjk.

Goals

- Provide an official plugin system for Next.js
- Allow hooking into different parts of the Next.js lifecycle
- Provide the right limits for plugins to prevent them from de-optimizing applications
- Warn users when a plugin is being inefficient
- Provide a zero-config approach to plugins
- Provide code splitting for plugins

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

And more

RFC

RFC: I18n Routing #17078

 timneutkens  Ideas  29d ago  79 replies



timneutkens

Maintainer

edited ▾ 29d ...

Goals

- Provide i18n libraries with the supported locales / the default locale / the current locale
- Provide different strategies as every application has different requirements
 - Subpath holding the locale: e.g. `/[locale]/blog/hello-world`
 - Different domains holding the region: e.g. `example.com`, `example.nl` (different TLD)
- Provide "language affinity", ensuring a certain user is redirected or shown their selected language once they access it for the first time
 - This is generally only done for the `/` route which then redirects to the previously selected language
 - This can be optional per-application
- Provide language detection
 - Leverage the `Accept-Language` header
 - Support setting a cookie that forces the language detection route to a certain locale, which is useful for cases where you have a language switcher and want to memorize the user's chosen language when they go back to the `/` route
- Support the hybrid server-side rendering and static generation use case

2.3 Web Framework 구성요소

UI Component Model

Pre-rendering HTML

Routing

CSS Encapsulation

Data Flow

Deployment / Releases

Compilation

Security

Bundling

Backend / Server logic

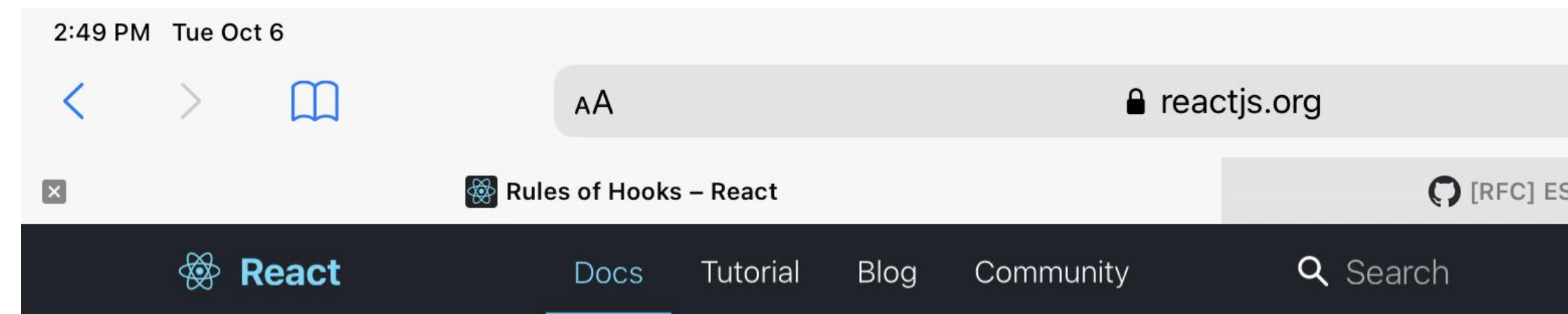
And more

RFC

[RFC] ESLint out of the box #17320

 prateekbh  Ideas  20d ago  15 replies

- Eslint-loader를 webpack 컴파일에 추가
- 👍 쉬운 구성
- 🗨️ 개발 시간 증가



Rules of Hooks

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class.

Hooks are JavaScript functions, but you need to follow two rules when using them. We provide a [linter plugin](#) to enforce these rules automatically:

• 02

• **03 Troubleshooting
& Tips**

3.1 Troubleshooting

SSR

server-side rendering

SSG

static site generation

Desktop browser screenshot showing SSR performance metrics. The Lighthouse score is 97. Metrics include: Performance (97), Accessibility (89), Best Practices (93), SEO (73), and Progressive Web App (PWA). Key metrics for Performance: First Contentful Paint (0.8 s), Time to Interactive (1.4 s), Speed Index (0.8 s), Total Blocking Time (0 ms), Largest Contentful Paint (1.1 s), and Cumulative Layout Shift (0.001). Opportunities include: Serve images in next-gen formats (0.24 s savings) and Remove unused JavaScript (0.16 s savings).

Desktop browser screenshot showing SSG performance metrics. The Lighthouse score is 99. Metrics include: Performance (99), Accessibility (97), Best Practices (100), SEO (80), and Progressive Web App (PWA). Key metrics for Performance: First Contentful Paint (0.6 s), Time to Interactive (1.1 s), Speed Index (0.6 s), Total Blocking Time (0 ms), Largest Contentful Paint (0.9 s), and Cumulative Layout Shift (0). Opportunities include: Serve images in next-gen formats (0.24 s savings) and Remove unused JavaScript (0.16 s savings).

Mobile browser screenshot showing SSR performance metrics. The Lighthouse score is 69. Metrics include: Performance (69), Accessibility (88), Best Practices (86), SEO (75), and Progressive Web App (PWA). Key metrics for Performance: First Contentful Paint (2.7 s), Time to Interactive (7.4 s), Speed Index (4.2 s), Total Blocking Time (150 ms), Largest Contentful Paint (4.8 s), and Cumulative Layout Shift (0.071). Opportunities include: Serve images in next-gen formats (2.1 s savings), Remove unused JavaScript (1.65 s savings), and Defer offscreen images (0.75 s savings).

Mobile browser screenshot showing SSG performance metrics. The Lighthouse score is 94. Metrics include: Performance (94), Accessibility (88), Best Practices (93), SEO (64), and Progressive Web App (PWA). Key metrics for Performance: First Contentful Paint (2.0 s), Time to Interactive (3.9 s), Speed Index (2.0 s), Total Blocking Time (220 ms), Largest Contentful Paint (2.6 s), and Cumulative Layout Shift (0). Opportunities include: Serve images in next-gen formats (2.1 s savings), Remove unused JavaScript (1.65 s savings), and Defer offscreen images (0.75 s savings).

3.1 Troubleshooting

1. Query기반 라우팅 불가


```
router.get('/category/category', async (ctx) => {
  const {
    query: { catId, cat_id },
  } = ctx

  const categoryId = Number(catId || cat_id || 0)

  if (categoryId > 50000010) {
    await app.render(ctx.req, ctx.res, `/search/category`, ctx.query)
  } else {
    ctx.redirect(`/category/category/${categoryId}`)
    return
  }
  ctx.respond = false
})
```

[RFC] Custom Routes #9081

 Timer  Ideas  360d ago  206 replies

 Timer Maintainer

Goals

- Support Redirects
- Support Rewrites
- Support Headers
- Provide an API for `next-routes` and related libraries (and eventual migration path)
- Reduce the necessity of a custom server

3.1 Troubleshooting

2. Custom app에서 getInitialProps만 사용가능
 - automatic static optimization X
 - Redux 사용시 모든 페이지의 reducer가 app.js에 번들됨
 - > code splitting 필요

3.1 Troubleshooting

3. 전역 css import가 _app.js에서만 가능함

```
import '../styles.css'  
  
// This default export is required in a new `pages/_app.js` file.  
export default function MyApp({ Component, pageProps }) {  
  return <Component {...pageProps} />  
}
```

These styles (`styles.css`) will apply to all pages and components in your application. Due to the global nature of stylesheets, and to avoid conflicts, you may **only import them inside `pages/_app.js`**.

Import styles from `node_modules`

Since Next.js **9.5.4**, importing a CSS file from `node_modules` is permitted anywhere in your application.


<https://github.com/vercel/next.js/blob/canary/docs/basic-features/built-in-css-support.md#adding-a-global-stylesheet>

3.1 Troubleshooting

4. Typescript barrel files에서 tree shaking 안됨

```
"sideEffects": false,
```

<https://github.com/vercel/next.js/issues/12557>

 **Tree shaking doesn't work with Typescript barrel files #12557**
majelbstoat opened this issue on 6 May · 6 comments

Describe the bug

When using a [barrel file](#) to re-export components from a single location, tree-shaking does not function correctly.

To Reproduce

I'm using Next 9.3.6 and I've arranged my components like:

```
components/  
  Header/  
    Header.tsx  
  Sidebar/  
    Sidebar.tsx  
  index.ts
```

Each component file exports a single component, like this:

```
export { Header }
```

index.ts is a barrel file that re-exports from each individual component file:

```
export * from './Header/Header.tsx'  
export * from './Sidebar/Sidebar.tsx'  
// ...
```

I then use a couple of components in _app.tsx like:

```
import { Header, Sidebar } from "../components"
```

3.1 Troubleshooting

5. Next 9.5.3  css preload cors

Next 9.5.2  FOUC with css module using dynamic/import

3.1 Troubleshooting

6. Shallow: true 옵션으로 페이지 이동 후 뒤로가기 시에 다시 데이터를 패칭하지 않음.

```
this.props.router.beforePopState(({ as, options }) => {  
  if (options.shallow) {  
    this.props.router.replace(as)  
    return false  
  }  
  return true  
})
```

3.1 Troubleshooting

7. node_modules, monorepo의 local packages 경우 transpile하지 않음
- next-transpile-modules를 사용해야함

```
const withTM = require('next-transpile-modules')([
  '@search/common-utils',
  '@search/client-utils',
  '@search/server-utils',
  '@search/types',
  'query-string',
  'decode-uri-component',
  'split-on-first',
  'strict-uri-encode',
])
```

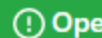
정형식 [쇼핑검색개발], 3 months ago • [MOBILE]

3.1 Troubleshooting

8. IE11에서 개발서버가 실행이 안됨 (프로덕션에서는 됨)

-> ShadowDOM polyfill 추가

IE11, development, HMR: Object doesn't support property or method 'attachShadow'. #13231

 Open 15 comments · May be fixed by #17418

```
if (typeof window !== 'undefined') {  
  require('@webcomponents/shadydom')  
}
```

3.1 Troubleshooting

9. react-router와 다르게 history pushState기능이 없음
-> query를 사용하거나 브라우저 Storage를 사용함

Possible to pass state to next/router? #771

Open 34 comments



bryan on 15 Jan 2017

With react-router I was able to do:

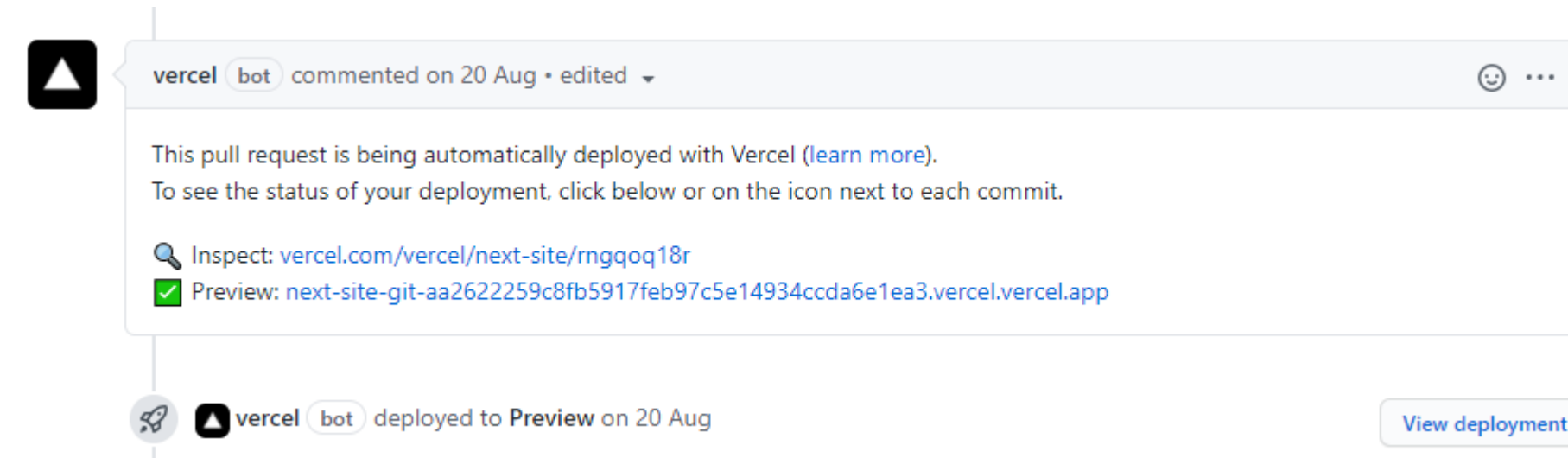
```
browserHistory.push({ pathname: "/", state: { message: 'This is a message forwarded in a state.' } })
```

Is it possible to do something like this using next/router or a way to redirect the user and also pass some value over to the new page?

17

3 .2 Custom Server

- 비추천
- HMR을 .next를 제외하고 새로 구성해야함.
- Vercel로 deploy 불가 (pr-preview)



3 .2 Custom Server

- 비추천
- HMR을 .next를 제외하고 새로 구성해야함.
- Vercel로 deploy 불가 (pr-preview)
- serverless functions, Automatic Static Optimization도 불가능
- 개발서버 띄우는 시간도 70~80% 증가

커스텀 서버

- 실행 시간 : **65초**

```
cross-env NODE_ENV=development ts-node-dev src/server/index.ts
[2020-10-17T14:20:51.021] [DEBUG] - Kafka Producer has been connected...
...
[2020-10-17T14:21:56.410] [ INFO] - 작업수행 시간: 390
```

next, api 서버 분리시

- 실행 시간 : **36초**

```
cross-env NODE_ENV=development concurrently "next dev" "ts-node-dev src/server/index.ts"
[2020-10-17T14:19:18.436] [DEBUG] - Kafka Producer has been connected...
...
[2020-10-17T14:19:54.799] [ INFO] - 작업수행 시간: 551
```

3 .3 next.config.js

- 트랜스파일 되지 않음 (Node.js 버전에 맞는 구문)

- Phase

```
const { PHASE_DEVELOPMENT_SERVER } = require('next/constants')

module.exports = (phase, { defaultConfig }) => {
  if (phase === PHASE_DEVELOPMENT_SERVER) {
    return {
      /* development only config options here */
    }
  }
}
```

```
export const PHASE_EXPORT = 'phase-export'
export const PHASE_PRODUCTION_BUILD = 'phase-production-build'
export const PHASE_PRODUCTION_SERVER = 'phase-production-server'
export const PHASE_DEVELOPMENT_SERVER = 'phase-development-server'
```

- Webpack

- server, client 2번 실행

```
module.exports = {
  webpack: (config, { buildId, dev, isServer, defaultLoaders, webpack }) => {
    // Note: we provide webpack above so you should not `require` it
    // Perform customizations to webpack config
    config.plugins.push(new webpack.IgnorePlugin(/\/__tests__\/))

    // Important: return the modified config
    return config
  },
}
```

3 .3 next.config.js

- 실험기능

- Modern : 트랜스파일 차등 로딩
- Profiling : 프로덕션에서 react profiling
- productionBrowserSourceMaps : 소스맵 생성
- scrollRestoration : 내부 링크에서만 가능
- optimizeImages, Fonts : preload, preconnect

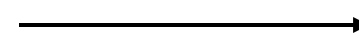
```
experimental: {  
  cpu: Math.max(  
    1,  
    (Number(process.env.CIRCLE_NODE_TOTAL) ||  
     (os.cpus() || { length: 1 }).length) - 1  
  ),  
  modern: false,  
  plugins: false,  
  profiling: false,  
  sriFlushToDisk: true,  
  reactMode: 'legacy',  
  workerThreads: false,  
  pageEnv: false,  
  productionBrowserSourceMaps: false,  
  optimizeFonts: false,  
  optimizeImages: false,  
  scrollRestoration: false,  
  i18n: false,  
}
```

3.4 plugins

- next-compose-plugins

```
// next.config.js
const withSass = require('@zeit/next-sass');
const withTypescript = require('@zeit/next-typescript');
const withImages = require('next-images');
const withOffline = require('next-offline');

module.exports = withSass(withOffline(withTypescript(withImages({
  {
    cssModules: true,
    cssLoaderOptions: {
      importLoaders: 1,
      localIdentName: '[local]__[hash:base64:5]',
    },
    typescriptLoaderOptions: {
      transpileOnly: false,
    },
    useFileSystemPublicRoutes: false,
    distDir: 'build',
    workerName: 'sw.js',
    imageTypes: ['jpg', 'png'],
  }
})))));
```



```
// next.config.js
const withPlugins = require('next-compose-plugins');

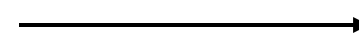
module.exports = withPlugins([...plugins], nextConfiguration);
```

3.4 plugins

- next-compose-plugins

```
// next.config.js
const withSass = require('@zeit/next-sass');
const withTypescript = require('@zeit/next-typescript');
const withImages = require('next-images');
const withOffline = require('next-offline');

module.exports = withSass(withOffline(withTypescript(withImages({
  {
    cssModules: true,
    cssLoaderOptions: {
      importLoaders: 1,
      localIdentName: '[local]__[hash:base64:5]',
    },
    typescriptLoaderOptions: {
      transpileOnly: false,
    },
    useFileSystemPublicRoutes: false,
    distDir: 'build',
    workerName: 'sw.js',
    imageTypes: ['jpg', 'png'],
  }
})))));
```



```
// next.config.js
const withPlugins = require('next-compose-plugins');

module.exports = withPlugins([...plugins], nextConfiguration);
```

```
const { withPlugins, optional } = require('next-compose-plugins');
const { PHASE_DEVELOPMENT_SERVER } = require('next/constants');

module.exports = withPlugins([
  optional(() => require('@zeit/next-sass')), { /* optional configuration */ }, [PHASE_DEVELOPMENT_SERVER]],
]);
```

3.4 plugins

- **next-optimized-images**

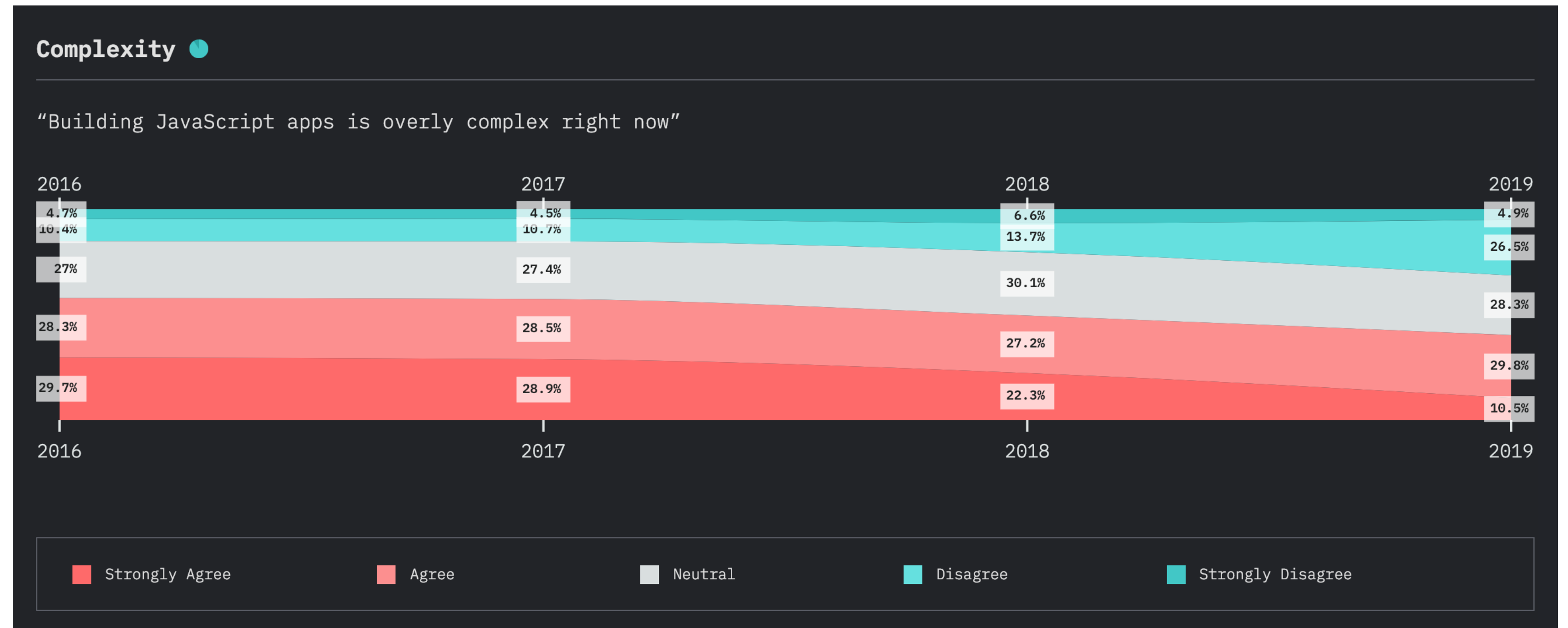
- nextjs 내장된 이미지 최적화 기능 없음. ([TODO](#))
- gatsby-image의 nextJS 버전
- 이미지 url에 쿼리를 추가하여 사용
(v3부터는 image Component로 처리 가능)

```
module.exports = withPlugins([
  [optimizedImages, {
    // these are the default values so you don't have to provide them if they are good enough for your use-case
    // but you can overwrite them here with any valid value you want.
    inlineImageLimit: 8192,
    imagesFolder: 'images',
    imagesName: '[name]-[hash].[ext]',
    handleImages: ['jpeg', 'png', 'svg', 'webp', 'gif'],
    removeOriginalExtension: false,
    optimizeImages: true,
    optimizeImagesInDev: false,
    mozjpeg: {
      quality: 80,
    },
    optipng: {
      optimizationLevel: 3,
    },
    pngquant: false,
    gifsicle: {
      interlaced: true,
      optimizationLevel: 3,
    },
    svgo: {
      // enable/disable svgo plugins here
    },
    webp: {
      preset: 'default',
      quality: 75,
    },
  }],
]);
```


- 리액트 개발은 여전히 쉽지 않음



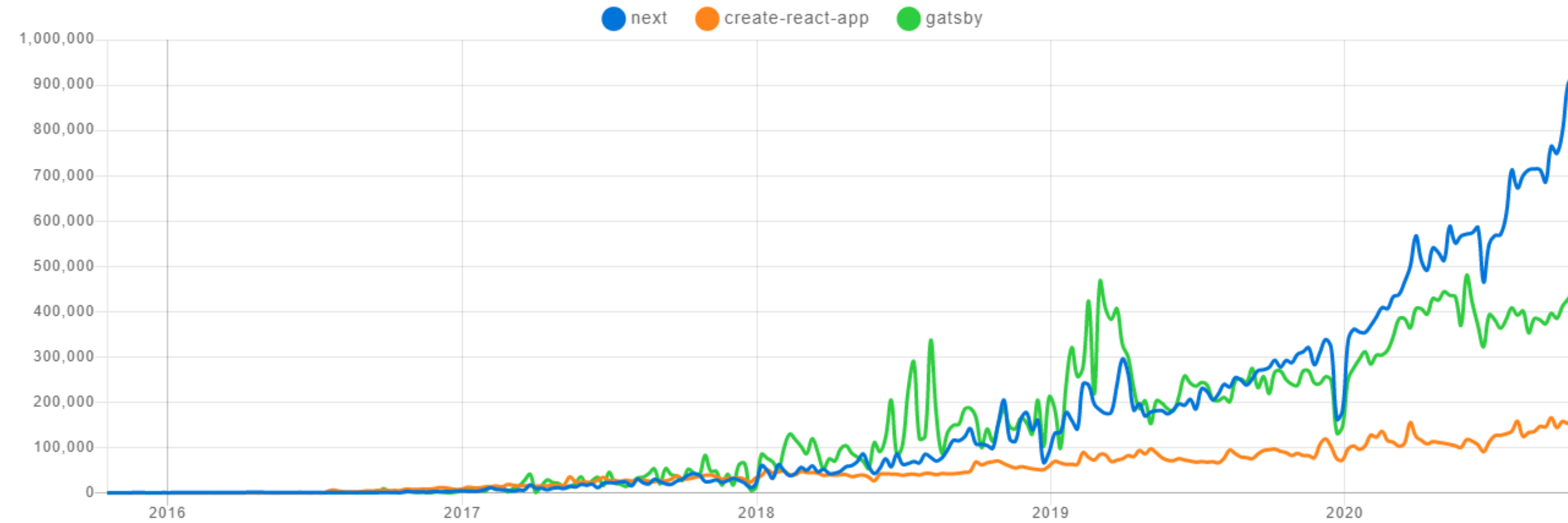
- 리액트 개발은 여전히 쉽지 않음
- 점점 복잡성을 줄이기 위해 노력



- 리액트 개발은 여전히 쉽지 않음
- 점점 복잡성을 줄이기 위해 노력
- 리액트 프레임워크 Next.js


next x create-react-app x gatsby x + nuxt + react-static + vuepress + razzle + koa

Downloads in past 5 Years



Stats

	stars 🌟	issues ⚠️	updated ✖️	created 🕒	size 📦
next	54,550	724	Oct 20, 2020	Oct 6, 2016	bundlephobia 403
create-react-app	82,836	715	Oct 20, 2020	Jul 17, 2016	bundlephobia timeout
gatsby	47,513	650	Oct 20, 2020	May 22, 2015	bundlephobia timeout



Q & A