

# Next.js 심화 3

## 1 Next.js

## Routing

(<https://nextjs.org/docs/routing/introduction>)

pages file system

Link client side route(viewport 에 들어오면 SSG로 prefetching)

`href={` /blog/${encodeURIComponent(post.slug)} `}`

`href={ { pathname: '/blog/[slug]', query: {slug: post.slug}, } }`

## Dynamic Routes

(<https://nextjs.org/docs/routing/dynamic-routes>)

[pid] 로 다이내믹한 url을 생성할 수 있게 한다.

router.query로 key와 value를 꺼낼 수 있다.

catch all routes with [...slug] 하면 query의 key slug에 배열로 값들이 존재

optional catch all routes [...slug] slug 없는 경우도 대응 가능

## Caveats(주의사항)

pages match 정적인게 먼저 매칭된다.

server side 코드 없이 빌드된 페이지에서 router.query는 처음에는 빈 객체다.

## Imperatively

(<https://nextjs.org/docs/routing/imperatively>)

next/link 없이 router에 직접 페이지를 푸시하는 방법

```
router.push('/about')
```

## Shallow Routing

(<https://nextjs.org/docs/routing/shallow-routing>)

데이터 페칭없이 browser 에 보이는 url만 변경

```
router.push('/?counter=10', undefined, {shallow: true})
```

다른 url 에서 하면 page가 변경됨

## API Routes

(<https://nextjs.org/docs/api-routes/introduction>)

pages/api/\* 는 페이지가 아닌 api 처럼 동작  
client side 코드에는 추가되지 않는다.

handler에 콘솔 추가 하고 / useEffect로 fetch 해보기

## Dynamic API Routes

(<https://nextjs.org/docs/api-routes/dynamic-api-routes>)

dynamic의 동작은 동일



## API Middelwares

(<https://nextjs.org/docs/api-routes/api-middlewares>)

api config 추가 혹은 다양한 middlewares 추가 가능

(<https://github.com/vercel/next.js/tree/canary/examples/api-routes-middleware>)

cookies 예시

## Response Helpers

(<https://nextjs.org/docs/api-routes/response-helpers>)

response 에 담을 수 있는 데이터

redirect 예시 /api/hello -> /api/bye with 307

## 정리

### Next.js 심화 3

Next.js의 라우팅

pages 폴더의 file system 활용

Dynamic Routes

[slug] / [...slug] / [[...slug]]

Router

직접 라우터를 조작 가능 push / shallow

API Routes

API 서버로의 동작 middlewares