

# 실전으로 배우는 웹 성능 최적화 for React

Lecture #1

# 실습내용

- 이미지 사이즈 최적화
- Code Split
- 텍스트 압축
- Bottleneck 코드 최적화

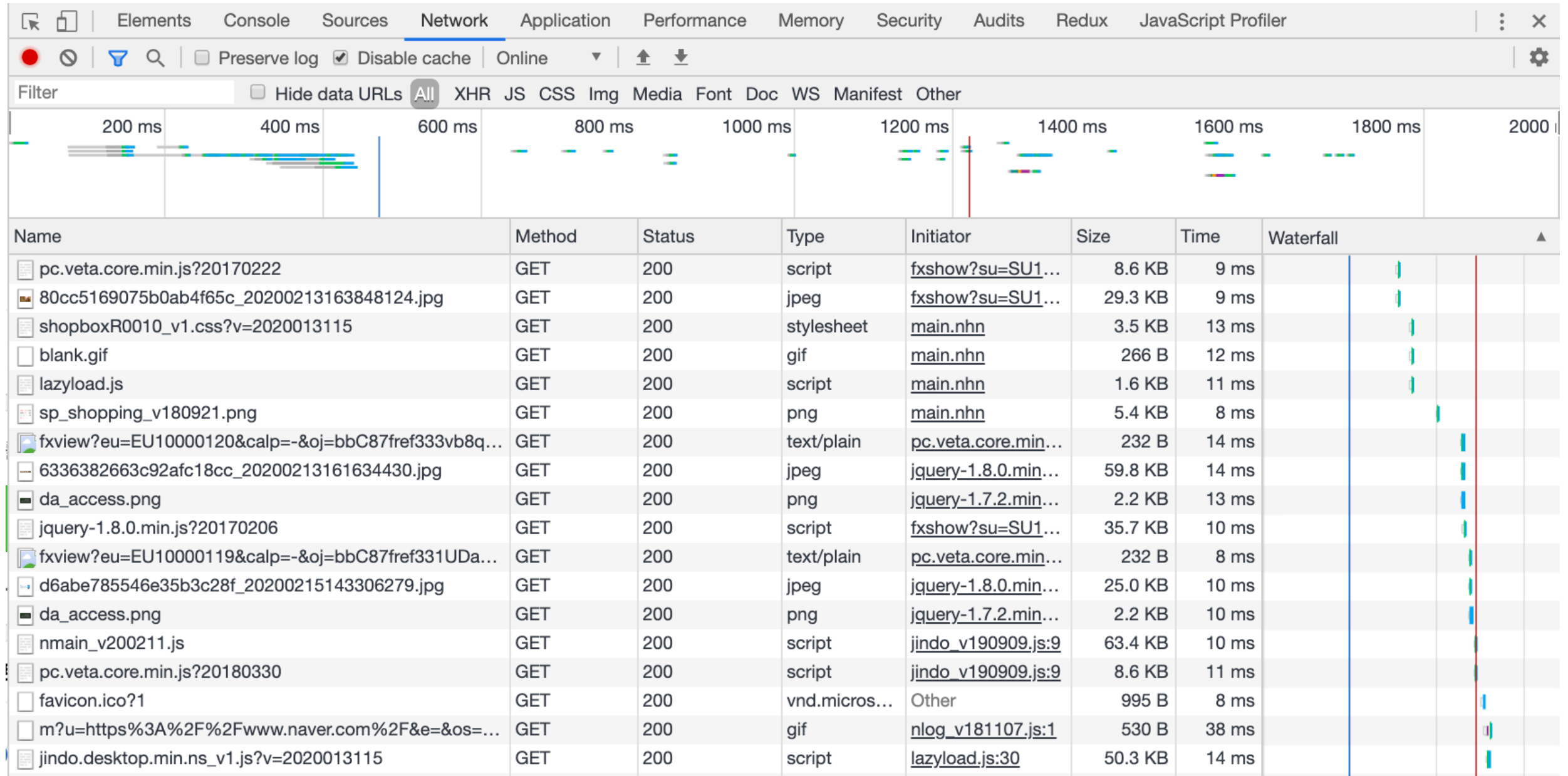
# 실습내용

- 이미지 사이즈 최적화
  - Code Split
  - 텍스트 압축
- 로딩 성능 최적화
- Bottleneck 코드 최적화 — 렌더링 성능 최적화

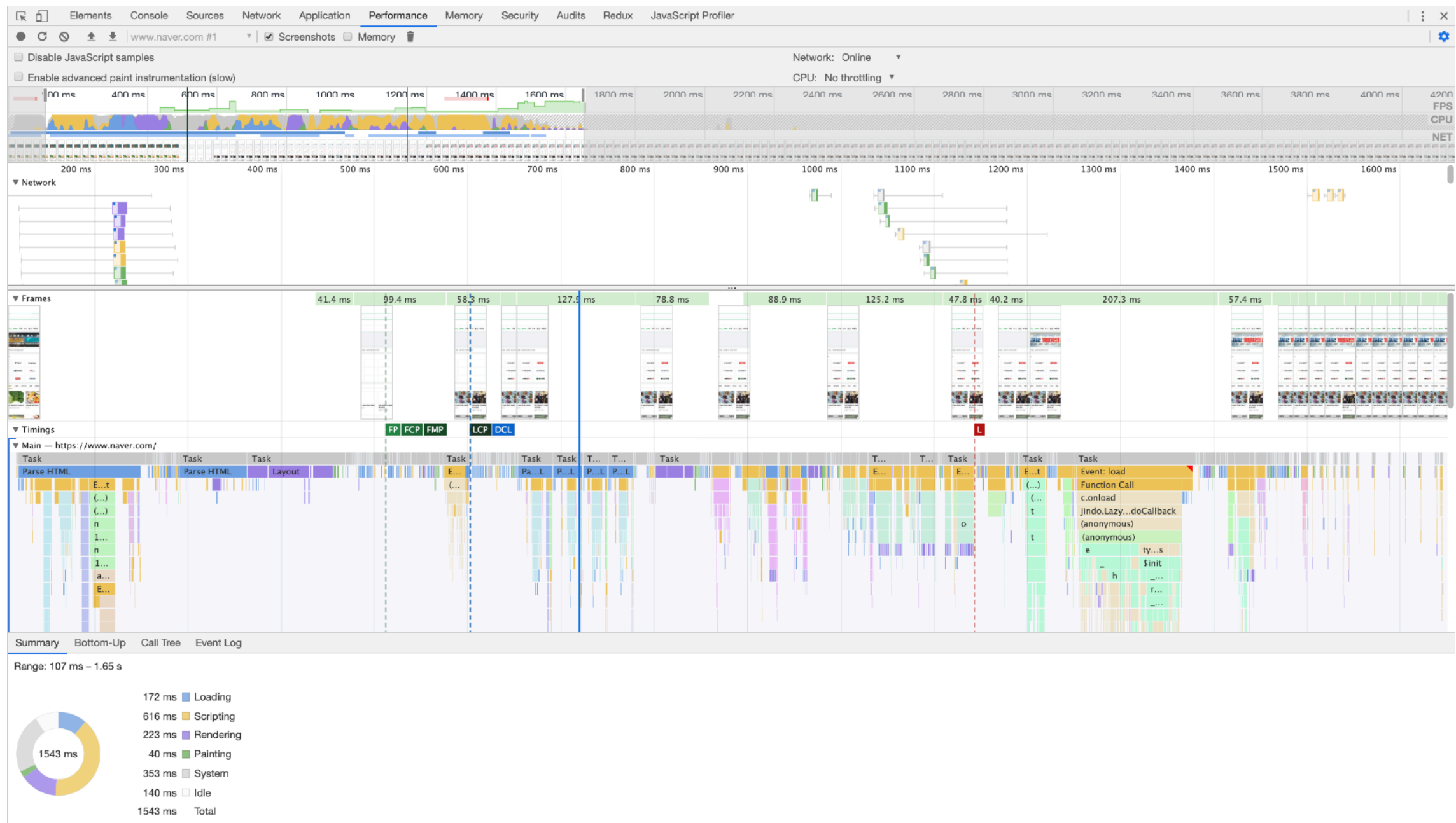
# 분석 툴

- 크롬 Network 탭
- 크롬 Performance 탭
- 크롬 Audit 탭 (Light house)
- webpack-bundle-analyzer

# 크롬 Network 탭



# 크롬 Performance 탭



# 크롬 Audit 탭 (Light house)

54

## Performance

### Metrics

■ First Contentful Paint	3.9 s	■ First Meaningful Paint	3.9 s
■ Speed Index	4.0 s	■ First CPU Idle	6.1 s
▲ Time to Interactive	9.2 s	▲ Max Potential First Input Delay	290 ms

[View Trace](#)

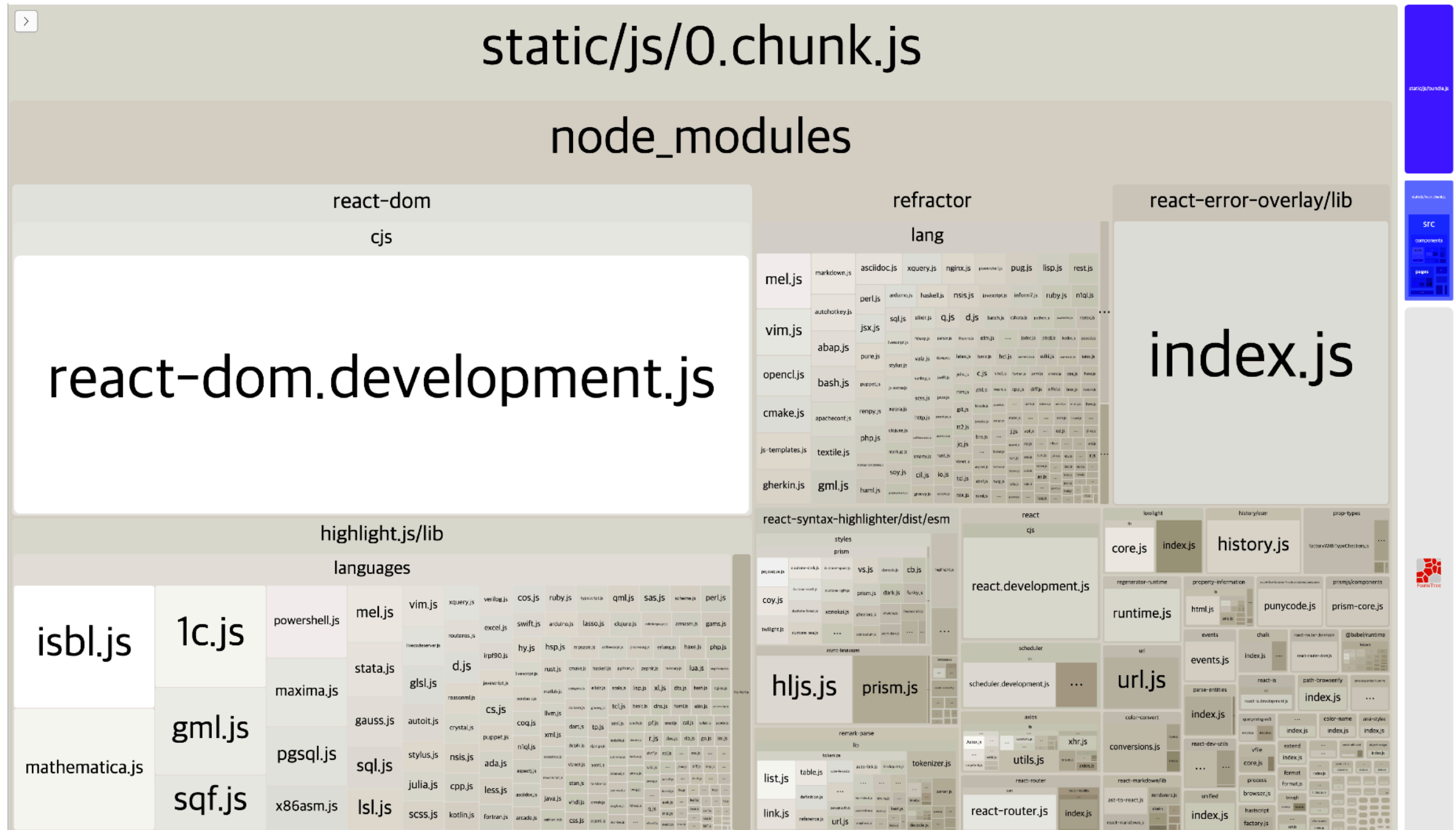
Values are estimated and may vary. The performance score is based only on these metrics.



**Opportunities** — These suggestions can help your page load faster. They don't directly affect the Performance score.

Opportunity	Estimated Savings
▲ Serve images in next-gen formats	4.69 s
▲ Defer offscreen images	2.25 s

# webpack-bundle-analyzer





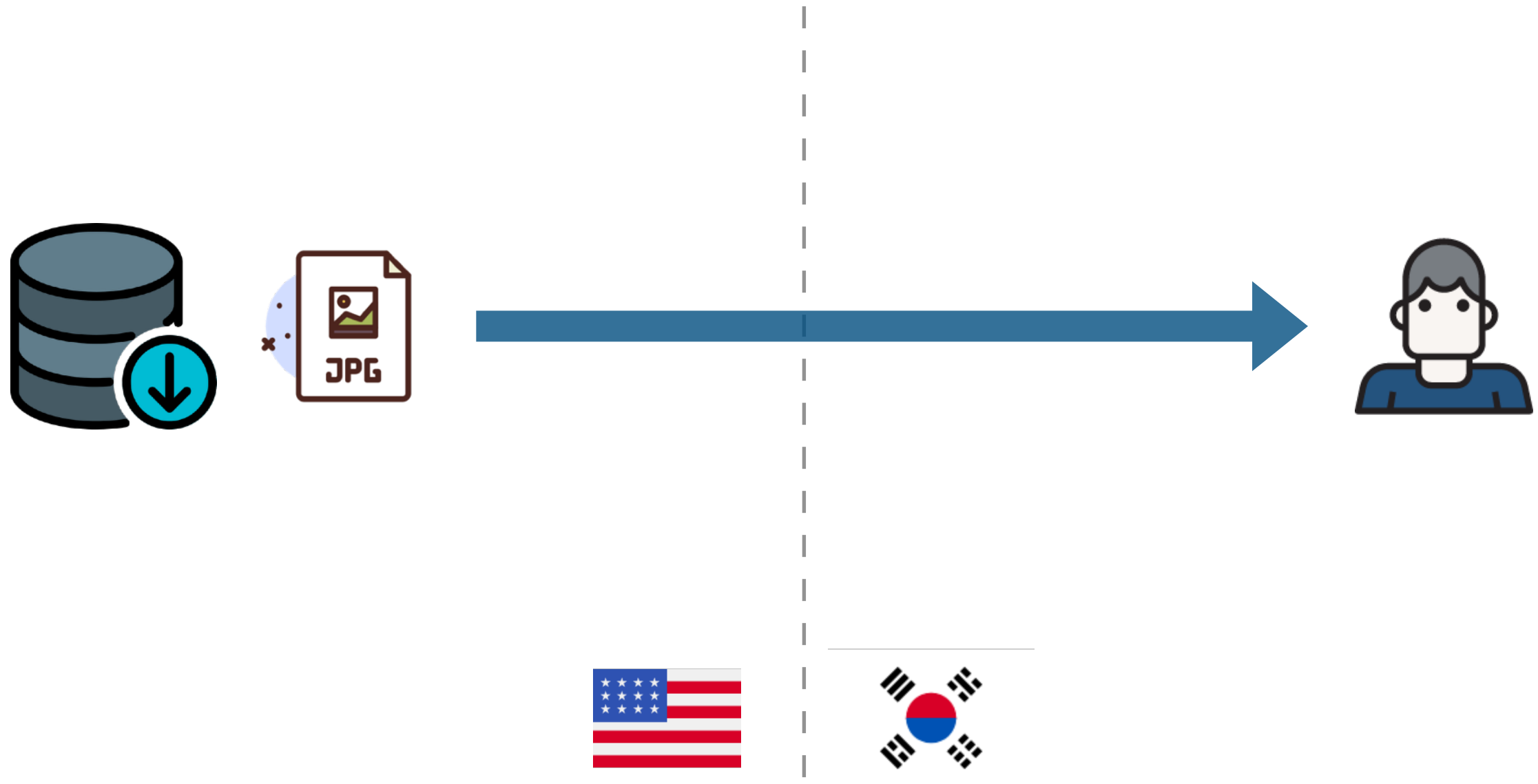
# CDN

Contents Delivery Network

물리적 거리의 한계를 극복하기 위해  
소비자(사용자)와 가까운 곳에  
컨텐츠 서버를 두는 기술

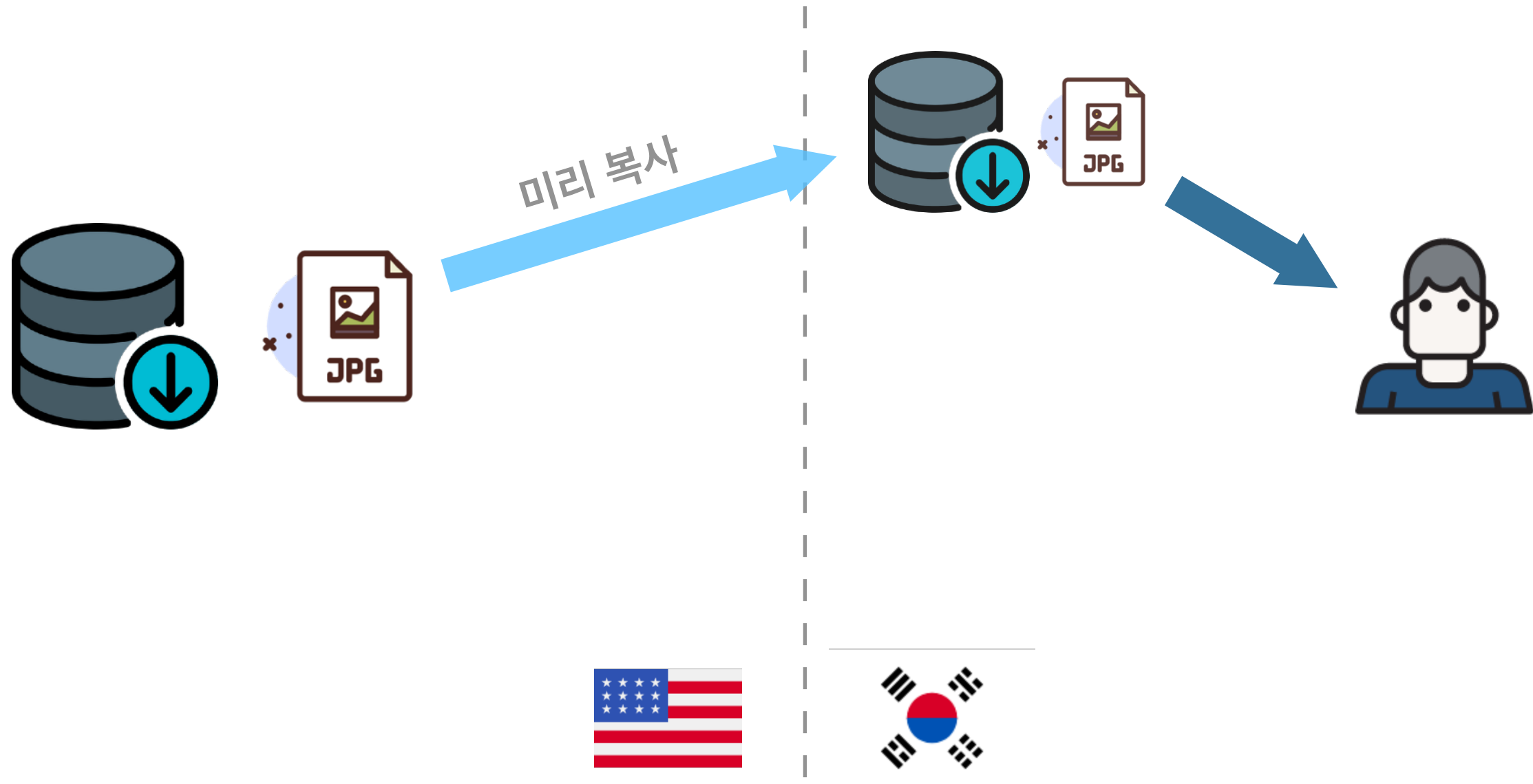
# CDN

Contents Delivery Network



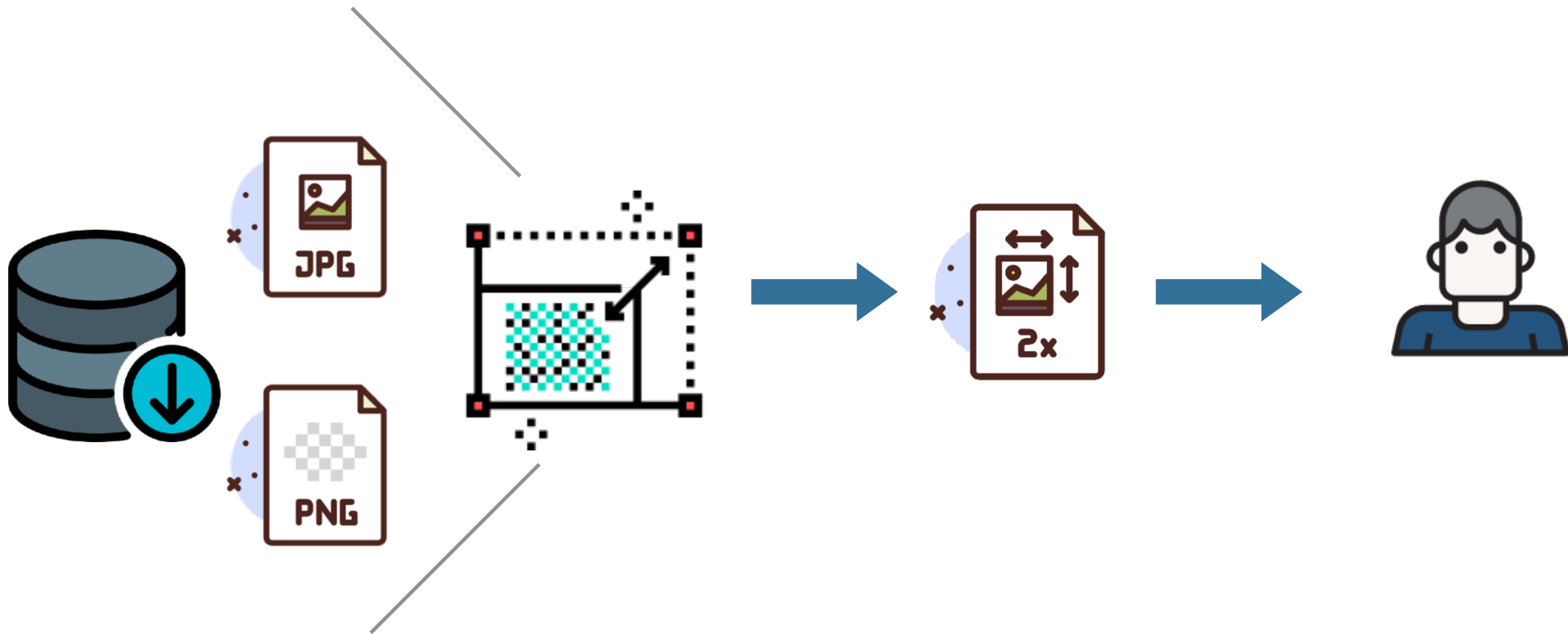
# CDN

Contents Delivery Network



# Image CDN

image processing CDN



# Image CDN

image processing CDN

`http://cdn.image.com?src=[img src]&width=200&height=100`

# Bottleneck 해결방안

- 특수 문자를 효율적으로 제거하기

- 작업하는 양 줄이기

# Bottleneck 해결방안

- 특수 문자를 효율적으로 제거하기
  - ✓ replace 함수와 정규식 사용
  - ✓ 마크다운의 특수문자를 지워주는 라이브러리 사용 (remove-markdown)
- 작업하는 양 줄이기

# Bottleneck 해결방안

- 특수 문자를 효율적으로 제거하기

- ✓ replace 함수와 정규식 사용

- ✓ 마크다운의 특수문자를 지워주는 라이브러리 사용 (remove-markdown)

- 작업하는 양 줄이기

대략 200자

```
This tutorial doesn't assume any existing React knowledge.\n\n##  
Before We Start the Tutorial {#before-we-start-the-tutorial}\n\nWe  
will build a small game during this tutorial. **You might be tempted  
to skip it because you're not building games -- but give it a  
chance.** The techniques you'll learn in the tutorial are fundamental  
to building any React app, and mastering it will give you a deep  
understanding of React.\n\n>\n>This tutorial is designed for  
people who prefer to **learn by using**.
```

최장 90021자...



# Code Splitting

코드를 분할하는 것

# Code Splitting

**Bundle.js**

**ListPage**

**axios**

**ViewPage**

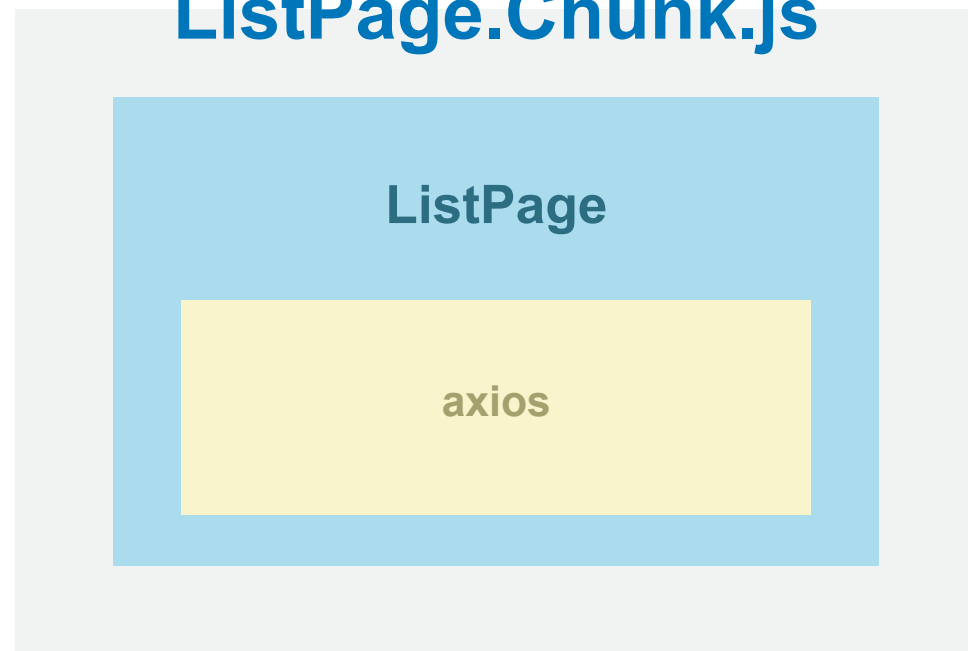
**react-markdown**

**react-syntax-highlighter**

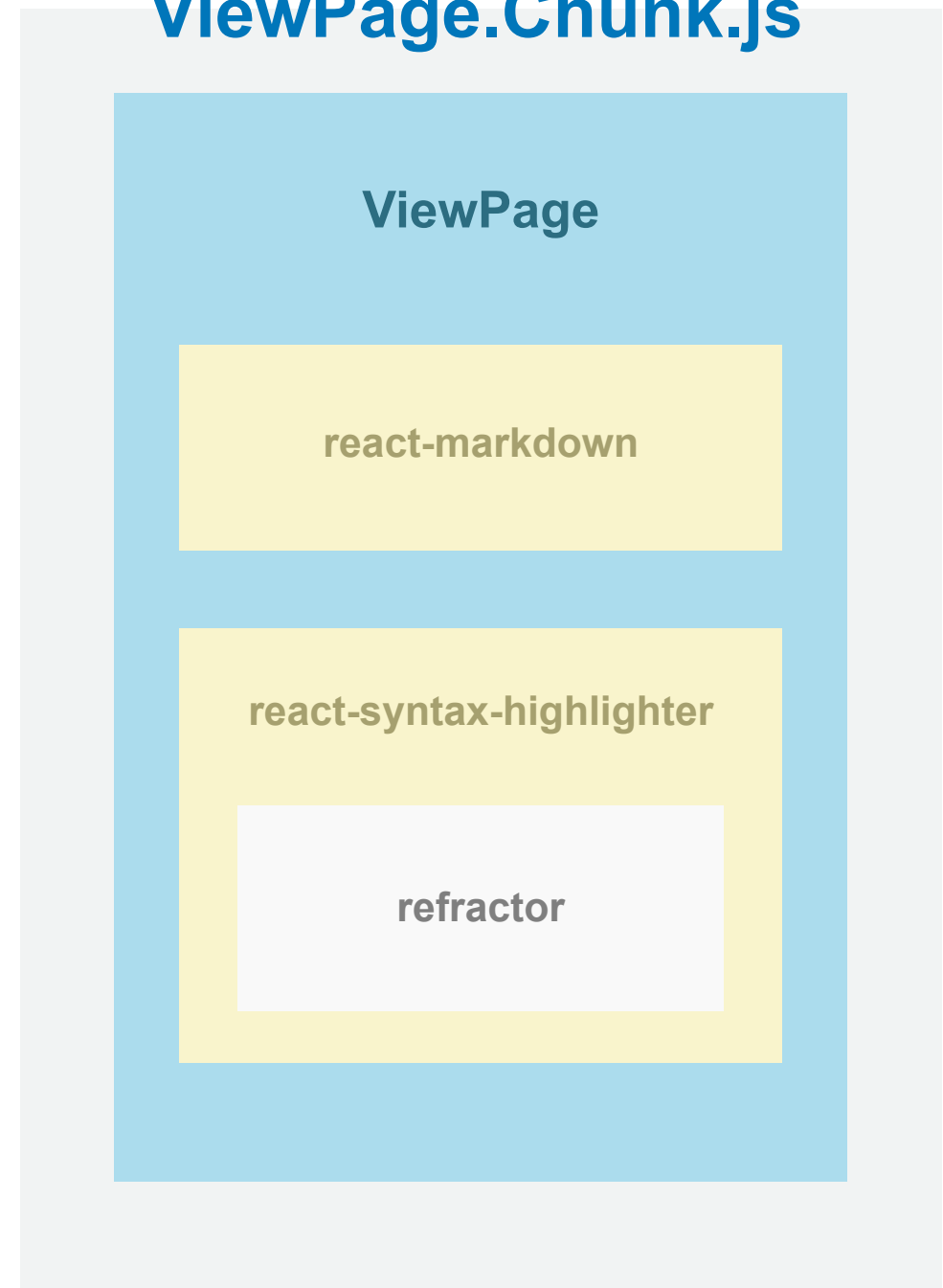
**refractor**

# Code Splitting

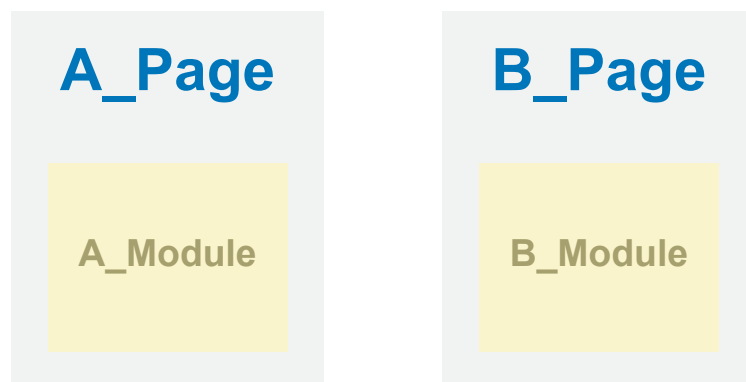
## ListPage.Chunk.js



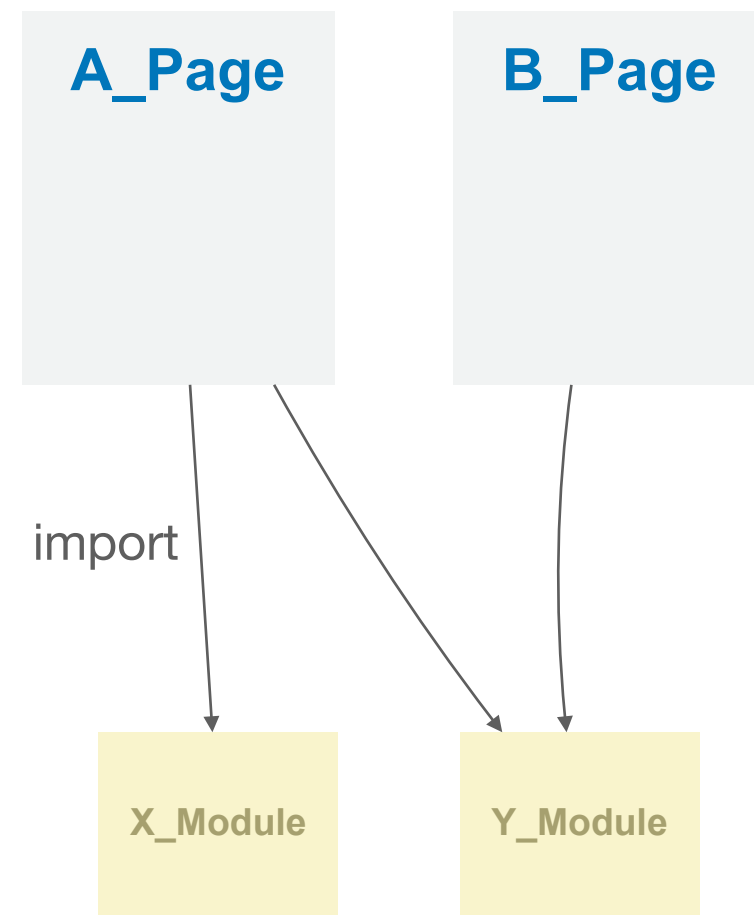
## ViewPage.Chunk.js



# Code Splitting



페이지 별로



모듈 별로

# Code Splitting

불필요한 코드 또는 중복되는 코드가 없이  
적절한 사이즈의 코드가  
적절한 타이밍에  
로드될 수 있도록 하는 것

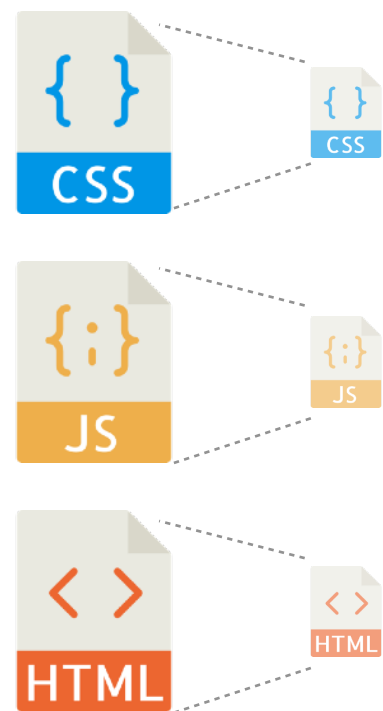
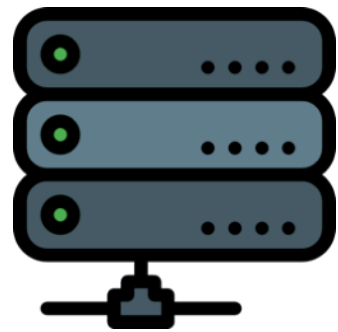
# 텍스트 압축

Text Compression

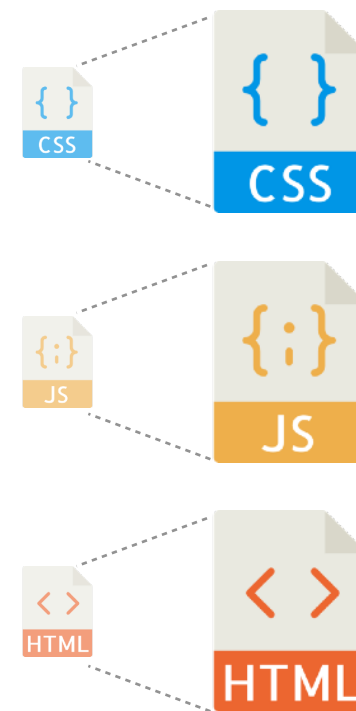


# 텍스트 압축

Text Compression



압축



압축 풀기



# 텍스트 압축

Text Compression

GZIP

Deflate



# 정리

**Audits** 툴을 이용한 페이지 감사



**이미지 사이즈 최적화**



**Bottleneck 코드 최적화**



**Code Splitting**



**텍스트 압축**

# 정리

Audits 툴을 이용한 페이지 감사



이미지 사이즈 최적화



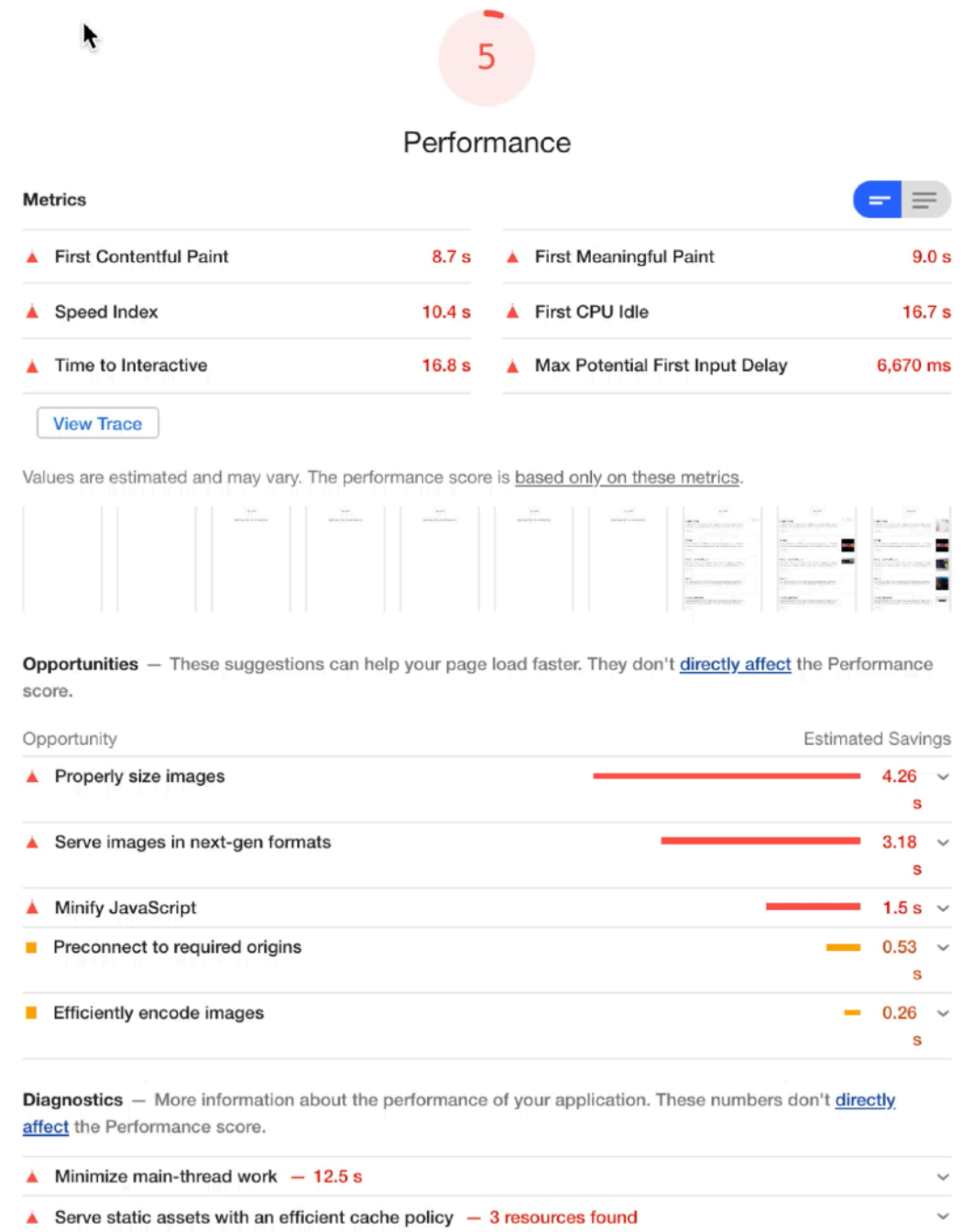
Bottleneck 코드 최적화



Code Splitting



텍스트 압축



# 정리

Audits 툴을 이용한 페이지 감사

↓  
이미지 사이즈 최적화

↓  
Bottleneck 코드 최적화

↓  
Code Splitting









↓  
텍스트 압축

**Opportunities** — These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

Opportunity Estimated Savings  
▲ Properly size images 4.26 s

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more.](#)

☒ Show 3rd-party resources (8)

URL	Size	Potential Savings
 /photo-154...-956c469947f6? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	193 KB	185 KB
 /photo-145...-c3d57bc86b40? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	192 KB	184 KB
 /photo-149...-c5249f4df085? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	181 KB	174 KB
 /photo-151...-14dd9538aa97? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	150 KB	144 KB
 /photo-152...-a2d4ac8945e2? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	145 KB	139 KB
 /photo-154...-185e1e68f34e? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	142 KB	137 KB
 /photo-153...-c283d32ef9db? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	131 KB	126 KB
 /photo-149...-80ce9b88a853? w=1200&h=1200&q=80&fm=jpg&fit=crop (images.unsplash.com)	88 KB	85 KB

```
</div>
</div>

```

# 정리

Audits 툴을 이용한 페이지 감사



이미지 사이즈 최적화



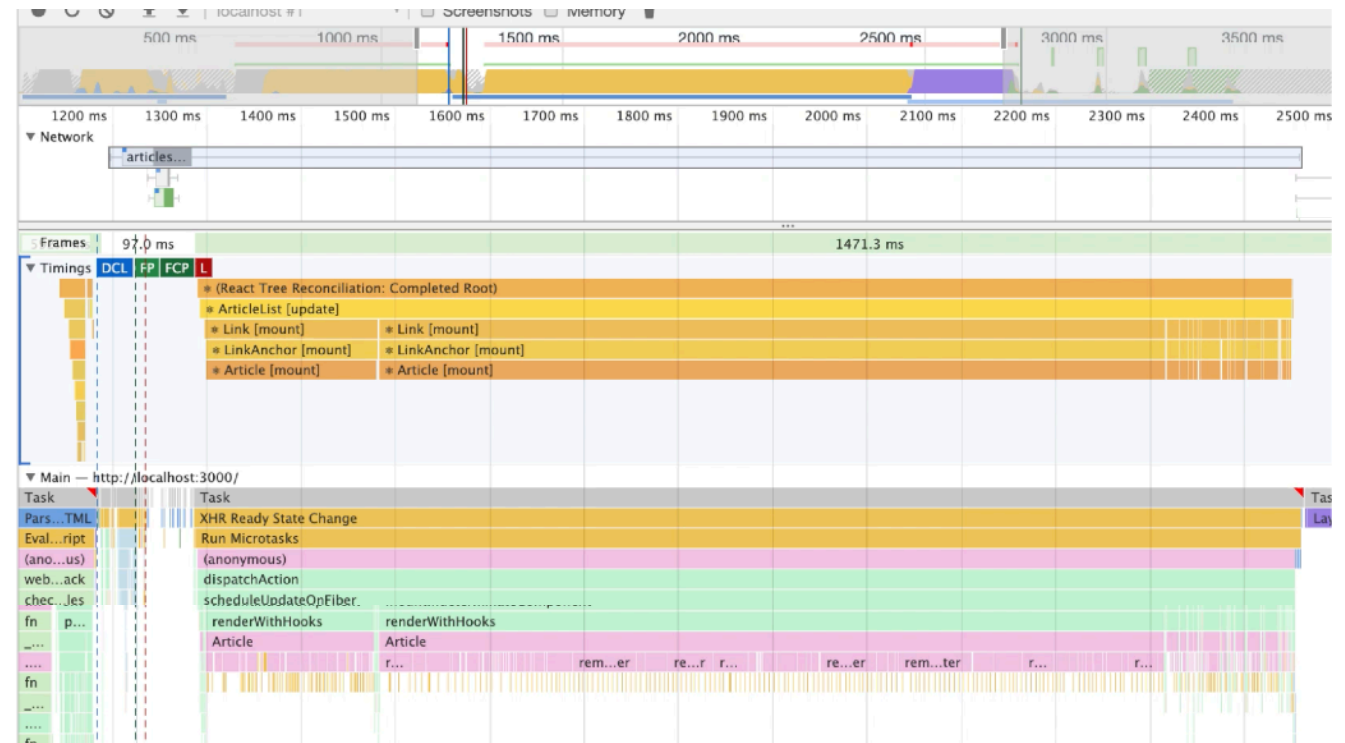
Bottleneck 코드 최적화



Code Splitting



텍스트 압축



```
function removeSpecialCharacter(str) {  
  /*  
   const removeCharacters = ['#', '_', '*', '~', '&', ';', '!', '[', ']', '\'', '>', '\n', '=', '-']  
   let _str = str  
   let i = 0,  
       j = 0  
  
   for (i = 0; i < removeCharacters.length; i++) {  
     j = 0  
     while (j < _str.length) {  
       if (_str[j] === removeCharacters[i]) {  
         _str = _str.substring(0, j).concat(_str.substring(j + 1))  
         continue  
       }  
       j++  
     }  
   }  
  */  
  
  let _str = str.substring(0, 300)  
  _str = _str.replace(/[\\#\\_\\*\\~\\&\\;\\~\\[\\]\\'\\>\\n\\=\\-]/g, '')  
  
  return _str  
}
```

# 정리

Audits 틀을 이용한 페이지 감사



이미지 사이즈 최적화



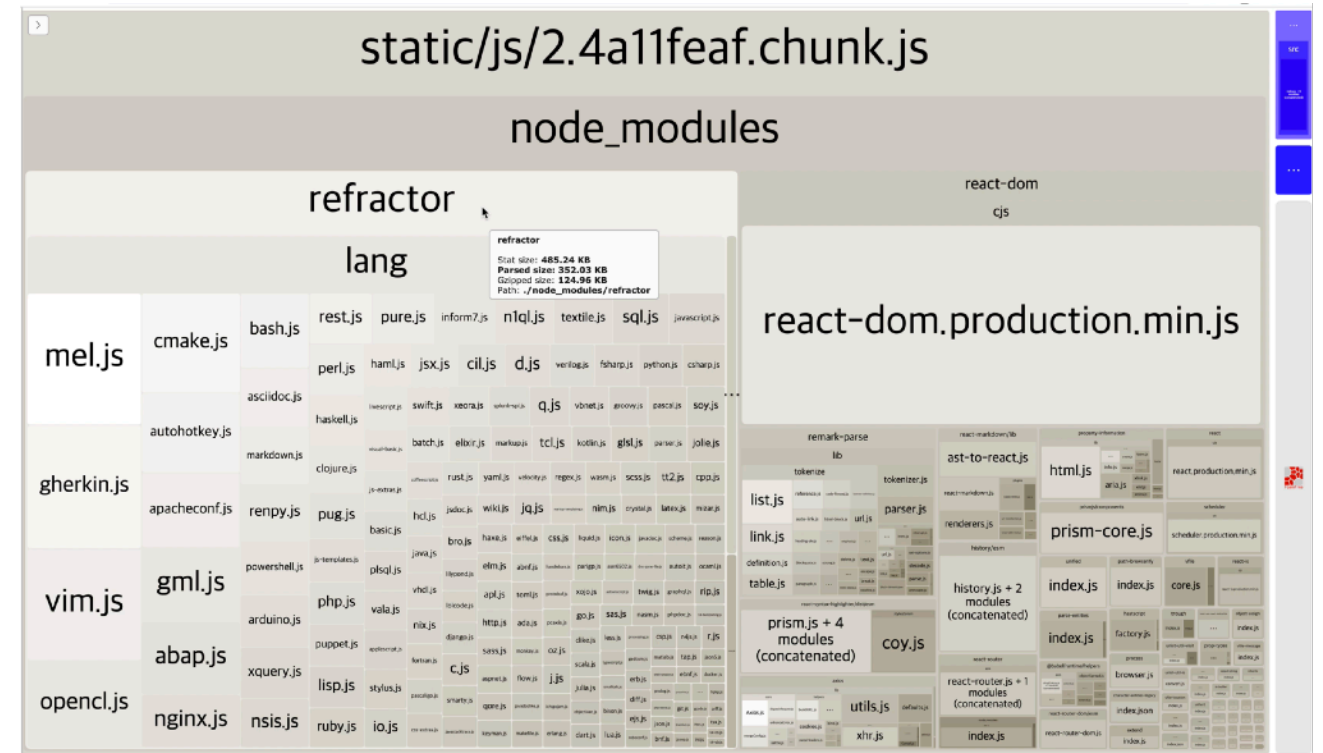
Bottleneck 코드 최적화



Code Splitting



텍스트 압축



```
const ListPage = lazy(() => import('./pages/ListPage/index'))
const ViewPage = lazy(() => import('./pages/ViewPage/index'))

function App() {
  return (
    <div className="App">
      <Suspense fallback=<div>로딩 중...</div>>
        <Switch>
          <Route path="/" component={ListPage} exact />
          <Route path="/view/:id" component={ViewPage} exact />
        </Switch>
      </Suspense>
    </div>
  )
}
```

# 정리

Audits 툴을 이용한 페이지 감사



이미지 사이즈 최적화



Bottleneck 코드 최적화



Code Splitting



텍스트 압축

**Opportunities** — These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

Opportunity	Estimated Savings
▲ Enable text compression	1.95 s
■ Eliminate render-blocking resources	0.15 s

4.c23ece81.chunk.js  
main.f4e47609.chunk.js  
react\_devtools\_backend.js  
0.012512ca.chunk.js  
5.7ec5e22b.chunk.css  
5.08c2dc88.chunk.js  
☐ articles  
☐ manifest.json  
☐ favicon.ico  
 photo-1542435503-956c469947f6?w=24...  
 photo-1537498425277-c283d32ef9db?w...  
 photo-1517694712202-14dd9538aa97?w...  
 photo-1517694712202-14dd9538aa97?w...

**Request Method:** GET  
**Status Code:** ● 200 OK  
**Remote Address:** [::1]:5000  
**Referrer Policy:** no-referrer-when-

▼ **Response Headers** [view source](#)  
**Accept-Ranges:** bytes  
**Connection:** keep-alive  
**Content-Disposition:** inline; filename=...  
**Content-Encoding:** gzip  
**Content-Type:** application/javascript  
**Date:** Thu, 14 May 2020 07:54:07  
**ETag:** "466e1bb1bbc37e6eccc82fd1"

# 1장 끝

수고하셨습니다

