

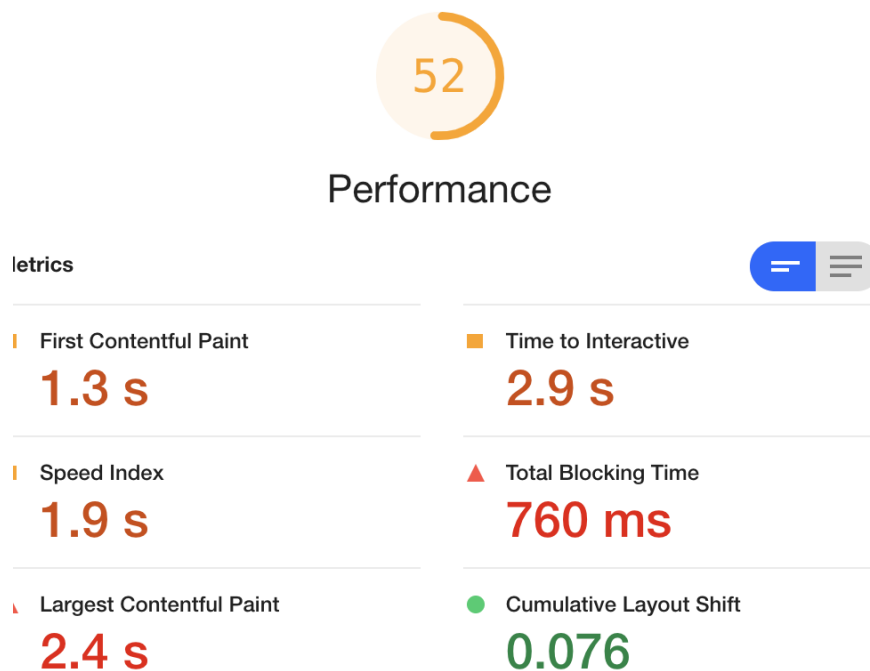
블로그 사이트 최적화

로딩 성능 최적화

- 이미지 사이즈 최적화
- code Split
- 텍스트 압축

Bottleneck 코드 최적화 - 렌더링 성능 회적화

Light House 성능 측정



측정 결과는 네트워크 및 CPU (또는 GPU) 등의 환경에 따라 검사 결과가 달라진다.

- Opportunities - 로딩 관점
- Diagnostics - 실행 관점 - 렌더링

이미지 사이즈 최적화

이미지 CDN을 이용한 최적화

CDN(Contents Delivery Network) 물리적 거리의 한계를 극복하기 위해 소비자와 가까운 곳에 콘텐츠 서버를 두는 기술

대부분의 서비스(회사)에서 자체 이미지 CDN을 구축해서 사용한다. 요즘에는 모든 기능들을 일일이 구현하지는 않고 클라우드 서비스를 통해서 구축하기도 합니다. (예를 들어 아마존의 S3 나 CloudFront 가 있습니다.)

이미지 CDN은 사이즈를 줄이거나 포맷을 바꾸는 처리과정을 거치는 기술이 아닌 이미지 제공에 특화된 CDN이다. 여기에 사이즈를 줄이거나 포맷을 바꾸는 처리과정

```
function getParametersForUnsplash({ width, height, quality, format }) {
  return `?w=${width}&h=${height}&q=${quality}&fm=${format}&fit=crop`;
}

<img
  src={
    props.image +
    getParametersForUnsplash({
      width: 240,
      height: 240,
      quality: 80,
      format: "jpg",
    })
  }
  alt="thumbnail"
/>
```

Code Splitting & Lazy Loading

코드 스플리팅이란 코드를 분할 하는 것 덩치 큰 번들 파일을 쪼개서 작은 사이즈의 파일로 만드는 것을 의미.

파일이 하나로 되어있을 경우 필요로 하지 않는 모듈을 로드하여 다운로드가 오래 걸리고 페이지 로딩 속도도 느려진다. 이 문제를 해결하기 위해 사용한다.

패턴

- 페이지 별로
- 모듈 별로

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

React 코드 분할

Route-based code splitting

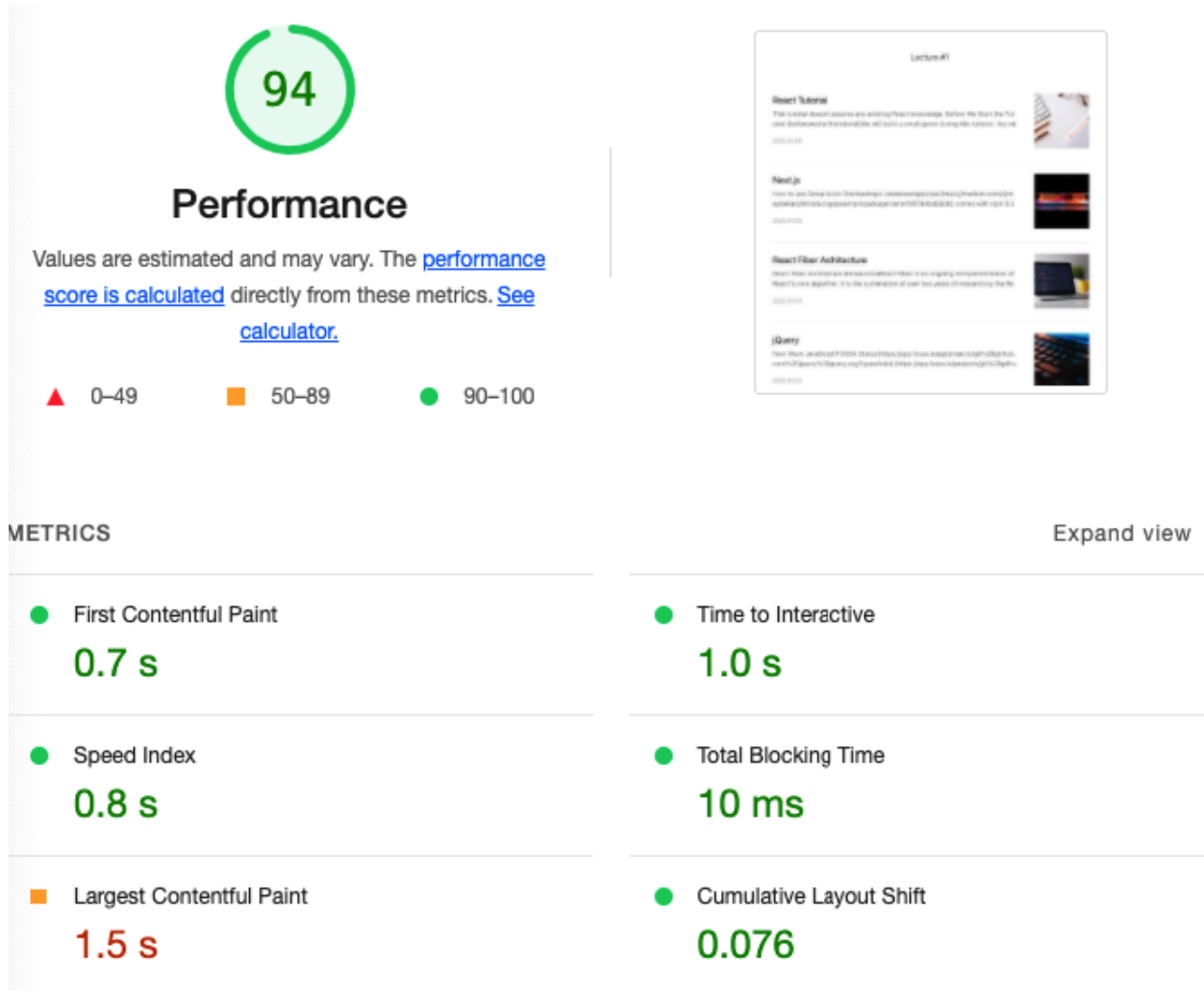
```
import React, { Suspense, lazy } from 'react';
import { BrowserRouter, Routes, Route } from 'react-router-dom';

const Home = lazy(() => import('./routes/Home'));
const About = lazy(() => import('./routes/About'));
/*
동적으로 lazy는 우리가 필요할 때 런타임 중에 Home 이라는 컴포넌트를 import를 하겠다.
실제 로드가 되는 시점은 정해진 주소로 접속을 했을 때 로드를 한다.

Suspense라는 모듈의 역할은 컴포넌트들이 동적으로 로드가 되다보니
어느 순간에는 아무것도 로드된 컴포넌트가 없는 순간이 생길 시 error를 발생 시키지 않고
fallback 안에 있는 요소를 노출한다.
*/
const App = () => (
  <Router>
    <Suspense fallback={<div>Loading...</div>}>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </Suspense>
  </Router>
);
```

코드 스플릿은 적용하는 주체가 리액트가 아닌 웹팩이기 때문에 웹팩에 설정을 해야한다.

<https://webpack.js.org/guides/code-splitting/>



이미지 사이즈 최적화, 병목 코드 제거, 코드 분할로 최적화를 하였습니다. 이미지 최적화만 했을 때는 점수가 3점 밖에 오르지 않아, 기대이하여서 실망을 하였습니다. 코드 분할로 인해 퍼포먼스 점수가 눈이 뜨게 올라가서 기분이 좋았습니다.

텍스트 압축

Enable text Compression - 서버로 부터 리소스를 받을 때 텍스트를 압축을 해서 받아라.

웹페이지를 로드 할때는 다양한 리소스를 다운 받게 된다 대표적으로 html, css, js 와 같은 텍스트 리소스들이다. 파일이 무거울 수록 다운 받는 속도는 느려진다. 이 문제를 해결하기 위해 텍스트 압축이다.

텍스트 압축이란 ? 파일을 압축하듯 서버에서 보내는 리소스를 압축하여 보내준다.

대표적인 압축 알고리즘

- GZIP
- Deflate