myshipdan

가장쉬운 식단, 마이쉽단

개발팀
프론트 엔드 개발자
정지우

서비스 성능 최적화

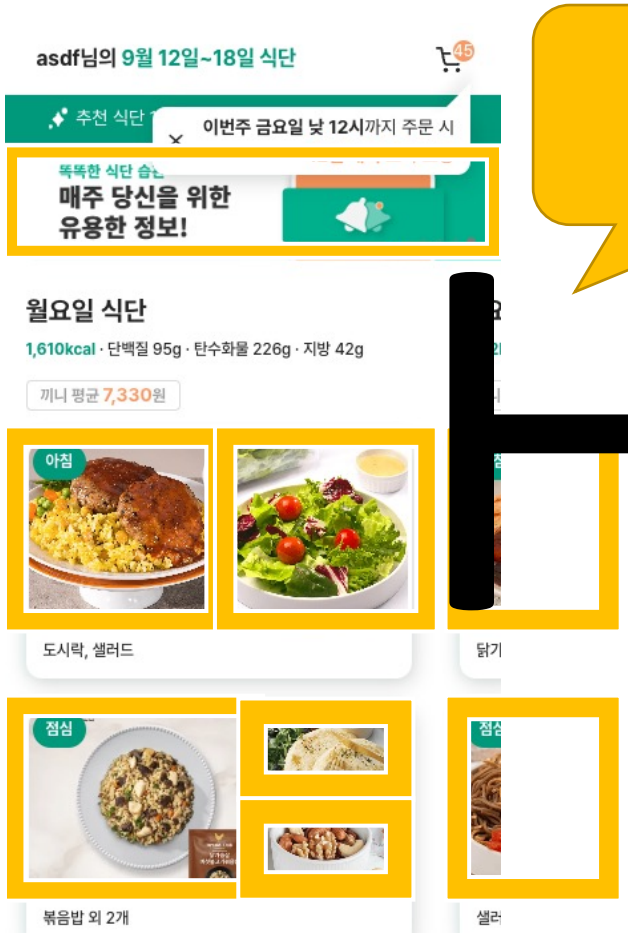
유저들에게 빠르게 보여주자

성능 최적화

- **개선의 여지**
  - 이미지 확장자(PNG)
  - 첫 화면 그리기 전, 카카오 관련 파일 다운로드를 기다린 후 다른 파일을 다운로드 받는다(script 동기 실행)
  - 화면이 그려진 후 사용자 정보를 받는다(CSR)
- **개선**
  - 이미지 확장자를 더 압축률 대비 퀄리티가 높은 것으로 변경(WEBP)
  - 첫 화면 그리기 전, 카카오 관련 파일과 함께 다른 파일을 다운로드(script 비동기 실행)
  - 사용자 정보는 첫 화면 그리기 전에 웹 서버에서 받는다(SSR)

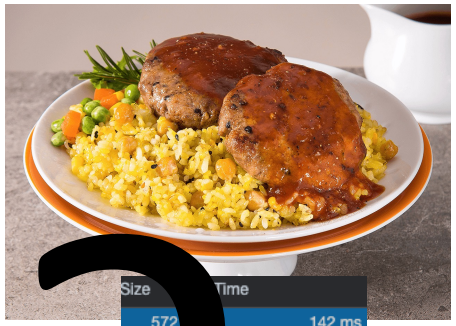
목차

이미지 확장자



모두 PNG

HOW?

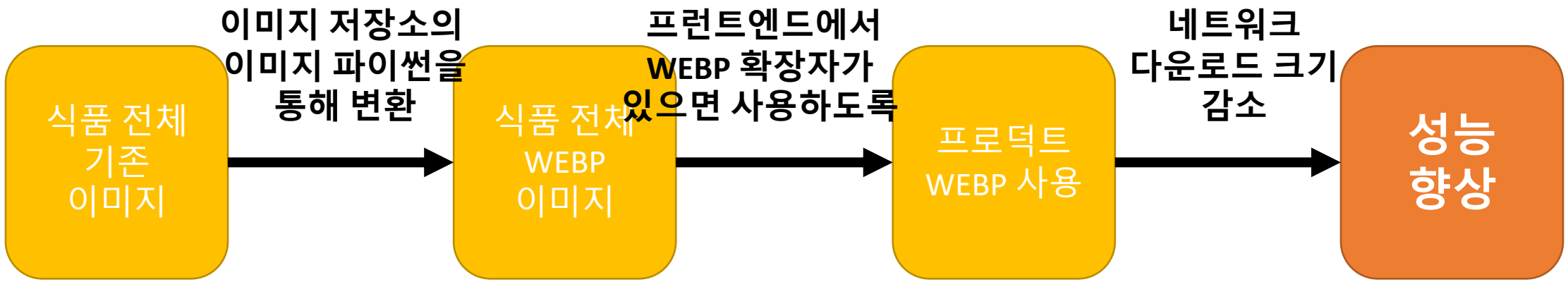


Size	Time
572	142 ms

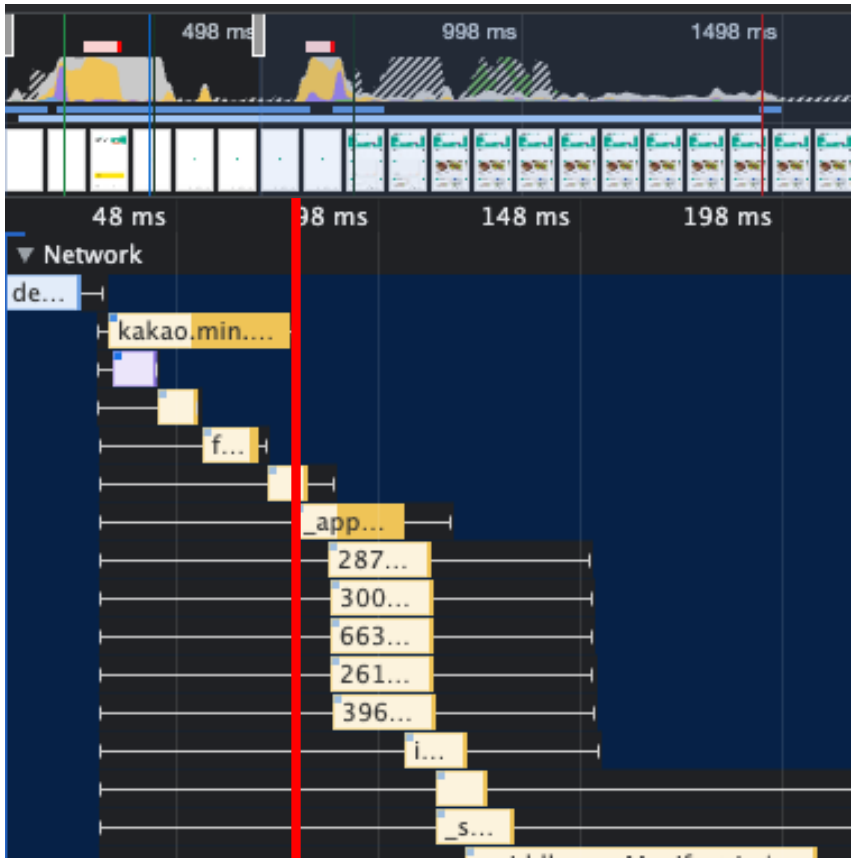
Size	Time
220 kB	32 ms

이미지 확장자

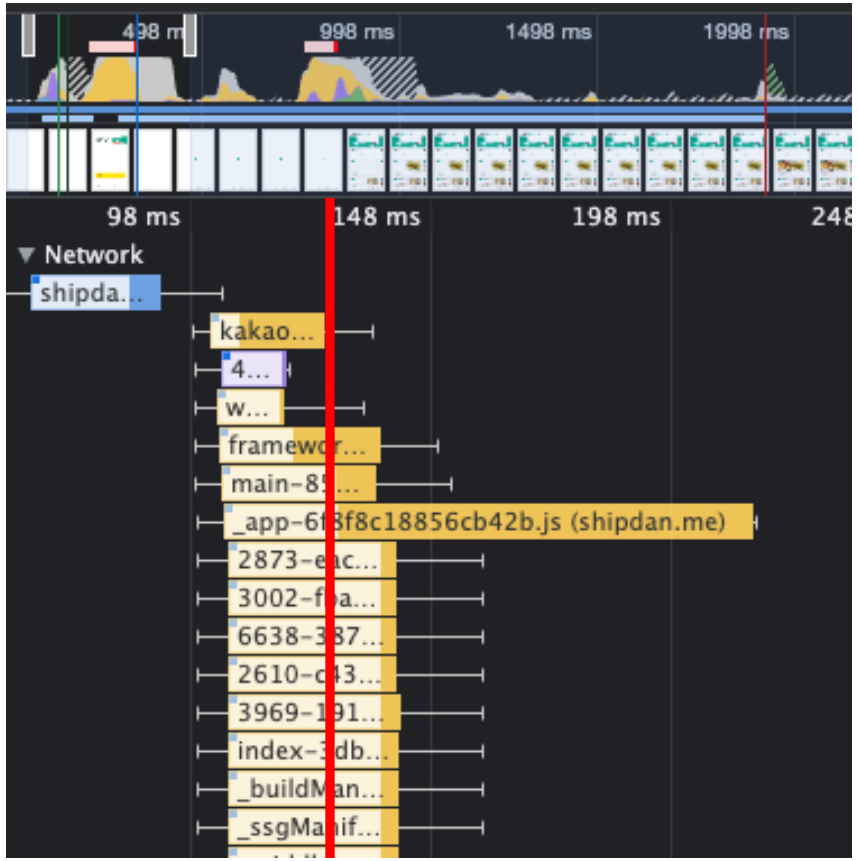
- WEBP
  - 기존 PNG, JPG 보다 크기가 25~35% 작다(비슷한 퀄리티에서)
  - 다운로드 받을 이미지의 크기가 줄어든다 -> 성능향상



첫 화면을 그리기 전 파일 다운로드



?



첫 화면을 그리기 전 파일 다운로드

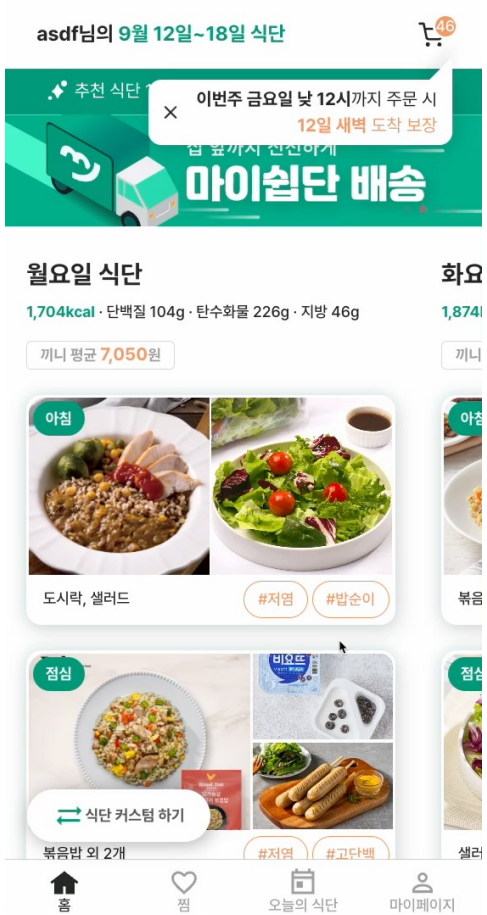
- kakao 관련 기능을 사용하기 위해 미리 다운로드 받아야하는 파일 다운로드 방식을 변경
  - 기존 : 동기(sync)
  - 변경 : 비동기(async/defer)
    - 참고 : async와 defer의 차이는 실행 시점에 있습니다.
      - async : 다운로드 받은 즉시 실행
      - defer : 다운로드 받고 화면이 그려진 후 실행

간단하게 말하면 kakao 파일 다운로드 안 기다린다!
같이 좀 다운 받자 !

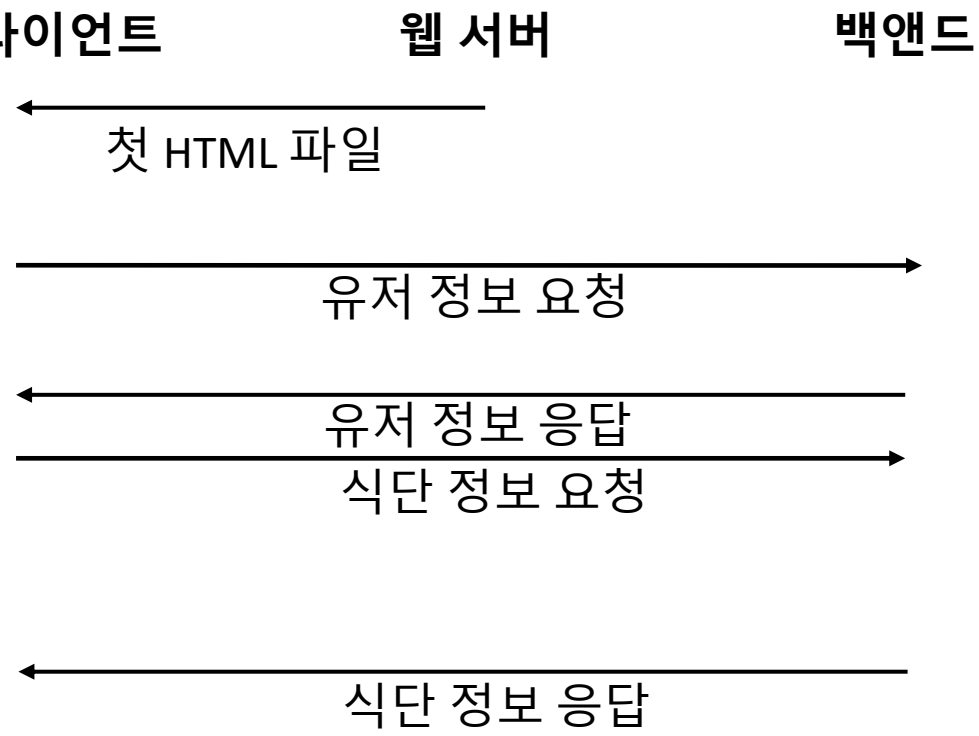
사용자 정보 얻기(적용 전)

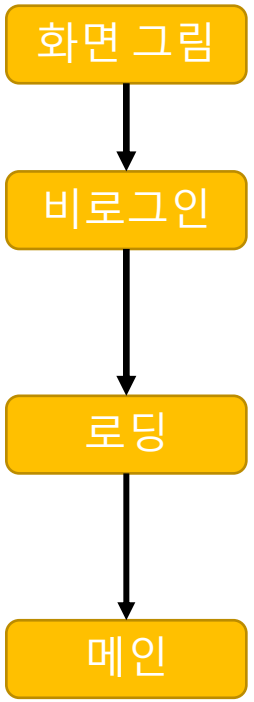
- 어디서 사용자 정보를 얻어서 화면에 그릴 건데?
  - 클라이언트에서!(CSR)
  - 서버에서!(SSR)
- 우리 프론트 엔드는 Next.js를 사용한다.
  - 리액트이지만 서버에서 데이터를 받아오는 것을 쉽게 해준다.
  - SSR이 쉽다

사용자 정보 얻기(적용 전)

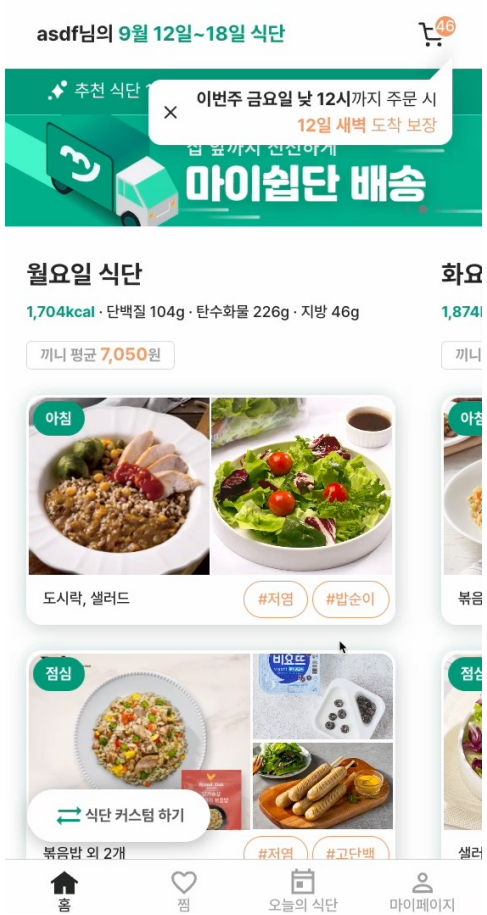


이미 로그인을 한 유저





사용자 정보 얻기(적용 전)



이미 로그인을 한 유저

웹 클라이언트 **웹 서버** **백앤드**

유저 정보 요청

유저 정보 응답

첫 HTML 파일

식단 정보 요청

식단 정보 응답

화면 그림

로딩

메인

결과 및 비교(이전)

METRICS Expand view

First Contentful Paint 2.1 s	Time to Interactive 18.3 s
Speed Index 11.4 s	Total Blocking Time 110 ms
Largest Contentful Paint 4.7 s	Cumulative Layout Shift 0.184

View Original Trace View Treemap

결과 및 비교(이후)

68

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 50–89 90–100

METRICS

First Contentful Paint 0.9 s	▲ Time to Interactive 18.3 s
▲ Speed Index 11.7 s	Total Blocking Time 230 ms
Largest Contentful Paint 2.7 s	Cumulative Layout Shift 0.183

View Original Trace View Treemap

73

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 50–89 90–100

METRICS Expand view

First Contentful Paint 0.9 s	▲ Time to Interactive 20.8 s
▲ Speed Index 12.9 s	Total Blocking Time 300 ms
Largest Contentful Paint 1.0 s	Cumulative Layout Shift 0.093

View Original Trace View Treemap

결과 및 비교(비교)

LCP

Largest Contentful Paint

▲ Largest Contentful Paint

4.7 s

Largest Contentful Paint

2.7 s

Largest Contentful Paint

1.0 s

- Largest Contentful Paint(LCP)란?
페이지가 처음으로 로드를 시작한 시점을 기준으로
화면 내에 있는 가장 큰 이미지 또는 텍스트의 나타나는 시간을 보고하는 지표 |

결론

다른 유명 플랫폼과 비교했을 때 LCP만 놓고 보면 좋은 성능이란 것을 알 수 있다.
LCP가 개발을 잘했음의 절대적인 지표가 될 수 없지만 그래도 타사와 비교했을 때
좋은 지표를 보여주고 있음은 확인 할 수 있다.

아직 PNG 확장자를 쓰고 있는 이미지들도 많고 개선사항도 많아
Performance 점수가 초록색은 아니다.
PNG 확장자 이미지들은 웹 서버에 캐시서버(CloudFront)를 붙여 서버에 캐시 되도록 해
유저에게 빠르게 이미지를 보여줄 수 있도록 처리하였다.

앞으로도 부족한 사항들도 개선하려고 노력하는 개발자/개발팀이 되겠습니다.
감사합니다.

*해당 발표자료의 모든 Lighthouse 성능 측정은 시크릿모드에서 실행됨. 캐시 저장 Disabled 상태에서 진행함.