

포트폴리오

프로젝트 소개

프로젝트명	다다 (DA,DA)
기간	2017-10-10 ~ 2017-11-03
역할	프론트엔드 개발
프로젝트 홈페이지	https://dada.downmix.net
Github	https://github.com/huusz/Da-Da
Youtube	https://www.youtube.com/watch?v=DWtnFShrpKM

1.스킬

프로젝트 전체 스택은 <https://github.com/huusz/Da-Da#about-tech>에 정리 되어 있습니다.

아래는 직접 활용한 스킬 및 툴입니다.

와이어프레임, 프로토타이핑	Google Presentation, Invision
UI 개발	<ul style="list-style-type: none">- React, React-Router, Redux- Semantic-ui-react, CSS- Draft.js (Text editor Framework)- recharts.js (Chart Library)
Ajax Network	fetch API
VCS	Git
Package Manager	yarn
Convention Formatter	<ul style="list-style-type: none">- editor config- Prettier

2. 개요

- React와 Redux 학습을 목적으로 하는 프로젝트입니다.
- 프론트엔드 2명, 백엔드 3명이 협업하였습니다.
- Git을 활용하여 버전 관리, Git Project를 통하여 역할 분담 및 일정 관리를 진행하였습니다.
- 최대한 다양한 기능 구현에 집중하기 위해 디자인 및 CSS에 들이는 시간을 최소화하였습니다.
- 텍스트 에디터 및 차트 기능은 외부 프레임워크와 라이브러리를 활용하였습니다.
- 팀원 간 컨벤션 통일을 위해 Formatter와 Linting tool을 사용하였습니다.

3. 특징

- 1) 데스크탑 환경을 우선시 합니다.
 - 1200px을 중단점으로 더 작은 환경에서는 데스크탑 이용을 권장하는 안내 페이지가 로드됩니다.

2) Semantic-ui-react + jsx inline-style + CSS

- css 프레임워크 semantic-ui-react를 기본으로, jsx inline-style과 css를 활용하여 커스텀 스타일링을 하였습니다.

3) SPA & 클라이언트사이드 렌더링

- React를 이용하여 SPA로 구현되었습니다.
- 다다는 회원제(비공개) 서비스입니다.
- 사용자가 다다 앱 내에서 작성하는 모든 기록은 본인 외 타인은 볼 수 없어야 합니다.
- 이러한 비공개성을 유지하기 위하여 SEO에 대한 고려는 하지 않았습니다.

4) 자동 로그인을 지원합니다.

- 최초 소셜 로그인을 통해 얻은 사용자의 JWT token을 Local Storage에 저장합니다. 이로써 사용자가 새 탭이나 새 윈도우로 방문하여도 로그인을 다시 요구하지 않습니다.

6) React-Router v4로 라우팅이 구현되었습니다.

7) 서버와의 모든 통신은 fetch API를 이용한 Ajax방식으로 이루어집니다.

- 회원 인증과 식단/운동/일기의 등록/수정/삭제를 비롯, 식단/운동/레시피 검색 등 앱 내 모든 기능이 Ajax 통신으로 이루어집니다.

8) React 앱 내 상태 관리는 Redux를 통해 이루어집니다.

- 컴포넌트 간 원활한 상태 관리를 위하여 Redux를 도입하였습니다.
- 미들웨어로는 redux-thunk와 redux-logger를 사용하였으며, 프로덕션 모드(배포 시)에서 redux-logger는 비활성화 합니다.

4. 기능 (제가 구현한 것은 밑줄 표시 하였습니다.)

1) 소셜로그인

- 페이스북/ 인스타그램/ 네이버/ 카카오를 통한 소셜 로그인을 지원합니다.
- 해당 SNS를 통한 인증이 완료되면 서버로부터 받은 유저의 토큰 정보를 Local Storage에 저장합니다. 토큰 정보가 성공적으로 저장되면, url을 Home으로 이동 시킵니다.
- 유저이름/ 프로필 사진은 유저의 토큰 정보를 기반으로 인증 API로부터 불러옵니다.
- 첫 로그인한 유저는 홈으로 이동하기 전 필수적으로 기본 정보를 입력 합니다.
- 유저가 입력한 정보를 기반으로 권장 체중과 권장 칼로리를 계산하여 입력창 밑에서 제안합니다.
- 유저가 입력한 정보는 API요청을 통해 데이터베이스에 저장됩니다.
- 토큰 정보가 존재할 때만(로그인 했을 경우) 서비스 페이지로 접근이 가능합니다.
- (라우팅) 모든 서비스 페이지에서 토큰 정보가 존재하지 않는 경우 로그인 페이지로 Redirect 합니다



2) 검색 기능

- 다이어리 페이지의 식단 & 운동 기록에서 텍스트 검색과 사진 검색을 지원합니다.
- 텍스트 검색: 사용자가 입력한 텍스트를 키워드로 DB에서 해당 내용을 검색합니다.
- 사진 검색: 사용자가 파일시스템을 통해 사진을 업로드 합니다. 클라이언트는 서버로 파일과 함께 요청을 보내고, 서버는 vision api를 통해 분석된 결과 후보를 응답으로 보내줍니다. 서버로부터 받은 데이터를 사용자가 보는 결과 뷰에 체크박스 형식으로 그려줍니다. 사용자는 원하는 결과를 선택할 수 있고, 원하는 결과가 없을 경우 사진 검색을 포기할 수 있습니다. 사용자가 선택한 결과 후보를 키워드로 텍스트 검색이 한번 더 이루어집니다.
- 사용자는 자신의 데스크탑에서 dada 앱으로 마우스 drag & drop 방식의 업로드도 가능합니다.

3) 네비게이션

- 해당 앱에는 총 4개의 네비게이션이 존재합니다.
- 라우터와 관련된 네비게이션은 2개입니다.

(1) 앱 내 페이지 간 이동을 구현하는 최상위 네비게이션



(2) 페이지 내 이동을 구현하는 서브 네비게이션

- 다이어리 페이지 내에서 라우팅을 통해 식단/운동/일기로 각기 다른 URL과 뷰를 가져옵니다.

- 라우터와 무관한 네비게이션으로 2개의 날짜 이동 네비게이션이 있습니다.
- 날짜 이동은 둘 모두 일 단위로 이루어집니다.

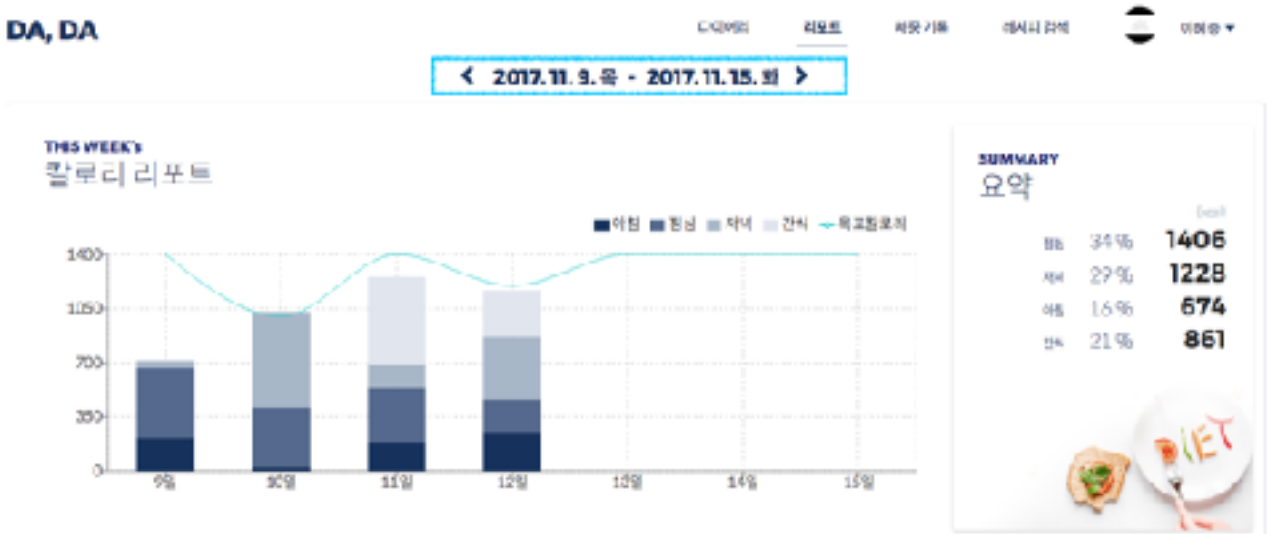
(1) 다이어리 페이지 내 날짜 네비게이션

- view에는 하루 단위로 표시됩니다. (ex. 2017.11.1 -> 2017.11.2)



(2) 리포트(주 별 통계) 페이지 내 날짜 네비게이션

- view에는 주 단위로 표시됩니다.
- ex. 2017.11.1 ~ 2017.11.7 -> 2017.11.2 ~ 2017.11.8



- 날짜 이동 이벤트가 발생하면 해당 날짜로 서버에 요청을 보냅니다. 받은 데이터로 view를 다시 렌더합니다.

- 순차적인 진행을 위해 날짜 이동 액션을 Promise로 선언하여 동기식으로 구현하였습니다.

4) 통계

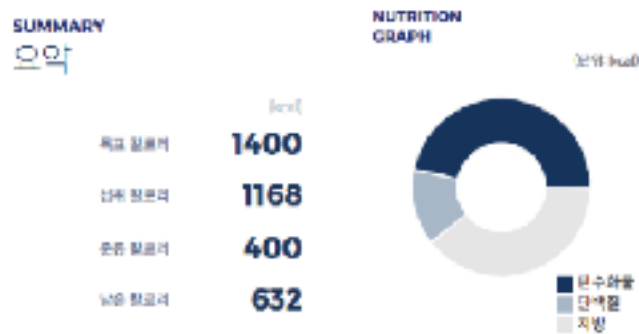
- 총 4가지의 그래프와 3개의 요약 정보를 제공합니다.

(1) 그래프

- React와 D3 기반의 라이브러리 Recharts를 활용합니다.
- 다이어리/ 리포트/ 체중기록 페이지에 각각 Pie/ Bar & Line/ Area & Line 그래프를 표현합니다.
- 사용자가 입력하여 DB에 등록된 데이터를 불러옵니다.
- 클라이언트에서 정해진 포맷([{ name: value }])에 맞게 가공한 후 그래프에 적용합니다.

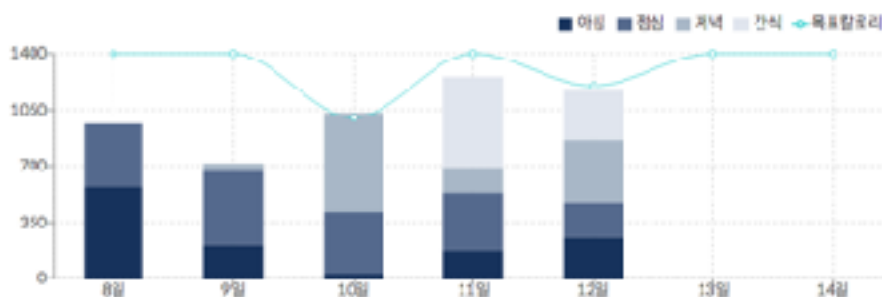
(2) 요약 정보

- 다이어리/ 리포트 페이지에서 리스트 형태로 요약 정보를 제공합니다.
- 사용자가 입력한 데이터를 기반으로 합계와 비율 계산 연산을 수행합니다.

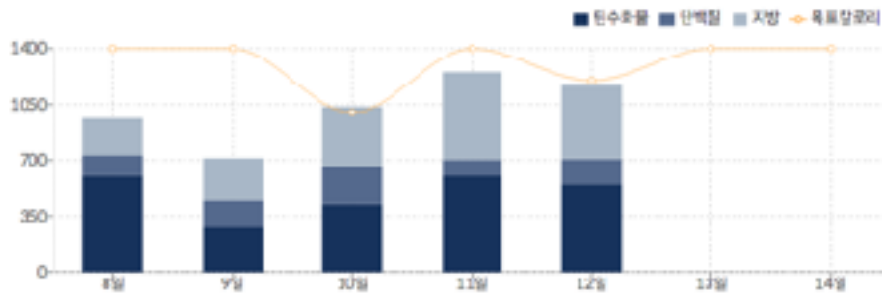


다이어리 페이지의 우측에 위치하는 요약 정보와 파이그래프

THIS WEEK'S 칼로리 리포트



THIS WEEK's 영양분 리포트



SUMMARY 요약

(kcal)

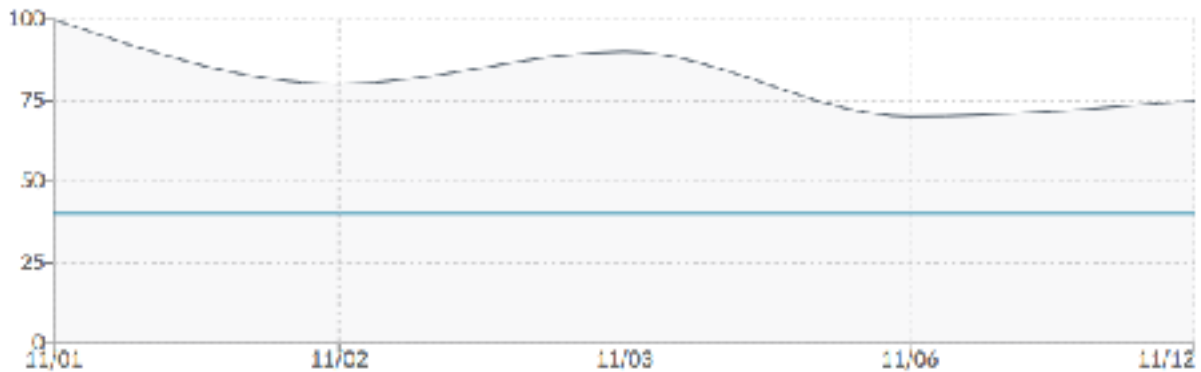
탄수화물 **60 %**

단백질 **19 %**

지방 **21 %**



리포트 페이지의 바 & 라인 그래프와 요약 정보



체중기록 페이지의 영역(Area) & Line 그래프

- 다이어리/ 체중 기록 페이지에서는 사용자의 데이터 입력/ 수정/ 삭제 이벤트가 발생합니다.
- 이벤트 발생 시, 동시에 그래프와 요약정보도 함께 갱신되는 이벤트가 호출됩니다.
- 그래프에 사용될 데이터를 가져오는 서버와의 통신은 페이지 첫 로드 시, 딱 한번만 수행합니다. 그래프와 요약정보의 갱신 이벤트는 통신과 무관하게 클라이언트에서만 처리됩니다.
- 사용자가 입력/ 수정/ 삭제 이벤트를 발생시키면 클라이언트에서 사용자가 입력한 데이터와 현재 상태(state)에 있는 데이터를 연산한 결과로 view만 다시 렌더합니다.

5) 일기 기록

- 다이어리 페이지 내 일기 페이지는 사용자로부터 2가지 데이터 입력을 받습니다.
- 두 입력은 서로 독립적입니다.
- 두 입력 모두 등록/ 수정/ 삭제 기능을 제공합니다.

(1) 반성 일기

- 반성 일기는 30자 이내의 단문만 입력할 수 있는 input 입니다.
- 사용자가 작성한 반성 일기는 다음 날 로그인 시 메인 페이지 로딩 화면 메시지로 나타납니다.
- 만약 전 날 반성 일기 데이터가 존재하지 않으면, 미리 준비한 default 메시지를 표시합니다.

- 오늘의 반성 일기 (30자 내외)

반성 일기 등록 화면

30자 초과 시 화면: 경고창 & 버튼 비활성화

반성 일기 등록 완료된 화면

등록 후 마우스 오버 시 화면 (해당 영역 클릭 시 수정 가능)



다음 날 로그인 시 홈으로 가는 로딩 화면 메시지로 표시됨

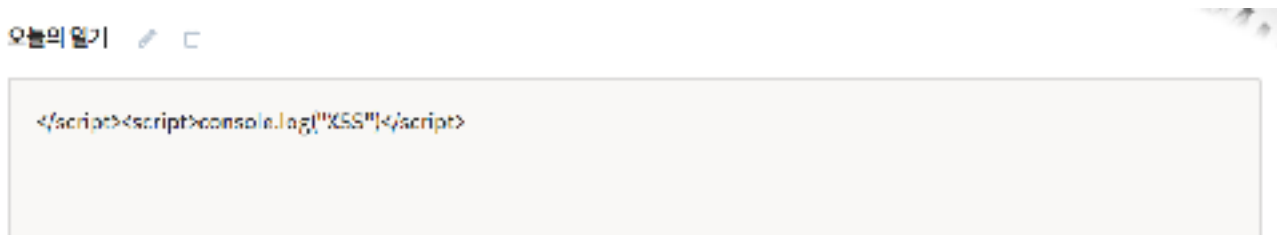
(2) 오늘의 일기

- 페이스북의 Rich Text Editor Framework인 Draft.js로 작성한 WYGIWYG 에디터입니다.
- 오늘의 일기는 글자 수 제한 없이 장문의 입력을 허용합니다.
- 볼드/ 이탤릭/ 밑줄/ 취소선/ H1/ 인용/ 정렬(왼쪽, 가운데, 오른쪽)/ ol/ ul/ 색상 변경을 제공합니다.
- 볼드(B)/ 이탤릭(I)/ 밑줄(U)/ 취소선(-)은 키 커맨드를 매칭하였습니다.
- 윈도우는 ctrl, 맥OS는 command가 default로 매칭 되어 있습니다.
- 에디터 위에 작성 중인 내용은 입력됨과 동시에 브라우저의 Local Storage에 저장됩니다.
- 따라서 사용자가 일기 작성 중 페이지를 이탈하여도 작성 중이던 내용은 보존됩니다.



오늘의 일기 에디터 화면

- 일기 등록 이벤트가 발생하면 Local Storage에 저장된 String을 JSON 객체(Draft.js의 ContentState)로, 그 다음 HTML로 순차적으로 변환합니다.
- 등록된 일기를 표시할 영역 태그(div)에 dangerouslySetInnerHTML객체를 전달하여 innerHTML 방식으로 HTML코드를 삽입합니다.
- XSS 공격에 대한 방어는 위험 요소 텍스트를 ContentState에서 유니코드화 하는 방식으로 대응됩니다. (이는 draft.js에서 제공하는 부분입니다.)

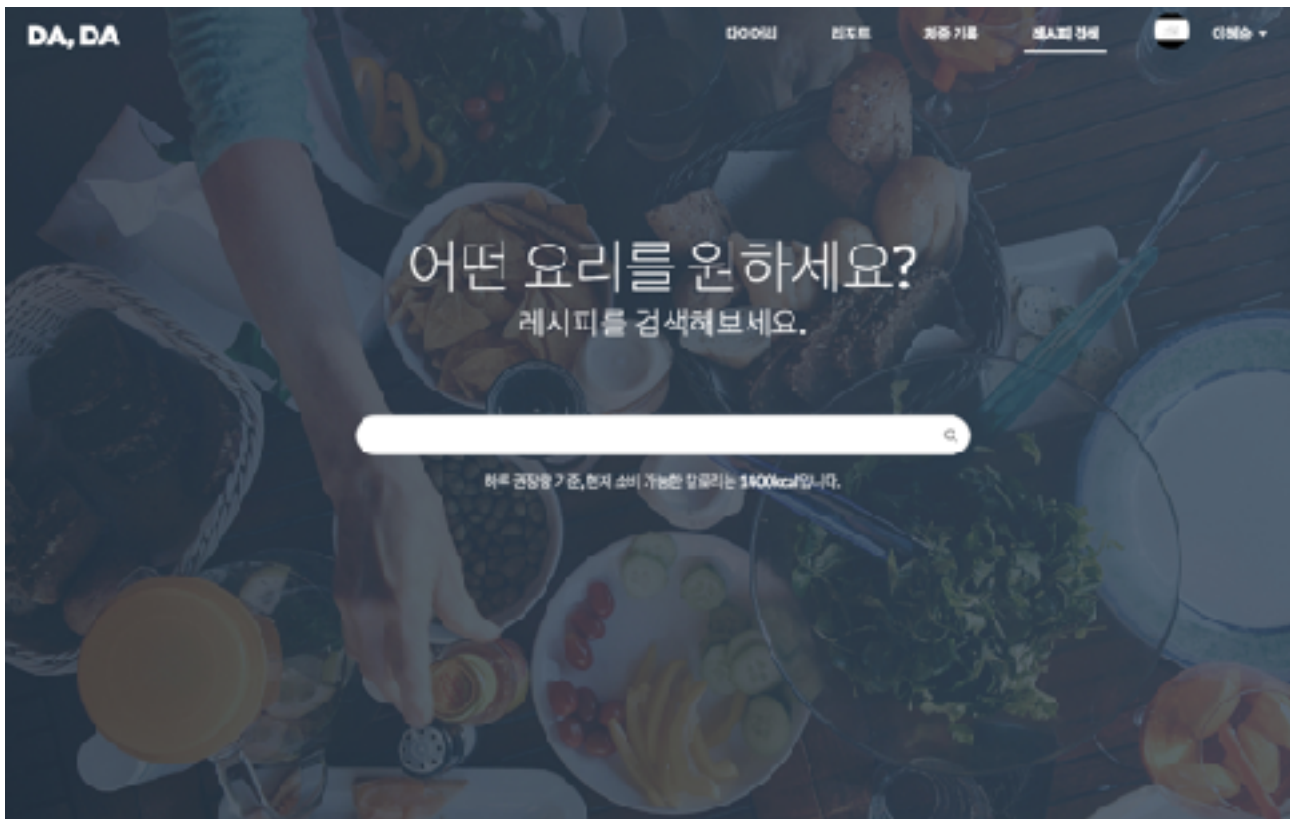


오늘의 일기 등록 후 화면

6) 레시피 검색

- 텍스트 검색을 지원합니다.

- 검색 키워드를 입력하면 해당 키워드를 url 파라미터로 하여 결과 페이지를 라우팅합니다.
- 사용자가 입력한 키워드에 대한 검색 알고리즘은 서버에서 구현합니다. 클라이언트는 해당 키워드에 대한 API 요청을 보낸 뒤 받은 결과를 렌더합니다.



레시피 검색 페이지

① <https://dada.downmbx.net/search/오>

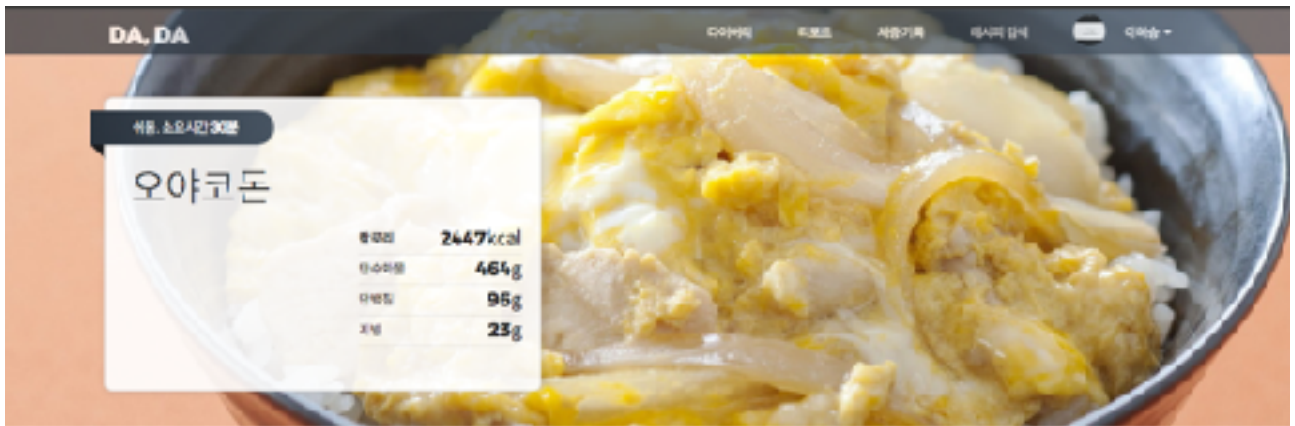
‘오’ 키워드로 검색한 결과 페이지의 url



‘오’ 키워드로 검색한 결과 화면

7) 레시피 조회

- 검색 결과로 조회된 레시피를 선택하면 레시피 페이지로 이동합니다.
- 레시피 페이지는 검색을 통해서만 접근이 가능합니다.



재료

2

인분

닭다리살 (뼈 있는 고기)

160g

양파

100g

밥

4공기

당근

3개

물

1컵

간장

4큰술

참깨

1큰술

* Tip: 타이완 스프

1

닭다리살을 고기어 놓는다

2

닭고기와 양파를 썰다. 닭고기는 한자 폭 2센티미터 정도로 가늘게 썰 다음 다시 두께 5밀리미터로 저른다. 양파는 폭 5밀리미터 정도로 얇게 썰다. 당근 3개를 불에 익혀준다.

3

맛있는 계란을 모두 프라이팬에 볶고 삶는다.

4

닭고기와 양파가 한대 끓어오르면 프라이팬에 잘 때 놓는다. 양파를 넣고 약한 불에서 4분 끓인다.

'오야코돈' 레시피 페이지

- 사용자는 재료란에서 드롭다운으로 몇 인분을 요리할 것인지 직접 설정할 수 있습니다.
- 설정된 양에 따라 개별 재료의 양을 연산하여 새로 렌더합니다.

재료

2

인분

닭다리살 (뼈 있는 고기)

80g

양파

50g

밥

2공기

당근

1.5개

물

0.5컵

간장

2큰술

참깨

0.5큰술

* Tip: 타이완 스프

재료

4

인분

닭다리살 (뼈 있는 고기)

160g

양파

100g

밥

4공기

당근

3개

물

1컵

간장

4큰술

참깨

1큰술

* Tip: 타이완 스프

인분 수에 따라 다르게 표시되는 재료 양

- 레시피를 스텝 별로 나누어 완료한 부분은 체크할 수 있도록 체크박스로 구현하였습니다.

1. 마늘 알을 저어 놓는다.
2. 달고기와 양파를 썰다. 달고기를 먼저 폭 3센티미터 정도로 가늘게 썰 다음 다시 두께 5밀리미터로 저절다. 양파는 폭 5밀리미터 정도로 잘게 썰다. 달걀 3개를 볼에다 풀어놓다.
3. 마늘을 재료들 모두 프라이팬에 넣고 끓인다.
4. 달고기와 양파가 반데 못치지 않도록 프라이팬 전치에 잘 펴 놓는다. 뚜껑을 덮고 약한 불에서 4분 끓인다.
5. 잘 끓으면 뚜껑을 열고 달걀을 팬에 돌려가면서 붓는다. 달걀이 반숙 정도로 익으면 불을 끄고 1, 2분 썰을 돌린다.
6. 골작한 사발에 따뜻한 밥을 담고 그 위에 전더기를 얹어 맛국물을 붓는다.

→ 기록 다이어리에 등록하기

레시피 스텝 체크박스

- 레시피를 통해 요리한 음식을 식단에 등록할 수 있습니다.
- 다이어리에 등록할 수 있는 버튼을 페이지 하단에 제공합니다.
- 버튼을 클릭하면, 모달을 통해 사용자는 시간대를 선택하여 등록할 수 있습니다.
- 재료 페이지에서 선택된 인분 수를 양(serve)으로 하여 API요청을 보냅니다.