API 설계서

1. 기본 정보

- **Base URL**: `http://localhost:8080/api`

인증 방식

- **방식**: JWT (Json Web Token)
- **전달 경로**: HttpOnly 쿠키 (jwtToken)
- **쿠키 예시**:

  ```
  Set-Cookie: jwtToken=eyJhbGciOiJIUzI1Nils...; HttpOnly; Secure; Path=/; Max-Age=86400
  ```

2. 공통 응답 형식

X

3. 인증 API

3.1 회원가입

- **POST** `/api/auth/register`
- 새 사용자를 등록합니다.

요청 예시 (JSON)

```
{
  "email": "user@example.com",
  "name": "홍길동",
  "password": "1234abcd!",
  "height": 175,
  "age": 28,
  "gender": "MALE",        // FEMALE, OTHER 도 가능
  "goal": "체중감량",       // 또는 "근력향상"
  "diseases": ["고혈압", "당뇨"]
}
```

응답 예시

```
{
  "message": "회원가입 성공"
}
```

상태 코드

3.2 로그인

- POST /api/auth/login
- 로그인 인증을 수행하고, JWT 토큰을 HTTP-only 쿠키로 발급합니다.

요청 예시

```
{
  "email": "user@example.com",
  "password": "1234abcd!"
}
```

응답

- 200 OK
- 별도 Body 없이 jwtToken 이라는 이름의 HttpOnly 쿠키가 발급됩니다.

3.3 로그아웃

- POST /api/auth/logout
- JWT 쿠키를 삭제하여 로그아웃 처리합니다.

응답

- 200 OK + 쿠키 만료됨 (jwtToken 삭제)

3.4 현재 로그인한 사용자 조회

- GET /api/auth/me
- 현재 인증된 사용자의 userId 및 email 을 반환합니다.

응답 예시

```
{
  "userId": 1,
  "email": "user@example.com"
}
```

4. Member API

4.1 내 정보 조회

- GET /api/members/me
- **설명**: 로그인된 사용자의 정보를 반환합니다.
- **인증**: JWT 쿠키 필요
- **응답 예시**

```
{
  "userId": 1,
  "email": "test@example.com",
  "name": "홍길동",
  "height": 175,
  "age": 30,
  "goal": "체중감량",
  "gender": "MALE",
  "diseases": ["당뇨", "고혈압"]
}
```

4.2 내 정보 수정

- PUT /api/members/me
- **설명**: 로그인된 사용자의 정보를 수정합니다.
- **인증**: JWT 쿠키 필요
- **요청 바디 예시**

```
{
  "name": "홍길순",
  "height": 165,
```

```
  "age": 28,
  "goal": "근력향상",
  "gender": "FEMALE",
  "diseases": ["관절염"]
}
```

4.3 특정 사용자 조회

- GET /api/members/{id}
- **설명**: 특정 userId 의 회원 정보를 조회합니다.
- **인증**: JWT 쿠키 필요

4.4 회원 목록 조회

- GET /api/members
- **설명**: 전체 회원 목록을 조회합니다.
- **인증**: JWT 쿠키 필요

4.5 회원 정보 일부 수정

- PATCH /api/members/{id}
- **설명**: 특정 회원의 일부 정보를 수정합니다.
- **요청 예시**

```
{
  "name": "김인태",
  "goal": "근력향상"
}
```

4.6 회원 삭제

- DELETE /api/members/{id}
- **설명**: 특정 회원을 삭제합니다.
- **인증**: JWT 쿠키 필요

5. Record API

5.1 기록 생성

- POST /api/records
- **설명**: 오늘 날짜로 건강 기록을 생성하고 AI 추천 결과(할 일, 식단, 응원 메시지)를 반환합니다.
- **인증**: JWT 쿠키 필요

요청 예시

```
{
  "sleep": 7.5,
  "weight": 65,
  "fat": 20.5,
  "muscle": 29.3,
  "bmr": 1400.5,
  "bmi": 22.1,
  "vai": 1.2,
  "date": "2025-07-10",
  "prompt": "운동 루틴 추천",
  "place": "헬스장"
}
```

응답 예시

```
{
  "todolists": [
    {
      "todoItem": "스트레칭 10분",
      "tip": "운동 전 스트레칭은 부상 방지에 좋아요"
    },
    {
      "todoItem": "런닝머신 30분",
      "tip": "유산소 운동으로 체지방 감량을 도와줘요"
    }
  ],
  "diet": [
    {
      "breakfast": "귀리죽",
      "lunch": "닭가슴살 샐러드",
```

```
      "dinner": "두부 샐러드"
    }
  ],
  "cheering": "오늘도 건강한 하루 되세요! 💪"
}
```

5.2 기록 단건 조회

- **GET** /api/records/{id}
- **설명**: 특정 기록을 ID 로 조회합니다.
- **인증**: JWT 쿠키 필요

응답 예시


```
{
  "id": 3,
  "userId": 1,
  "sleep": 7.5,
  "weight": 65,
  "fat": 20.5,
  "muscle": 29.3,
  "bmr": 1400.5,
  "bmi": 22.1,
  "vai": 1.2,
  "date": "2025-07-10"
}
```

5.3 내 기록 전체 조회

- **GET** /api/records/my
- **설명**: 로그인된 사용자의 모든 기록을 반환합니다.
- **인증**: JWT 쿠키 필요

응답 예시


```
[
  {
    "id": 1,
    "userId": 1,
    "sleep": 7.0,
```

```
    "weight": 66,
    "fat": 21.0,
    "muscle": 28.9,
    "bmr": 1380.0,
    "bmi": 22.5,
    "vai": 1.3,
    "date": "2025-07-08"
  },
  {
    "id": 2,
    "userId": 1,
    "sleep": 6.5,
    "weight": 65,
    "fat": 20.5,
    "muscle": 29.1,
    "bmr": 1395.0,
    "bmi": 22.0,
    "vai": 1.2,
    "date": "2025-07-09"
  }
]
```

5.4 기록 전체 페이징 조회

- **GET** /api/records?page=0&size=10
- **설명**: 모든 사용자의 기록을 페이징 처리하여 반환합니다.(관리자용)
- **인증**: JWT 쿠키 필요

응답 예시 (Page 0)


```
{
  "content": [ /* 기록 리스트 */ ],
  "pageable": { ... },
  "totalPages": 3,
  "totalElements": 25,
  "last": false,
  "size": 10,
  "number": 0,
  ...
}
```

5.5 기록 수정

- PUT /api/records/{id}
- **설명**: 해당 ID 의 기록을 수정합니다. 본인의 기록만 수정할 수 있습니다.
- **인증**: JWT 쿠키 필요

요청 예시

```
{
  "sleep": 6.0,
  "weight": 64
}
```

응답 예시

```
{
  "id": 3,
  "userId": 1,
  "sleep": 6.0,
  "weight": 64,
  "fat": 20.5,
  "muscle": 29.3,
  "bmr": 1400.5,
  "bmi": 22.1,
  "vai": 1.2,
  "date": "2025-07-10"
}
```

5.6 기록 삭제

- DELETE /api/records/{id}
- **설명**: 해당 ID 의 기록을 삭제합니다. 본인의 기록만 삭제할 수 있습니다.
- **인증**: JWT 쿠키 필요

5.7 그래프용 기록 조회

- GET /api/records/graph?duration=1w&category=체지방

- **설명**: 기간(1w, 1m, 3m)과 항목(체중, BMI, 수면시간 등)에 따른 사용자 기록을 그래프용으로 조회합니다.
- **인증**: JWT 쿠키 필요

응답 예시


```
{
  "myRecords": [
    { "date": "2025-07-03", "value": 20.5 },
    { "date": "2025-07-04", "value": 20.2 },
    { "date": "2025-07-05", "value": 20.0 }
  ],
  "category": "체지방"
}
```

5.8 전체 사용자 비교 분석

- GET /api/records/analysis
- **설명**: 내 최신 기록과 비슷한 연령대 평균 기록을 비교하여 분석 데이터를 제공합니다.
- **인증**: JWT 쿠키 필요

응답 예시


```
{
  "age": 24,
  "average": {
    "weight": 66.2,
    "fat": 21.4,
    "muscle": 29.1,
    "bmr": 1390.3,
    "bmi": 22.2,
    "vai": 1.3,
    "date": "2025-07-10"
  },
  "myrecord": {
    "weight": 65,
    "fat": 20.5,
    "muscle": 29.3,
    "bmr": 1400.5,
    "bmi": 22.1,
```

```
    "vai": 1.2,
    "date": "2025-07-10"
  }
}
```

5.9 내 최신 기록 조회

- GET /api/records/latest
- **설명**: 내 가장 최근 기록 1 건을 반환합니다.
- **인증**: JWT 쿠키 필요

응답 예시

```
{
  "id": 5,
  "userId": 1,
  "sleep": 7.5,
  "weight": 65,
  "fat": 20.5,
  "muscle": 29.3,
  "bmr": 1400.5,
  "bmi": 22.1,
  "vai": 1.2,
  "date": "2025-07-10"
}
```

6. Todolist API

6.1 Todolist 생성 또는 조회 (특정 날짜)

- POST /api/todolists/daily
- **설명:** 특정 날짜의 Todolist 가 없으면 새로 생성하고, 있으면 해당 Todolist 를 반환합니다.
- **인증:** JWT 쿠키 필요

요청 예시

```
{
  "date": "2025-07-10",
  "allclear": false
```

```
}
```

응답 예시

```
{
  "todolistId": 5,
  "userId": 2,
  "date": "2025-07-10",
  "allclear": false,
  "todos": [
    {
      "todoItemId": 301,
      "todolistId": 5,
      "todoitem": "30분 조깅",
      "tip": "스트레칭 잊지 말기",
      "youtubeId": "abcd1234",
      "youtubeTitle": "걷기 전 스트레칭",
      "complete": false,
      "date": "2025-07-10"
    }
  ],
  "diet": {
    "dietId": 15,
    "todolistId": 5,
    "breakfast": "귀리 쉐이크",
    "lunch": "닭가슴살 샐러드",
    "dinner": "두부 스테이크",
    "date": "2025-07-10"
  }
}
```

6.2 특정 사용자의 Todolist 전체 조회

- **GET** /api/todolists/user/{userId}
- **설명:** 특정 사용자의 모든 Todolist를 조회합니다.
- **인증:** JWT 쿠키 필요

응답 예시

```
[
```

```
  {
    "todolistId": 5,
    "userId": 2,
    "date": "2025-07-10",
    "allclear": false,
    "todos": [ ... ],
    "diet": { ... }
  },
  {
    "todolistId": 6,
    "userId": 2,
    "date": "2025-07-11",
    "allclear": true,
    "todos": [ ... ],
    "diet": { ... }
  }
]
```

6.3 특정 날짜의 Todolist 단건 조회

- **GET** /api/todolists/{userId}/date/{date}
- **설명:** 특정 사용자의 특정 날짜에 해당하는 Todolist를 조회합니다.
- **인증:** JWT 쿠키 필요

응답 예시

```
{
  "todolistId": 5,
  "userId": 2,
  "date": "2025-07-10",
  "allclear": false,
  "todos": [
    {
      "todoItemId": 301,
      "todolistId": 5,
      "todoitem": "30분 조깅",
      "tip": "스트레칭 잊지 말기",
      "youtubeId": "abcd1234",
      "youtubeTitle": "걷기 전 스트레칭",
      "complete": false,
```

```
      "date": "2025-07-10"
    }
  ],
  "diet": {
    "dietId": 15,
    "todolistId": 5,
    "breakfast": "귀리 쉐이크",
    "lunch": "닭가슴살 샐러드",
    "dinner": "두부 스테이크",
    "date": "2025-07-10"
  }
}
```


7. Todo API

7.1 Todo 생성

- POST /api/todos
- **설명**: 사용자의 특정 날짜에 할 일을 추가합니다. 해당 날짜에 Todolist 가 없으면 자동 생성됩니다.
- **인증**: JWT 쿠키 필요

요청 예시


```
{
  "todoitem": "30 분 유산소 운동",
  "complete": false,
  "date": "2025-07-10"
}
```


응답 예시


```
{
  "todoItemId": 12,
  "todolistId": 5,
  "todoitem": "30 분 유산소 운동",
  "tip": null,
  "youtubeId": null,
  "youtubeTitle": null,
  "complete": false,
  "date": "2025-07-10"
```

```
}
```

7.2 주간 Todo 조회

- **GET** /api/todos/week/{userId}
- **설명**: 사용자의 최근 7 일치(오늘 포함) Todo 목록을 반환합니다.
- **인증**: JWT 쿠키 필요

응답 예시


```
[
  {
    "todoItemId": 11,
    "todoListId": 4,
    "todoitem": "물 충분히 마시기",
    "tip": null,
    "youtubeId": null,
    "youtubeTitle": null,
    "complete": true,
    "date": "2025-07-09"
  },
  {
    "todoItemId": 12,
    "todoListId": 5,
    "todoitem": "30 분 유산소 운동",
    "tip": null,
    "youtubeId": null,
    "youtubeTitle": null,
    "complete": false,
    "date": "2025-07-10"
  }
]
```

7.3 월간 투두 조회

- **GET** /api/todos/month?ym={yyyy-MM}
- **설명**: JWT 인증된 사용자의 지정 월(예: 2025-07)에 해당하는 Todo 목록을 조회합니다.
- **인증**: JWT 쿠키 필요

응답 예시


```
[
  {
    "todoItemId": 7,
    "todoListId": 2,
    "todoItem": "아침 스트레칭",
    "tip": null,
    "youtubeId": null,
    "youtubeTitle": null,
    "complete": true,
    "date": "2025-07-03"
  }
]
```

7.4 투두 수정

- PATCH /api/todos/{id}
- **설명**: 투두 항목을 수정합니다.
- **인증**: JWT 쿠키 필요

요청 예시


```
{
  "todoItem": "아침 스트레칭",
  "complete": true
}
```


응답 예시


```
{
  "todoItemId": 7,
  "todoListId": 2,
  "todoItem": "아침 스트레칭",
  "tip": null,
  "youtubeId": null,
  "youtubeTitle": null,
  "complete": true,
  "date": "2025-07-03"
}
```

7.5 투두 삭제

- DELETE /api/todos/{id}
- **설명**: 투두 항목을 삭제합니다.
- **인증**: JWT 쿠키 필요

8. Diet API

8.1 식단 ID 로 식단 조회

- GET /api/diet/dietid/{dietId}
- 식단 ID 를 기반으로 단일 식단 데이터를 조회합니다.

Path Variable

이름	타입	설명
dietId	Long	조회할 식단 ID

응답 예시

```
{
  "dietId": 3,
  "todolistId": 5,
  "breakfast": "삶은 계란 2 개",
  "lunch": "닭가슴살 샐러드",
  "dinner": "현미밥과 된장국",
  "date": "2024-07-10"
}
```

8.2 Todolist ID 로 식단 조회

- GET /api/diet/todolistid/{todolistId}
- 특정 Todolist 에 연결된 식단 정보를 조회합니다.

Path Variable

이름	타입	설명

todolistId	Long	할 일 목록 ID

응답 예시


```
{
  "dietId": 5,
  "todolistId": 7,
  "breakfast": "요거트",
  "lunch": "샐러드",
  "dinner": "고구마 + 닭가슴살",
  "date": "2024-07-08"
}
```

8.3 월별 식단 조회

- GET /api/diet/month?ym=YYYY-MM
- 로그인한 사용자의 월별 식단 데이터를 조회합니다.

Query Parameter

이름	타입	설명
ym	String	조회할 월 (YYYY-MM)

응답 예시 (배열)


```
[
  {
    "dietId": 10,
    "todolistId": 13,
    "breakfast": "고구마",
    "lunch": "현미밥과 오징어볶음",
    "dinner": "닭가슴살 도시락",
    "date": "2024-07-01"
  },
  {
    "dietId": 11,
    "todolistId": 14,
    "breakfast": "두유",
    "lunch": "닭가슴살 샐러드",
```

```
    "dinner": "두부김치",
    "date": "2024-07-02"
  }
]
```

9. HTTP 상태 코드

코드	설명	사용 예시 및 의미
200	성공	요청이 정상적으로 처리됨(GET, POST, PATCH, DELETE 등 모두 해당)
400	잘못된 요청	클라이언트 요청 파라미터 오류, 유효성 검사 실패 등 (@Valid 실패 포함)
401	인증 실패	JWT 토큰 없음/만료/잘못됨 등 인증 헤더 문제 발생 시
403	권한 없음	로그인은 했으나 권한이 부족한 경우 (예: ROLE_USER 가 ADMIN API 접근)
404	리소스 없음	요청한 데이터나 API 엔드포인트가 존재하지 않음
409	충돌	이미 존재하는 리소스 중복 등록 시 (ex. 이메일 중복 회원가입 등)
500	서버 오류	예기치 않은 내부 서버 오류, NullPointerException 등 시스템 문제 발생 시