

API 문서 - 이룸오더

정보

- **API 버전:** v1.0.0
- **설명:** 이룸오더 API.

서버

- **기본 URL:** `http://localhost:8080`
- **설명:** 생성된 서버 URL.

태그

- **Order Management:** 주문 관리를 위한 API
- **Payment API:** 결제 처리를 위한 API
- **routine-controller:** 루틴 관리를 위한 API
- **order-mgmt-controller:** 주문 상태 관리를 위한 API
- **menu-controller:** 메뉴 관리를 위한 API
- **category-controller:** 카테고리 관리를 위한 API
- **auth-controller:** 사용자 인증을 위한 API
- **fcm-controller:** 푸시 알림 관리를 위한 API
- **sales-controller:** 판매 관리를 위한 API

routine-controller

기존 루틴 업데이트

EndPoint: `PUT /api/users/{userId}/routines/{routineId}`

태그: `routine-controller`

요약: 루틴 ID를 기반으로 기존 루틴의 세부 정보를 업데이트합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
<code>routineId</code>	<code>path</code>	업데이트할 루틴의 ID	<code>true</code>	<code>integer</code>	<code>1</code>

요청 본문

- **콘텐츠 타입:** `application/json`
- **스키마:** `RoutineDto`

응답

상태 코드	설명	스키마
200	루틴이 성공적으로 업데이트됨	RoutineResponseDto
404	루틴을 찾을 수 없음	RoutineResponseDto
400	유효하지 않은 입력	RoutineResponseDto

루틴 삭제

EndPoint: DELETE /api/users/{userId}/routines/{routineId}

태그: routine-controller

요약: 특정 사용자의 루틴 ID를 기반으로 루틴을 삭제합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
routineId	path	삭제할 루틴의 ID	true	integer	1
userId	path	루틴을 소유한 사용자의 ID	true	integer	123

응답

상태 코드	설명
204	루틴이 성공적으로 삭제됨
404	루틴을 찾을 수 없음
403	해당 루틴을 삭제할 권한이 없는 사용자

새 루틴 생성

EndPoint: POST /api/users/{userId}/routines/add

태그: routine-controller

요약: 사용자에게 새로운 루틴을 추가합니다.

요청 본문

- **콘텐츠 타입:** application/json
- **스키마:** RoutineDto

응답

상태 코드	설명	스키마
201	루틴이 성공적으로 생성됨	RoutineResponseDto
400	유효하지 않은 입력	RoutineResponseDto

사용자별 루틴 조회

EndPoint: GET /api/users/{userId}/routines

태그: routine-controller

요약: 특정 사용자 ID에 해당하는 모든 루틴을 조회합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
userId	path	루틴을 조회할 사용자의 ID	true	integer	123

응답

상태 코드	설명	스키마
200	루틴을 성공적으로 조회함	배열 RoutineResponseDto
404	사용자 또는 루틴을 찾을 수 없음	배열 RoutineResponseDto

스키마

RoutineDto

- 업데이트할 루틴의 세부 정보를 나타냅니다.

RoutineResponseDto

- 루틴 작업에 대한 응답을 나타냅니다.

Order Management

주문 생성

EndPoint: POST /orders/{userId}/order

태그: Order Management

요약: 제공된 세부 정보를 기반으로 새 주문을 생성합니다.

요청 본문

- 콘텐츠 타입: application/json
- 스키마: OrderRequestDto

응답

상태 코드	설명	스키마
201	주문이 성공적으로 생성됨	OrderResponseDto

상태 코드	설명	스키마
400	유효하지 않은 입력	OrderResponseDto

주문 상태 업데이트

EndPoint: POST /orderMgmt/orderStatus
태그: order-mgmt-controller
요약: 주어진 주문 ID와 상태를 기반으로 주문 상태를 업데이트합니다.

매개변수

이름	위치	설명	필수 여부	타입	값 예시
orderId	query	상태를 업데이트할 주문의 ID	true	integer	123
status	query	새로운 주문 상태	true	string	"NEW_ORDER", "WAITING", "COMPLETED" 등

응답

상태 코드	설명	스키마
200	성공	object

주문 확인

EndPoint: POST /orderMgmt/checkOrder
태그: order-mgmt-controller
요약: 주어진 주문 ID를 기반으로 주문을 확인합니다.

매개변수

이름	위치	설명	필수 여부	타입	값 예시
orderId	query	확인할 주문의 ID	true	integer	123

응답

상태 코드	설명	스키마
200	성공	object

스키마

OrderRequestDto

- 주문 생성 요청에서 사용되는 데이터의 스키마를 나타냅니다.

OrderResponseDto

- 주문 작업에 대한 응답 데이터를 나타냅니다.

Payment API

결제 요청 준비

EndPoint: POST /order/pay/ready

태그: Payment API

요약: 결제 요청에 필요한 tid를 생성합니다.

요청 본문

- 콘텐츠 타입: application/json
- 스키마: PayOrderForm

응답

상태 코드	설명	스키마
200	결제 준비 완료	PayReadyResponse
400	사용자 ID 불일치	PayReadyResponse
500	서버 오류	PayReadyResponse

결제 완료

EndPoint: POST /order/pay/approval

태그: Payment API

요약: 사용자의 인증 후 결제를 승인합니다.

요청 본문

- 콘텐츠 타입: application/json
- 스키마: PayApproveRequest

응답

상태 코드	설명	스키마
200	결제 승인 성공	PayApproveResponse
400	제공된 입력값이 유효하지 않음	PayApproveResponse
500	내부 서버 오류	PayApproveResponse

스키마

PayOrderForm

- 결제 요청을 준비하기 위해 필요한 데이터의 스키마를 나타냅니다.

PayReadyResponse

- 결제 요청 준비 작업의 응답 데이터를 나타냅니다.

PayApproveRequest

- 결제 승인을 요청할 때 필요한 데이터의 스키마를 나타냅니다.

PayApproveResponse

- 결제 승인의 응답 데이터를 나타냅니다.

auth-controller

이메일 인증

EndPoint: POST /auth/verifyEmail
태그: auth-controller
요약: 사용자가 받은 인증 코드를 검증합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
email	query	인증할 이메일 주소	true	string	user@example.com
code	query	사용자가 받은 인증 코드	true	string	1234

응답

상태 코드	설명	스키마
200	이메일 인증이 완료되었습니다.	string
400	인증 번호가 올바르지 않습니다.	string

회원가입

EndPoint: POST /auth/signUp
태그: auth-controller
요약: 새로운 사용자를 등록합니다.

요청 본문

- 콘텐츠 타입: `application/json`
- 스키마: `SignUpRequest`

응답

상태 코드	설명	스키마
201	회원가입 완료	string
400	회원가입 실패	string

이메일 인증 코드 전송

EndPoint: `POST /auth/sendEmailVerification`

태그: `auth-controller`

요약: 사용자의 이메일로 인증 코드를 전송합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
email	query	인증 코드를 받을 이메일 주소	true	string	user@example.com

응답

상태 코드	설명	스키마
200	인증 번호가 이메일로 전송되었습니다.	string
400	잘못된 요청입니다.	string

로그인

EndPoint: `POST /auth/login`

태그: `auth-controller`

요약: 사용자의 아이디와 비밀번호로 로그인합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
id	query	사용자의 아이디	true	string	user123
password	query	사용자의 비밀번호	true	string	password

응답

상태 코드	설명	스키마
200	로그인 성공	integer

상태 코드	설명	스키마
401	로그인 실패	integer

아이디 중복 체크

EndPoint: POST /auth/checkId

태그: auth-controller

요약: 사용 가능한 아이디인지 확인합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
id	query	확인할 아이디	true	string	user123

응답

상태 코드	설명	스키마
200	사용 가능한 아이디입니다.	string
400	아이디가 이미 존재합니다.	string

스키마

SignUpRequest

- 회원가입 시 필요한 데이터를 나타냅니다.

fcm-controller

FCM 토큰 등록

EndPoint: POST /api/fcm/register-token

태그: fcm-controller

요약: FCM 토큰을 등록합니다.

요청 본문

- 콘텐츠 타입: application/json
- 스키마: TokenRequest

응답

상태 코드	설명	스키마
200	OK	string

스키마

RoutineDto

- 새 루틴을 생성하기 위한 데이터의 스키마를 나타냅니다.

RoutineResponseDto

- 루틴 생성 또는 작업에 대한 응답 데이터를 나타냅니다.

TokenRequest

- FCM 토큰 등록 요청 시 필요한 데이터를 나타냅니다.

menu-controller

특정 메뉴 조회

EndPoint: GET /menu/getOneMenu

태그: menu-controller

요약: 메뉴 ID를 기반으로 메뉴를 조회합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
menuId	query	조회할 메뉴의 ID	true	integer	1

응답

상태 코드	설명	스키마
200	메뉴 조회 완료	MenuEntity
400	잘못된 요청입니다.	MenuEntity

카테고리 별 메뉴 조회

EndPoint: GET /menu/getMenu

태그: menu-controller

요약: 카테고리 ID를 기반으로 메뉴를 조회합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
categoryId	query	조회할 카테고리의 ID	true	integer	1

응답

상태 코드	설명	스키마
200	메뉴 조회 완료	배열 MenuEntity
400	잘못된 요청입니다.	배열 MenuEntity

메뉴 삭제

EndPoint: DELETE /menu/deleteMenu

태그: menu-controller

요약: 메뉴 ID를 기반으로 메뉴를 삭제합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
menuId	query	삭제할 메뉴의 ID	true	integer	1

응답

상태 코드	설명	스키마
200	메뉴 삭제 완료	string
404	메뉴를 찾을 수 없음	string

스키마

MenuEntity

- 사용자의 주문 정보를 나타냅니다.

category-controller

특정 카테고리 조회

EndPoint: GET /category/getCategory

태그: category-controller

요약: 카테고리 ID를 기반으로 카테고리를 조회합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
categoryId	query	조회할 카테고리의 ID	true	integer	1

응답

상태 코드	설명	스키마
200	카테고리 조회 완료	CategoryEntity
400	잘못된 요청입니다.	CategoryEntity

전체 카테고리 조회

EndPoint: GET /category/getAllCategory

태그: category-controller

요약: 모든 카테고리를 조회합니다.

응답

상태 코드	설명	스키마
200	전체 카테고리 조회 완료	배열 CategoryEntity
400	잘못된 요청입니다.	배열 CategoryEntity

카테고리 삭제

EndPoint: DELETE /category/deleteCategory

태그: category-controller

요약: 카테고리 ID를 기반으로 카테고리를 삭제합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
categoryId	query	삭제할 카테고리의 ID	true	integer	1

응답

상태 코드	설명	스키마
200	카테고리 삭제 완료	string
400	잘못된 요청입니다.	string

스키마

CategoryEntity

- 카테고리 정보를 나타냅니다.

sales-controller

어제 매출 조회

EndPoint: GET /api/sales

태그: sales-controller

요약: 특정 날짜와 사용자 ID에 해당하는 매출 데이터를 조회합니다.

매개변수

이름	위치	설명	필수 여부	타입	예시
date	query	매출 데이터를 조회할 날짜	true	string	2023-11-30
userId	query	데이터를 요청하는 사용자 ID	true	integer	12345

응답

상태 코드	설명	스키마
200	매출 데이터를 성공적으로 조회함	SalesResponse
400	유효하지 않은 입력 매개변수	SalesResponse
403	데이터를 조회할 권한이 없는 사용자	SalesResponse
500	내부 서버 오류	SalesResponse

스키마

SalesResponse

- 매출 데이터를 나타냅니다.

스키마

LocalTime

- 설명:** 루틴이 실행될 시간
- 타입:** object

속성

이름	타입	형식	설명	예시
hour	integer	int32	시간	9
minute	integer	int32	분	0
second	integer	int32	초	0
nano	integer	int32	나노 초	0

MenuDetailDto

- **설명:** 메뉴 옵션
- **타입:** `object`

속성

이름	타입	예시	설명
<code>useCup</code>	<code>string</code>	<code>TakeOut</code>	컵 사용 여부
<code>addShot</code>	<code>boolean</code>	<code>true</code>	샷 추가 여부
<code>addVanilla</code>	<code>boolean</code>	<code>false</code>	바닐라 추가 여부
<code>addHazelnut</code>	<code>boolean</code>	<code>true</code>	헤이즐넛 추가 여부
<code>light</code>	<code>boolean</code>	<code>true</code>	라이트 여부

RoutineDto

- **설명:** 생성할 루틴의 세부 정보
- **타입:** `object`
- **필수 항목:** `isActivated`, `menuId`, `routineDays`, `routineTime`, `userId`

속성

이름	타입	설명	예시
<code>userId</code>	<code>integer</code>	사용자 ID (long)	<code>12345</code>
<code>menuId</code>	<code>integer</code>	메뉴 ID	<code>10</code>
<code>menuOptions</code>	<code>object</code>	메뉴 옵션	<code>MenuDetailDto</code>
<code>routineDays</code>	<code>array</code>	루틴 실행 요일	<code>["Mon", "Wed", "Fri"]</code>
<code>routineTime</code>	<code>object</code>	루틴 실행 시간	<code>LocalTime</code>
<code>isActivated</code>	<code>boolean</code>	알람 활성화 여부	<code>true</code>

RoutineResponseDto

- **설명:** 루틴 생성 또는 조회에 대한 응답 데이터
- **타입:** `object`

속성

이름	타입	설명	예시
<code>routineId</code>	<code>integer</code>	루틴 ID	<code>1</code>
<code>userId</code>	<code>integer</code>	사용자 ID	<code>12345</code>
<code>menuId</code>	<code>integer</code>	메뉴 ID	<code>10</code>

이름	타입	설명	예시
menuName	string	메뉴명	아이스 아메리카노
menuDetailId	integer	메뉴 세부 정보 ID	5
menuDetail	object	메뉴 옵션	MenuDetailDto
routineDays	array	루틴 실행 요일	["Mon", "Wed", "Fri"]
routineTime	object	루틴 실행 시간	LocalTime
activated	boolean	활성화 여부	true

OrderMenuDto

- **설명:** 주문 시 포함된 메뉴의 세부 정보
- **타입:** object

속성

이름	타입	설명	예시
menuId	integer	메뉴 ID	1
quantity	integer	수량	2
menuOptions	object	메뉴 옵션	MenuDetailDto

OrderRequestDto

- **설명:** 생성할 주문의 세부 정보
- **타입:** object

속성

이름	타입	형식	설명	예시
userId	integer	int32	사용자 ID	1
totalPrice	integer	int32	총 주문 금액	3200
pickUp	object		픽업 시간	LocalTime
orderMenuOptions	array		주문 메뉴 세부 정보	배열 OrderMenuDto

OrderResponseDto

- **설명:** 주문 생성 또는 조회에 대한 응답 데이터
- **타입:** object

속성

이름	타입	형식	설명	예시
orderId	integer	int32	주문 ID	1
userId	integer	int32	사용자 ID	1
totalPrice	integer	int32	총 주문 금액	3000
orderStatus	string		주문 상태	WAITING
pickUp	object		픽업 시간	LocalTime
payment	string	date-time	결제 시간	2023-12-01T12:00:00
orderMenuOptions	array		주문 메뉴 세부 정보	배열 OrderMenuDto

PayOrderForm

- **설명:** 결제 준비 요청 객체
- **타입:** object
- **필수 항목:** totalPrice

속성

이름	타입	설명	예시
cid	string	상점 코드 (테스트용)	TC00NETIME
order_id	integer	주문 ID	1001
user_id	integer	사용자 ID	1
item_name	string	상품 이름	Product Name
quantity	integer	상품 수량	2
totalPrice	integer	상품 총 금액	20000
tax_free_amount	integer	면세 금액	0

PayReadyResponse

- **설명:** 결제 준비 응답 객체
- **타입:** object

속성

이름	타입	설명	예시
tid	string	트랜잭션 ID	TID123456789
next_redirect_pc_url	string	PC 리다이렉트 URL	https://pay.kakao.com/redirect

이름	타입	설명	예시
next_redirect_mobile_url	string	모바일 리다이렉트 URL	https://m.pay.kakao.com/redirect

PayApproveRequest

- **설명:** 결제 승인 요청 객체
- **타입:** object

속성

이름	타입	설명	예시
pg_token	string	승인 인증 토큰 (?pg_token=~~ URL에서 확인 가능)	abcdef123456
userId	integer	사용자 ID	1
orderId	integer	주문 ID	1001
cid	string	상점 코드 (테스트용)	TC00NETIME

PayApproveResponse

- **설명:** 결제 승인 응답 객체
- **타입:** object

속성

이름	타입	설명	예시
aid	string	요청 고유 ID	AID123456789
tid	string	트랜잭션 ID	TID123456789
cid	string	상점 코드	TC00NETIME
partner_order_id	string	상점 주문 ID	1001
partner_user_id	string	상점 사용자 ID	12345
payment_method_type	string	결제 방법	CARD
item_name	string	상품 이름	Product Name
item_code	string	상품 코드	P001
quantity	string	상품 수량	2
created_at	string	요청 생성 시간	2023-12-01T12:00:00
approved_at	string	결제 승인 시간	2023-12-01T12:10:00
payload	string	승인 관련 추가 정보	Additional info

MenuRequest

- **설명:** 생성할 메뉴의 정보
- **타입:** `object`

속성

이름	타입	설명	예시
<code>name</code>	<code>string</code>	메뉴 이름	아이스 아메리카노
<code>price</code>	<code>integer</code>	메뉴 가격	3200
<code>categoryId</code>	<code>integer</code>	카테고리 ID	1

SignUpRequest

- **설명:** 회원가입 요청 정보
- **타입:** `object`

속성

이름	타입	설명	예시
<code>id</code>	<code>string</code>	사용자 아이디	user123
<code>password</code>	<code>string</code>	사용자 비밀번호	password123
<code>email</code>	<code>string</code>	사용자 이메일 주소	user@example.com

TokenRequest

- **설명:** FCM 토큰 등록 요청 객체
- **타입:** `object`

속성

이름	타입	설명	예시
<code>fcmToken</code>	<code>string</code>	FCM 토큰	abcdef123456
<code>id</code>	<code>string</code>	사용자 아이디	user123

OrderDto

- **설명:** 주문 메뉴 세부 정보
- **타입:** `object`

속성

이름	타입	설명	예시
menuId	integer	메뉴 ID	1
menuName	string	메뉴명	Americano
quantity	integer	수량	2
useCup	string	컵 사용 여부	true
addShot	boolean	샷 추가 여부	false
addVanilla	boolean	바닐라 추가 여부	true
addHazelnut	boolean	헤이즐넛 추가 여부	false
light	boolean	라이트 여부	true

OrdersCheckResponseDto

- **설명:** 주문 상태 및 세부 정보 응답
- **타입:** object

속성

이름	타입	설명	예시
orderId	integer	주문 ID	1
userId	integer	사용자 ID	1
totalPrice	integer	총 주문 금액	3000
orderStatus	string	주문 상태	WAITING
pickUp	object	픽업 시간	LocalTime
orderMenuOptions	array	주문 메뉴 세부 정보	배열 OrderDto

CategoryEntity

- **설명:** 카테고리 정보
- **타입:** object

속성

이름	타입	설명	예시
id	integer	카테고리 ID	1
name	string	카테고리 이름	음료

MenuEntity

- **설명:** 메뉴 정보
- **타입:** `object`

속성

이름	타입	설명	예시
<code>menuId</code>	<code>integer</code>	메뉴 ID	<code>1</code>
<code>name</code>	<code>string</code>	메뉴명	<code>아이스 아메리카노</code>
<code>price</code>	<code>integer</code>	메뉴 가격	<code>3200</code>
<code>category</code>	<code>object</code>	카테고리	<code>CategoryEntity</code>

SalesResponse

- **설명:** 매출 응답 데이터
- **타입:** `object`

속성

이름	타입	설명	예시
<code>totalSales</code>	<code>integer</code>	총 매출	<code>50000</code>