

백오피스포탈 메뉴 관리 API 명세

백오피스포탈 메뉴 관리 API 명세

API 목록 (Quick Reference)

카테고리	Method	Endpoint	설명
클라이언트 (기존)	POST	/api/v1/backoffice-clients	클라이언트 생성
	GET	/api/v1/backoffice-clients	클라이언트 목록 조회
	GET	/api/v1/backoffice-clients/{clientId}	클라이언트 상세 조회
	PUT	/api/v1/backoffice-clients/{clientId}	클라이언트 수정
클라이언트 (v2)	GET	/api/v2/clients/authorized	권한 있는 클라이언트 목록 조회 (메뉴 포함 옵션)
메뉴 관리	GET	/api/v2/menus?clientId={clientId}	메뉴 목록 조회
	GET	/api/v2/menus/authorized?	사용자 권한 기반 메뉴 조회 (다중 클라이언트 지원)

		keycloakClientI ds={ids}	
	GET	/api/v2/menus/{ menuId}	메뉴 상세 조회
	PUT	/api/v2/menus? keycloakClientI d= {keycloakClient Id}	메뉴 일괄 Upsert
	DELETE	/api/v2/menus/{ menuId}	메뉴 삭제
메뉴 리소스	GET	/api/v2/menus/{ menuId}/resourc es	메뉴 리소스 조회
	PUT	/api/v2/menus/{ menuId}/resourc es	메뉴 리소스 수정
Resource 관리	GET	/api/v2/keycloa k/resources	Resource 목록 조회
	GET	/api/v2/keycloa k/resources/{id }	Resource 상세
	POST	/api/v2/keycloa k/resources	Resource 등록
	POST	/api/v2/keycloa k/resources/bat ch	Resource 일괄 생성

	PUT	/api/v2/keycloak/resources/{resourceId}	Resource 수정
	PATCH	/api/v2/keycloak/resources	Resource 일괄 수정
	DELETE	/api/v2/keycloak/resources/{resourceId}	Resource 삭제
Role 관리	GET	/api/v2/keycloak/roles	Role 목록 조회
	POST	/api/v2/keycloak/roles	Role 생성
	PUT	/api/v2/keycloak/roles/{roleId}	Role 수정
	DELETE	/api/v2/keycloak/roles/{roleId}	Role 삭제
Mock (개발용)	GET	/mock/v2/menus	Mock 메뉴 트리 조회
	GET	/mock/v2/menus/authorized	Mock 권한별 메뉴 조회
Migration (임시)	POST	/v2/migration/backoffice-client	Backoffice Client 마이그레이션 (RBAC → UMA 2.0)

상세 API 명세

- 클라이언트 & 메뉴 API 상세 - 클라이언트, 메뉴 관리, 메뉴 리소스, 권한 기반 메뉴 API
- Resource & Role API 상세 - Keycloak Resource, Role 관리 API
- Migration API 상세 - RBAC에서 UMA 2.0으로 마이그레이션 API (임시)

1. 개요

1.1 Base URL

개발 환경 (Dev)

```
1 https://backoffice-gateway.socar.me
```

운영 환경 (Production)

```
1 https://backoffice-gateway.socarcorp.co.kr
```

참고: 모든 API 엔드포인트는 Gateway에서 `/api` prefix가 추가됩니다.

예: 서버의 `/v2/menus` → Gateway를 통해 `/api/v2/menus` 로 노출

1.2 인증 방식

- **Type:** Bearer Token (JWT)
- **Header:** `Authorization: Bearer {access_token}`
- Keycloak에서 발급받은 JWT 토큰 사용

1.3 공통 Request Headers

```
1 Content-Type: application/json
2 Authorization: Bearer {access_token}
```

1.4 공통 Response 형식

Success Response

```
1 {
2   "success": true,
3   "data": { /* API별 응답 데이터 */ }
4 }
```

Error Response

에러 응답은 **SOCAR 표준**을 따릅니다. 상세 사항은 [Exception 처리 가이드](#)를 참고하세요.

```

1 {
2   "error": {
3     "code": 400,
4     "message": "잘못된 요청 파라미터입니다.",
5     "status": "BAD_REQUEST",
6     "details": [
7       {
8         "@type": "type.googleapis.com/google.rpc.ErrorInfo",
9         "reason": "INVALID_CLIENT_ID",
10        "domain": "backoffice-portal",
11        "metadata": {
12          "clientId": "parking-admin"
13        }
14      },
15      {
16        "@type": "type.googleapis.com/google.rpc.LocalizedMessage",
17        "locale": "ko-KR",
18        "message": "잘못된 요청 파라미터입니다."
19      }
20    ]
21  }
22 }

```

1.5 HTTP Status Codes

HTTP Status	Status Name	설명	사용 예시
200 OK	OK	성공	조회/수정 성공
201 Created	CREATED	리소스 생성 성공	신규 등록 완료
204 No Content	NO_CONTENT	성공 (응답 본문 없음)	삭제 성공
400 Bad Request	BAD_REQUEST	잘못된 요청 파라미터	Validation 실패, 선행 조건 실패
403 Forbidden	FORBIDDEN	권한 없음	접근 권한 부족
404 Not Found	NOT_FOUND	리소스 없음	Client/Resource 없음
500 Internal Server Error	INTERNAL_SERVER_ERROR	서버 내부 오류	예상치 못한 오류
504 Gateway Timeout	GATEWAY_TIMEOUT	타임아웃	요청 처리 시간 초과