

클라이언트 & 메뉴 API 상세

클라이언트 & 메뉴 API 상세

2. 클라이언트 API (기존 v1)

2.1 클라이언트 목록 조회

GET /api/v1/backoffice-clients

등록된 백오피스 클라이언트 목록을 조회합니다.

Query Parameters

Parameter	Type	Required	Description
condition	object	No	검색 조건
page	number	No	페이지 번호
size	number	No	페이지 크기

Response 200 OK

```
1 {
2   "success": true,
3   "data": {
4     "clients": [
5       {
6         "id": 2,
7         "clientId": "backoffice-admin",
8         "clientName": "백오피스 어드민",
9         "description": "백오피스 관리자 시스템",
10        "createdAt": "2025-01-15T09:00:00.000Z",
11        "updatedAt": "2025-01-15T09:00:00.000Z"
12      },
13      {
14        "id": 3,
15        "clientId": "partner-center",
16        "clientName": "파트너센터",
17        "description": "파트너 관리 시스템",
18        "createdAt": "2025-01-15T09:00:00.000Z",
19        "updatedAt": "2025-01-15T09:00:00.000Z"
20      }
21    ]
22  }
23 }
```

2.2 클라이언트 상세 조회

GET `/api/v1/backoffice-clients/{clientId}`

특정 클라이언트의 상세 정보를 조회합니다.

Path Parameters

Parameter	Type	Required	Description
clientId	number	Yes	백오피스 클라이언트 ID

Response 200 OK

```
1 {
2   "success": true,
3   "data": {
4     "id": 2,
5     "clientId": "backoffice-admin",
6     "clientName": "백오피스 어드민",
7     "description": "백오피스 관리자 시스템",
8     "createdAt": "2025-01-15T09:00:00.000Z",
9     "updatedAt": "2025-01-15T09:00:00.000Z"
10  }
11 }
```

Error Responses

404 Not Found

```
1 {
2   "error": {
3     "code": 404,
4     "message": "Client를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }
```

2.3 클라이언트 생성

POST `/api/v1/backoffice-clients`

새로운 백오피스 클라이언트를 생성합니다.

Request Body

```
1 {
2   "clientId": "new-backoffice",
3   "clientName": "새 백오피스",
4   "description": "새로운 백오피스 시스템"
5 }
```

Response 200 OK

```
1 {
2   "success": true
3 }
```

2.4 클라이언트 수정

PUT /api/v1/backoffice-clients/{clientId}

기존 백오피스 클라이언트 정보를 수정합니다.

Path Parameters

Parameter	Type	Required	Description
clientId	number	Yes	백오피스 클라이언트 ID

Request Body

```
1 {
2   "clientName": "수정된 백오피스",
3   "description": "수정된 설명"
4 }
```

Response 200 OK

```
1 {
2   "success": true
3 }
```

Error Responses

404 Not Found

```
1 {
2   "error": {
3     "code": 404,
4     "message": "Client를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }
```

2.5 권한 있는 클라이언트 목록 조회 (v2)

GET /api/v2/clients/authorized

JWT 토큰의 `resource_access` 클레임에서 롤을 하나라도 가진 클라이언트 목록을 조회합니다.

Query Parameters

Parameter	Type	Required	Description
includeMenus	boolean	No	메뉴 포함 여부 (기본 값: false)

Request Headers

Header	Required	Description
Authorization	Yes	Bearer {access_token}

Request Example

```
1 GET /api/v2/clients/authorized?includeMenus=true
2 Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response 200 OK (includeMenus=false)

```
1 {
2   "success": true,
3   "data": {
4     "clients": [
5       {
6         "id": 1,
7         "name": "피닉스",
8         "keycloakClientId": "phoenix-test1",
9         "type": "BACK_OFFICE",
10        "url": "https://phoenix.socar.kr",
11        "imageUrl": null,
12        "menus": []
13      },
14      {
15        "id": 2,
16        "name": "파트너센터",
17        "keycloakClientId": "partner-center",
18        "type": "BACK_OFFICE",
19        "url": "https://partner.socar.kr",
20        "imageUrl": null,
21        "menus": []
22      }
23    ]
24  }
25 }
```

Response 200 OK (includeMenus=true)

```
1 {
2   "success": true,
3   "data": {
4     "clients": [
5       {
6         "id": 1,
```

```

7      "name": "피닉스",
8      "keycloakClientId": "phoenix-test1",
9      "type": "BACK_OFFICE",
10     "url": "https://phoenix.socar.kr",
11     "imageUrl": null,
12     "menus": [
13         {
14             "id": 10,
15             "parentId": null,
16             "name": "대시보드",
17             "type": "ITEM",
18             "url": "/dashboard",
19             "displayOrder": 1,
20             "displayYn": true,
21             "privacyIncludeYn": false,
22             "locationIncludeYn": false,
23             "scopes": ["GET"],
24             "children": []
25         }
26     ]
27 }
28 ]
29 }
30 }

```

Response Fields

Field	Type	Description
id	number	백오피스 클라이언트 ID
name	string	클라이언트 이름
keycloakClientId	string	Keycloak 클라이언트 ID
type	string	접근 타입 (BACK_OFFICE 등)
url	string	클라이언트 접근 URL
imageUrl	string	클라이언트 이미지 URL
menus	array	권한 기반 메뉴 목록 (includeMenus=false면 빈 배열)

롤 필터링 규칙

1. JWT 토큰의 **resource_access** 클레임에서 각 클라이언트별 **roles** 배열 확인
2. **roles** 배열이 비어있지 않은 클라이언트만 반환
3. **socar-backoffice-portal** 클라이언트는 자체 포탈이므로 제외

4. DB에서 활성화된(activityYn=true) 클라이언트만 반환

3. 메뉴 관리 API

3.1 메뉴 목록 조회 (관리자용)

GET /api/v2/menus?keycloakClientId={keycloakClientId}

특정 클라이언트의 메뉴 목록을 조회합니다.

참고: 이 API는 메뉴 관리용이므로 모든 메뉴가 반환됩니다. 실제 사용자에게 보여줄 메뉴는 /api/v2/menus/authorized API를 사용하세요.

Query Parameters

Parameter	Type	Required	Description
keycloakClientId	string	Yes	Keycloak 클라이언트 ID (예: "phoenix2")
format	string	No	flat (기본값) 또는 tree

Response 200 OK (format=flat)

```
1 {
2   "success": true,
3   "data": {
4     "menus": [
5       {
6         "id": 10,
7         "parentId": null,
8         "name": "회원 관리",
9         "type": "GROUP",
10        "url": null,
11        "displayOrder": 1,
12        "description": "회원 관련 메뉴",
13        "displayYn": true,
14        "privacyIncludeYn": false,
15        "locationIncludeYn": false,
16        "createdAt": "2025-01-15T09:00:00.000Z",
17        "updatedAt": "2025-01-15T09:00:00.000Z"
18      },
19      {
20        "id": 11,
21        "parentId": 10,
22        "name": "사용자 관리",
23        "type": "ITEM",
24        "url": "/backoffice/users",
```

```

25     "displayOrder": 1,
26     "description": "사용자 관리 화면",
27     "displayYn": true,
28     "privacyIncludeYn": true,
29     "locationIncludeYn": false,
30     "createdAt": "2025-01-15T09:00:00.000Z",
31     "updatedAt": "2025-01-15T09:00:00.000Z"
32   }
33 ]
34 }
35 }

```

Response 200 OK (format=tree)

```

1  {
2    "success": true,
3    "data": {
4      "keycloakClientId": "phoenix2",
5      "clientName": "피닉스2",
6      "menus": [
7        {
8          "id": 10,
9          "parentId": null,
10         "name": "회원 관리",
11         "type": "GROUP",
12         "url": null,
13         "displayOrder": 1,
14         "description": "회원 관련 메뉴",
15         "displayYn": true,
16         "privacyIncludeYn": false,
17         "locationIncludeYn": false,
18         "children": [
19           {
20             "id": 11,
21             "parentId": 10,
22             "name": "사용자 관리",
23             "type": "ITEM",
24             "url": "/backoffice/users",
25             "displayOrder": 1,
26             "displayYn": true,
27             "privacyIncludeYn": true,
28             "locationIncludeYn": false,
29             "children": []
30           }
31         ]
32       }
33     ]
34   }
35 }

```

Error Responses

400 Bad Request

```

1  {
2    "error": {
3      "code": 400,
4      "message": "유효하지 않은 Client ID입니다.",

```

```
5     "status": "BAD_REQUEST",
6     "details": [...]
```

403 Forbidden

```
1 {
2   "error": {
3     "code": 403,
4     "message": "해당 작업을 수행할 권한이 없습니다.",
5     "status": "FORBIDDEN",
6     "details": [...]
```

404 Not Found

```
1 {
2   "error": {
3     "code": 404,
4     "message": "Client를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
```

3.2 메뉴 상세 조회

GET /api/v2/menus/{menuId}

특정 메뉴의 상세 정보를 조회합니다.

Path Parameters

Parameter	Type	Required	Description
menuId	number	Yes	메뉴 ID

Response 200 OK

```
1 {
2   "success": true,
3   "data": {
4     "id": 11,
5     "parentId": 10,
6     "name": "사용자 관리",
7     "type": "ITEM",
8     "url": "/backoffice/users",
9     "displayOrder": 1,
10    "description": "사용자 관리 화면",
11    "displayYn": true,
12    "privacyIncludeYn": true,
13    "locationIncludeYn": false,
```



```

14     "createdAt": "2025-01-15T09:00:00.000Z",
15     "updatedAt": "2025-01-15T09:00:00.000Z",
16     "resources": [
17       {
18         "id": 1,
19         "resourceId": "f7a4c8d3-9b2e-4f1a-8c7d-3e5f9a2b1c0d",
20         "resourceName": "user-management",
21         "displayName": "사용자 관리",
22         "scopes": ["GET", "POST", "PUT", "DELETE"]
23       }
24     ]
25   }
26 }

```

Error Responses

404 Not Found

```

1 {
2   "error": {
3     "code": 404,
4     "message": "메뉴를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }

```

3.3 메뉴 일괄 Upsert

PUT /api/v2/menus?keycloakClientId={keycloakClientId}

메뉴를 일괄 생성/수정/삭제합니다.

- **id** 가 있는 메뉴: 수정
- **id** 가 없는 메뉴: 생성
- **deleteIds** 에 포함된 메뉴: 삭제

Query Parameters

Parameter	Type	Required	Description
keycloakClientId	string	Yes	Keycloak 클라이언트 ID (예: "phoenix2")

Request Body

```

1 {
2   "menus": [
3     {
4       "id": 10,
5       "parentId": null,

```

```

6     "name": "회원 관리",
7     "type": "GROUP",
8     "url": null,
9     "displayOrder": 1,
10    "description": "회원 관련 메뉴",
11    "displayYn": true,
12    "privacyIncludeYn": false,
13    "locationIncludeYn": false
14  },
15  {
16    "id": 11,
17    "parentId": 10,
18    "name": "사용자 관리",
19    "type": "ITEM",
20    "url": "/backoffice/users",
21    "displayOrder": 1,
22    "description": "사용자 관리 화면",
23    "displayYn": true,
24    "privacyIncludeYn": true,
25    "locationIncludeYn": false
26  },
27  {
28    "parentId": 10,
29    "name": "신규 메뉴",
30    "type": "ITEM",
31    "url": "/backoffice/new",
32    "displayOrder": 2,
33    "displayYn": true
34  }
35 ],
36 "deleteIds": [15, 16]
37 }

```

Request Body Fields

Field	Type	Required	Description
menus	array	Yes	Upsert할 메뉴 목록
menus[].id	number	No	메뉴 ID (없으면 생성, 있으면 수정)
menus[].parentId	number	No	부모 메뉴 ID (null이면 최상위)
menus[].name	string	Yes	메뉴명
menus[].type	string	Yes	GROUP 또는 ITEM
menus[].url	string	Conditional	ITEM일 때 필수
menus[].displayOrder	number	Yes	표시 순서

menus[].description	string	No	설명
menus[].displayYn	boolean	No	노출 여부 (기본값: true)
menus[].privacyIncludeYn	boolean	No	개인정보 포함 여부 (기본값: false)
menus[].locationIncludeYn	boolean	No	위치정보 포함 여부 (기본값: false)
deleteIds	number[]	No	삭제할 메뉴 ID 목록

Validation Rules

1. **ITEM** 타입 메뉴: `url` 필수
2. **displayOrder**: 동일 레벨에서 중복 불가
3. **id**: 전체 트리에서 고유해야 함
4. **url**: ITEM 타입에서 필수, GROUP 타입에서는 null

Response 200 OK

```

1  {
2    "success": true,
3    "data": {
4      "menuGroupId": 1,
5      "created": 1,
6      "updated": 2,
7      "deleted": 2,
8      "results": [
9        { "id": 10, "action": "updated" },
10       { "id": 11, "action": "updated" },
11       { "id": 22, "action": "created" },
12       { "id": 15, "action": "deleted" },
13       { "id": 16, "action": "deleted" }
14     ]
15   }
16 }
```

Error Responses

400 Bad Request - Validation Error

```

1  {
2    "error": {
3      "code": 400,
4      "message": "Validation 실패",
5      "status": "BAD_REQUEST",
6      "details": [
7        { "field": "menus[2].url", "message": "ITEM 타입은 url이 필수입니다." }
8      ]
9    }
10 }
```

```
9   }
10 }
```

403 Forbidden

```
1 {
2   "error": {
3     "code": 403,
4     "message": "메뉴 관리 권한이 없습니다.",
5     "status": "FORBIDDEN",
6     "details": [...]
7   }
8 }
```

404 Not Found

```
1 {
2   "error": {
3     "code": 404,
4     "message": "Client를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }
```

3.4 메뉴 삭제

DELETE /api/v2/menus/{menuId}

특정 메뉴를 삭제합니다.

Path Parameters

Parameter	Type	Required	Description
menuId	number	Yes	메뉴 ID

Query Parameters

Parameter	Type	Required	Description
cascade	boolean	No	하위 메뉴 포함 삭제 (기본값: false)

Response 200 OK

```
1 {
2   "success": true,
3   "data": {
4     "deletedId": 12,
5     "deletedChildren": []
6   }
7 }
```

```
6   }
7 }
```

Error Responses

400 Bad Request (하위 메뉴 존재 시)

```
1 {
2   "error": {
3     "code": 400,
4     "message": "하위 메뉴가 존재합니다. cascade=true로 요청하거나 하위 메뉴를 먼저 삭제하세요.",
5     "status": "BAD_REQUEST",
6     "details": {
7       "childrenCount": 3
8     }
9   }
10 }
```

403 Forbidden

```
1 {
2   "error": {
3     "code": 403,
4     "message": "메뉴 삭제 권한이 없습니다.",
5     "status": "FORBIDDEN",
6     "details": [...]
7   }
8 }
```

404 Not Found

```
1 {
2   "error": {
3     "code": 404,
4     "message": "메뉴를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }
```

4. 메뉴 리소스 API

4.1 메뉴 리소스 조회

GET `/api/v2/menus/{menuId}/resources`

특정 메뉴에 연결된 리소스 목록을 조회합니다.

Path Parameters

Parameter	Type	Required	Description
menuId	number	Yes	메뉴 ID

Response 200 OK

```
1 {
2   "success": true,
3   "data": {
4     "menuId": 11,
5     "resources": [
6       {
7         "id": 1,
8         "resourceId": "f7a4c8d3-9b2e-4f1a-8c7d-3e5f9a2b1c0d",
9         "resourceName": "user-management",
10        "displayName": "사용자 관리",
11        "scopes": ["GET", "POST", "PUT", "DELETE"]
12      }
13    ]
14  }
15 }
```

Error Responses

404 Not Found

```
1 {
2   "error": {
3     "code": 404,
4     "message": "메뉴를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }
```

4.2 메뉴 리소스 수정

PUT /api/v2/menus/{menuId}/resources

특정 메뉴에 연결된 리소스를 수정합니다.

Side Effect - 개인정보/위치정보 플래그 자동 업데이트:

- 연결된 리소스 중 `personalInfoHandleYn=true` 인 리소스가 있으면 해당 메뉴와 모든 상위 메뉴의 `privacyIncludeYn` 이 `true` 로 설정됩니다.
- 연결된 리소스 중 `locationInfoHandleYn=true` 인 리소스가 있으면 해당 메뉴와 모든 상위 메뉴의 `locationIncludeYn` 이 `true` 로 설정됩니다.
- 기존 리소스가 제거되거나 교체되면 플래그가 재계산됩니다. (해당 속성을 가진 리소스가 없으면 `false` 로 변경)
- 상위 메뉴의 플래그는 하위 메뉴들의 플래그를 기준으로 재계산됩니다. (하위 메뉴 중 하나라도 `true` 면 상위도 `true`)

Path Parameters

Parameter	Type	Required	Description
menuId	number	Yes	메뉴 ID

Query Parameters

Parameter	Type	Required	Description
keycloakClientId	string	Yes	Keycloak 클라이언트 ID

Request Body

```
1 {
2   "resources": [
3     { "resourceId": "f7a4c8d3-9b2e-4f1a-8c7d-3e5f9a2b1c0d" },
4     { "resourceId": "a1b2c3d4-5e6f-7890-abcd-ef1234567890" }
5   ]
6 }
```

Response 200 OK

```
1 {
2   "success": true
3 }
```

400 Bad Request

```

1 {
2   "error": {
3     "code": 400,
4     "message": "유효하지 않은 Resource ID가 포함되어 있습니다.",
5     "status": "BAD_REQUEST",
6     "details": [...]
7   }
8 }

```

403 Forbidden

```

1 {
2   "error": {
3     "code": 403,
4     "message": "메뉴 리소스 수정 권한이 없습니다.",
5     "status": "FORBIDDEN",
6     "details": [...]
7   }
8 }

```

404 Not Found

```

1 {
2   "error": {
3     "code": 404,
4     "message": "메뉴를 찾을 수 없습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }

```

5. 사용자 권한 기반 메뉴 API

5.1 권한 기반 메뉴 조회

GET /api/v2/menus/authorized

JWT Access Token에서 사용자 ID(sub claim)를 추출하고, Keycloak Policy Evaluation API를 통해 해당 사용자가 접근 가능한 리소스를 확인하여 권한 있는 메뉴만 반환합니다.

Query Parameters

Parameter	Type	Required	Description
keycloakClientIds	string	Yes	Keycloak 클라이언트 ID (콤마로 구분하여 여러 개 지정 가능, 예:

			"phoenix2,another-client")
--	--	--	----------------------------

Request Headers

Header	Required	Description
Authorization	Yes	Bearer {access_token}

Request Example (단일 클라이언트)

```
1 GET /api/v2/menus/authorized?keycloakClientIds=phoenix2
2 Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

Request Example (다중 클라이언트)

```
1 GET /api/v2/menus/authorized?keycloakClientIds=phoenix2,another-client
2 Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response 200 OK

```
1 {
2   "success": true,
3   "data": [
4     {
5       "keycloakClientId": "phoenix2",
6       "clientName": "피닉스2",
7       "accessUrl": "https://phoenix.socar.kr",
8       "menus": [
9         {
10          "id": 10,
11          "parentId": null,
12          "name": "회원 관리",
13          "type": "GROUP",
14          "url": null,
15          "displayOrder": 1,
16          "description": "회원 관련 메뉴",
17          "displayYn": true,
18          "privacyIncludeYn": false,
19          "locationIncludeYn": false,
20          "scopes": null,
21          "children": [
22            {
23              "id": 11,
24              "parentId": 10,
25              "name": "사용자 관리",
26              "type": "ITEM",
27              "url": "/backoffice/users",
28              "displayOrder": 1,
29              "description": "사용자 관리 화면",
30              "displayYn": true,
31              "privacyIncludeYn": true,
32              "locationIncludeYn": false,
33              "scopes": ["GET", "POST", "PUT", "DELETE"],
```

```

34         "children": []
35     }
36 ]
37 }
38 ]
39 },
40 {
41     "keycloakClientId": "another-client",
42     "clientName": "다른 클라이언트",
43     "accessUrl": "https://another.socar.kr",
44     "menus": []
45 }
46 ]
47 }

```

Response Fields

Field	Type	Description
keycloakClientId	string	Keycloak 클라이언트 ID
clientName	string	클라이언트 이름 (백오피스 클라이언트에 등록된 이름)
accessUrl	string	백오피스 클라이언트 접근 URL

Response Fields (scopes)

Field	Type	Description
scopes	string[] null	해당 메뉴에서 사용자가 허용된 HTTP 메서드 목록. GROUP 타입은 null, ITEM 타입은 권한에 따라 GET/POST/PUT/DELETE 등이 포함됨. UI에서 버튼 활성화/비활성화에 활용

권한 필터링 규칙

1. JWT 토큰에서 **sub** claim으로 사용자 ID 추출
2. Keycloak Policy Evaluation API를 호출하여 해당 사용자가 접근 가능한 Resource ID 목록 조회
3. ITEM 타입 메뉴에 연결된 Resource와 비교
4. **OR 조건**: resources 중 하나라도 권한이 있으면 메뉴 표시
5. 모든 resources에 권한 없는 ITEM 메뉴는 응답에서 제외
6. 하위 메뉴가 모두 권한 없으면 상위 GROUP도 제외

7. JWT 토큰에서 userId 추출 실패 시 빈 메뉴 목록 반환

Error Responses

404 Not Found - 클라이언트가 존재하지 않을 때

```
1 {
2   "error": {
3     "code": 404,
4     "message": "keycloakClientId가 phoenix2인 클라이언트가 존재하지 않습니다.",
5     "status": "NOT_FOUND",
6     "details": [
7       {
8         "@type": "type.googleapis.com/google.rpc.ErrorInfo",
9         "reason": "BACKOFFICE_CLIENT_NOT_FOUND",
10        "domain": "menu",
11        "metadata": {
12          "keycloak_client_id": "phoenix2"
13        }
14      }
15    ]
16  }
17 }
```

404 Not Found - 메뉴 그룹이 존재하지 않을 때

```
1 {
2   "error": {
3     "code": 404,
4     "message": "clientId가 1인 메뉴 그룹이 존재하지 않습니다.",
5     "status": "NOT_FOUND",
6     "details": [...]
7   }
8 }
```

8. Mock API (개발용)

주의: Mock API는 **dev**, **local** 프로파일에서만 사용 가능합니다. 프론트엔드 개발 시 백엔드 연동 없이 테스트할 수 있습니다.

8.1 Mock 메뉴 트리 조회

GET /mock/v2/menus

Mock 메뉴 트리를 반환합니다. Keycloak 연동 없이 프론트엔드 개발이 가능합니다.

Query Parameters

Parameter	Type	Required	Description
-----------	------	----------	-------------

keycloakClientId	string	Yes	클라이언트 ID (phoenix , portal, test, backoffice 지 원)
------------------	--------	-----	--

Response 200 OK

```

1  {
2    "success": true,
3    "data": {
4      "keycloakClientId": "phoenix",
5      "clientName": "피닉스 백오피스",
6      "menus": [
7        {
8          "id": 1,
9          "parentId": null,
10         "name": "회원 관리",
11         "type": "GROUP",
12         "url": null,
13         "displayOrder": 1,
14         "displayYn": true,
15         "privacyIncludeYn": true,
16         "locationIncludeYn": false,
17         "children": [
18           {
19             "id": 2,
20             "parentId": 1,
21             "name": "일반 회원 관리",
22             "type": "ITEM",
23             "url": "/members/general",
24             "displayOrder": 1,
25             "displayYn": true,
26             "privacyIncludeYn": true,
27             "locationIncludeYn": false,
28             "scopes": ["GET", "POST", "PUT", "DELETE"]
29           }
30         ]
31       }
32     ]
33   }
34 }
```

지원되는 Mock 클라이언트

keycloakClientId	clientName	설명
phoenix	피닉스 백오피스	회원 관리, 차량 관리 메뉴 포함
portal	포털 백오피스	대시보드, 권한 관리, 시스템 설정 메뉴 포함

test	테스트 백오피스	간단한 테스트용 메뉴 포함
backoffice	통합 백오피스	운영 관리, 정산 관리, 고객 지원 메뉴 포함

8.2 Mock 권한별 메뉴 조회

GET /mock/v2/menus/authorized

Mock 권한별 메뉴를 반환합니다. 실제 API와 동일한 형식으로 응답하며, 모든 메뉴에 scopes가 포함됩니다.

Query Parameters

Parameter	Type	Required	Description
keycloakClientId	string	Yes	클라이언트 ID (phoenix , portal , test , backoffice 지 원)

Response 200 OK

8.1과 동일한 형식으로 응답합니다.