



ACCESS CONTROL INTEGRATION DOCUMENT

(DECEMBER 2025)

GENERAL

This document is to identify key APIs used for access control integration. The below captures the following scenarios

- Get Resident: The purpose is to check if the user already exists or not before creating the resident
- Create Resident: To create the resident (owner, family member, tenant)
- Get Identify: To generate the dynamic QR code for the resident
- Get Visitor QR: To generate a QR code for a visitor
- Delete Resident: To delete the Owner and family members in case of sale or other. It should be noted that tenants should be automatically deleted based on their end date.

NOTES

- Lyve keeps the relationship between resident and units
- Lyve controls quota for invitations
- Lyve does not track actual scans but the generation of the badges
- All units must be created with a standard naming convention aligned with the client

GET RESIDENT

Retrieves a resident's details using their email and community.

PARAMETERS

| Name | Type | Description |
|-----------|--------|------------------------|
| email | string | Resident email address |
| community | string | Community identifier |

RESPONSE

| Name | Type | Mandatory | Description |
|-----------|--------|-----------|-------------------------|
| ownerId | string | YES | Unique owner identifier |
| phone | string | NO | Resident phone number |
| name | string | NO | Resident full name |
| email | string | NO | Resident email address |
| community | string | NO | Community identifier |
| type | string | NO | Resident type |
| from | string | NO | Start date of validity |
| to | string | NO | End date of validity |
| unitId | string | YES | Linked unit identifier |

EXAMPLE RESPONSE

```
{  
  "ownerId": "12345",  
  "phone": "+201234567890",  
  "name": "John Doe",  
  "email": "john@example.com",  
  "community": "GreenHills",  
  "type": "resident",  
  "from": "2025-01-01",  
  "to": "2025-12-31",  
  "unitId": "U1001"  
}
```

CREATE RESIDENT

Creates a new Resident record in the specified community.

PARAMETERS

| Name | Type | Description |
|-----------|--------|------------------------|
| name | string | Resident full name |
| phone | string | Resident phone number |
| email | string | Resident email address |
| community | string | Community identifier |
| from | string | Start date of validity |
| to | string | End date of validity |
| ownerType | string | Owner type |
| unitId | string | Unit ID |

RESPONSE

| Name | Type | Mandatory | Description |
|-----------|--------|-----------|-------------------------|
| ownerId | string | YES | Unique owner identifier |
| phone | string | NO | Resident phone number |
| name | string | NO | Resident full name |
| email | string | NO | Resident email address |
| community | string | NO | Community identifier |
| type | string | NO | Resident type |
| from | string | NO | Start date of validity |
| to | string | NO | End date of validity |
| unitId | string | NO | Linked unit identifier |

EXAMPLE RESPONSE

```
{  
  "ownerId": "67890",  
  "phone": "+201011223344",  
  "name": "Jane Smith",  
  "email": "jane@example.com",  
  "community": "PalmHeights",  
  "type": "tenant",  
  "from": "2025-02-01",  
  "to": "2025-12-31",  
  "unitId": "U2002"  
}
```

GET IDENTITY

Retrieves identity details for a given unit and owner. This is the API that calls the resident's dynamic QR code

PARAMETERS

| Name | Type | Description |
|---------|--------|------------------|
| unitId | string | Unit identifier |
| ownerId | string | Owner identifier |

RESPONSE

| Name | Type | Mandatory | Description |
|-----------|--------|-----------|---|
| id | string | NO | Unique identity identifier |
| ownerId | string | NO | Owner identifier |
| ownerType | string | NO | Type of owner (e.g., resident, tenant) |
| unitId | string | NO | Unit identifier |
| qrCode | string | YES | Dynamic QR code for access (String and NOT IMAGE AND NOT BASE64) |

EXAMPLE RESPONSE

```
{  
  "id": "ID1001",  
  "ownerId": "12345",  
  "ownerType": "resident",  
  "unitId": "U1001",  
  "qrCode": "QR1234567890"  
}
```

GET VISITOR QR

Generates a visitor QR code for a scheduled visit.

PARAMETERS

| Name | Type | Description |
|-------------|--------|-------------------|
| unitId | string | Unit identifier |
| ownerId | string | Owner identifier |
| visitorName | string | Visitor full name |
| visitDate | string | Date of visit |

RESPONSE

| Name | Type | Mandatory | Description |
|-------------|--------|-----------|--|
| visitId | string | NO | Unique visit identifier |
| unitId | string | NO | Unit identifier |
| ownerId | string | NO | Owner identifier |
| ownerType | string | NO | Owner type |
| visitorName | string | NO | Visitor name |
| visitDate | string | NO | Visit date |
| qrCode | string | YES | Generated visitor QR code (String and NOT IMAGE AND NOT BASE64) |

EXAMPLE RESPONSE

```
{  
  "visitId": "V1001",  
  "unitId": "U1001",  
  "ownerId": "12345",  
  "ownerType": "resident",  
  "visitorName": "Mark Lee",  
  "visitDate": "2025-03-15",  
  "qrCode": "QRV123456789"  
}
```

DELETE RESIDENT

Removes a Resident record based on owner ID.

PARAMETERS

| Name | Type | Description |
|---------|--------|------------------|
| ownerId | string | Owner identifier |
| unitID | String | Unit ID |

RESPONSE

| Name | Type | Description |
|---------|---------|--------------------------------------|
| success | boolean | Indicates if the operation succeeded |

EXAMPLE RESPONSE

```
{ "success": true }
```