# AL-Raed Reward Program - QA Test Guide

**Version:** 1.0
**Last Updated:** 2026-02-24
**API Status:** Ready for QA Testing
**Build Status:** 0 errors, 0 warnings

## Table of Contents

## 1. System Overview

### What Is This System?

AL-Raed Reward Program is an onboarding platform for AL-Raed (a Saudi FMCG/retail distribution company). It digitizes the process of registering retail partners (shop owners, sellers, technicians) into a reward program, with a two-tier approval workflow.

### The 6 Roles

| Role | Who They Are | How They Get In |
|---|---|---|
| **SystemAdmin** | IT/operations admin | Pre-seeded in database |
| **ZoneManager** | Regional manager overseeing a geographic zone | Pre-seeded in database |
| **SalesMan** | AL-Raed distribution rep assigned to cities | Pre-seeded in database |
| **ShopOwner** | Retail store owner (has VAT, CRN, physical shop) | Self-registers via API |
| **Seller** | Employee working inside a shop | Self-registers with ShopCode |
| **Technician** | Field service technician | Self-registers via API |

### Geographic Hierarchy

```
Region (top level, managed by 1 ZoneManager)
  └── City (has 1 ApprovalSalesMan)
        └── District (optional)
```

### Two-Tier Approval Workflow

```
User Registers
    │
    ▼
Status: PendingSalesman ──► SalesMan Reviews
    │                           │
    │                     ┌─────┴─────┐
    │                  Approve      Reject
    │                     │           │
    │                     ▼           ▼
    │            PendingZoneManager  Rejected (final)
    │                     │
    │               ZoneManager Reviews
    │                     │
    │               ┌─────┴─────┐
    │            Approve      Reject
    │               │           │
    │               ▼           ▼
    │           Approved      Rejected (final)
    │           (can login)   (cannot login)
    │
    └── ShopOwner gets 6-char ShopCode on final approval
```

### Authentication Method

- **No passwords** — purely OTP-based (WhatsApp/SMS via Twilio)
- OTP sent on registration and login
- JWT access tokens (60 min) + refresh tokens (7 days)

## 2. Environment Setup

### 2.1 URLs

| Environment | Base URL | Swagger |
|---|---|---|
| Development (local) | `http://localhost:5000` | `/swagger` |
| Staging | `http://staging.raedrewardapp.com` | `/swagger` |
| Production | `http://raedrewards-001-site1.atempurl.com` | Disabled |

### 2.2 Twilio Mock Mode

In **Development** and **Staging**, Twilio runs in mock mode:

- OTPs are **not actually sent** via WhatsApp
- The mock OTP code is logged and stored in the database
- Check the `OtpCodes` table in the database for the actual PinId and verification status
- Check application logs (Serilog) for OTP details

> **Important:** To get the OTP in mock mode, you need database access or log access.

## 2.3 First-Time Setup

On first run in Development:

1. Database auto-migrates (tables created automatically)
2. DataSeeder runs (creates roles, 31 users, 8 regions, 140 cities)
3. Swagger available at `/swagger`

## 2.4 Authentication in Swagger

1. Open Swagger UI
2. Click the **Authorize** button (lock icon, top right)
3. Enter: `Bearer {your-jwt-token}` (include the word "Bearer")
4. Click Authorize
5. All subsequent requests will include the JWT header

---

# 3. Seeded Test Data

## 3.1 Roles (6)

| Role | Purpose |
|---|---|
| SystemAdmin | Full system access |
| ZoneManager | Regional approval (Stage 2) |
| SalesMan | City-level approval (Stage 1) |
| ShopOwner | Registered shop owners |
| Seller | Shop employees |
| Technician | Field technicians |

## 3.2 Pre-Seeded Users (31)

### SystemAdmin (1)

| Name | Mobile | Role | Status |
|---|---|---|---|
| مدير النظام | 0500000001 | SystemAdmin | Approved |

### Pure ZoneManagers (5)

| Name | Mobile | Manages Region |
|---|---|---|
| فرحان ممدوح | 0500000002 | الرياض (Riyadh) |
| الطيب حسين | 0500000003 | المدينة المنورة (Madinah) |
| محمد العجوز | 0500000004 | جازان (Jazan) |
| نيازي عمر | 0500000005 | المنطقة الجنوبية (Southern) |
| محمد اسماعيل | 0500000006 | تبوك و الشمال (Tabuk & Northern) |

### Dual-Role Users — ZoneManager + SalesMan (3)

| Name | Mobile | ZM Region | SM Cities |
|---|---|---|---|
| نعيم عوض | 0500000007 | المنطقة الغربية (Western) | Various Western cities |
| سيد بخيت | 0500000008 | الشرقية (Eastern) | Various Eastern cities |
| وليد السكري | 0500000009 | القصيم (Qassim) | Various Qassim cities |

### Pure SalesMen (22)

| Name | Mobile |
|---|---|
| محمود حجازي | 0500000010 |
| احمد سمير | 0500000011 |
| احمد جمال | 0500000012 |
| ابراهيم الشعراوي | 0500000013 |
| محمد المشير | 0500000014 |
| محمد اياد | 0500000015 |
| يوسف جابر | 0500000016 |
| احمد الجن | 0500000017 |
| احمد القليوبي | 0500000018 |
| اشرف وائل | 0500000019 |
| محمد فؤاد | 0500000020 |
| اسامة عبد العليم | 0500000021 |
| محمد مدبولي | 0500000022 |
| نبيل صلاح | 0500000023 |
| محمد مراد | 0500000024 |
| معتز مكرم | 0500000025 |
| هيثم محمد | 0500000026 |
| محمد الصاوي | 0500000027 |
| عمرو عادل | 0500000028 |
| ايهاب صلاح | 0500000029 |
| حسام حسن | 0500000030 |
| محمد ناصر | 0500000031 |

> **To log in as any seeded user:** Use `POST /api/auth/login` with their mobile number. They are all pre-approved.

## 3.3 Regions (8)

| Arabic Name | English Name | ZoneManager |
|---|---|---|
| الرياض | Riyadh | فرحان ممدوح |
| المنطقة الغربية | Western Region | نعيم عوض |
| المدينة المنورة | Madinah | الطيب حسين |
| الشرقية | Eastern Region | سيد بخيت |
| جازان | Jazan | محمد العجوز |
| المنطقة الجنوبية | Southern Region | نيازي عمر |
| تبوك و الشمال | Tabuk & Northern | محمد اسماعيل |
| القصيم | Qassim | وليد السكري |

### 3.4 Cities (140 total)

Use `GET /api/lookup/regions` then `GET /api/lookup/regions/{regionId}/cities` to get actual IDs.

**Sample (Riyadh region — 12 cities):**

| City | SalesMan |
|---|---|
| الرياض | محمود حجازي |
| الخرج | احمد سمير |
| الأفلاج | احمد سمير |
| القويعية | احمد جمال |
| ... | ... |

## 4. API Endpoints Reference

### 4.1 Lookup Endpoints (Public — No Auth Required)

#### GET /api/lookup/regions

Returns all regions. Cached for 1 hour.

**Response (200):**

```
[
  {
    "id": "guid-string",
    "nameAr": "الرياض",
    "nameEn": "Riyadh"
  }
]
```

### GET /api/lookup/regions/{regionId}/cities

Returns cities in a region. Cached for 1 hour.

**Response (200):**

```
[
  {
    "id": "guid-string",
    "nameAr": "الرياض",
    "nameEn": "Riyadh",
    "regionId": "region-guid"
  }
]
```

**Error:** 404 if regionId not found.

### GET /api/lookup/cities/{cityId}/districts

Returns districts in a city. Cached for 1 hour.

**Response (200):**

```
[
  {
    "id": "guid-string",
    "nameAr": "حي الصفا",
    "nameEn": "As Safa",
    "cityId": "city-guid"
  }
]
```

**Error:** 404 if cityId not found.

---

## 4.2 Registration Endpoints (Public — No Auth Required)

### POST /api/auth/register/shop-owner

**Content-Type:** `multipart/form-data`

| Field | Type | Required | Rules |
|---|---|---|---|
| StoreName | string | Yes | 2-150 characters |
| OwnerName | string | Yes | 2-100 characters |
| MobileNumber | string | Yes | Format: `05XXXXXXXX` (10 digits) |
| VAT | string | Yes | Exactly 15 digits, starts and ends with 3 |
| CRN | string | Yes | Exactly 10 digits |
| RegionId | string | Yes | Must exist |
| CityId | string | Yes | Must belong to region, must have SalesMan |
| DistrictId | string | No | If provided, must belong to city |
| ShopImage | file | Yes | JPG or PNG only, max 5 MB |
| NationalAddress.BuildingNumber | int | Yes | Greater than 0 |
| NationalAddress.Street | string | Yes | 1-100 characters |
| NationalAddress.PostalCode | string | Yes | Exactly 5 digits |
| NationalAddress.SubNumber | int | Yes | Greater than 0 |

**Success Response (200):**

```
{
  "pinId": "twilio-pin-id-string",
  "maskedMobileNumber": "0500****56"
}
```

**Possible Errors:**

- 400: Validation failed, city not found, no salesman for city, invalid district
- 409: Mobile/VAT/CRN already registered

## POST /api/auth/register/seller

**Content-Type:** `application/json`

```
{
  "name": "اسم البائع",
  "mobileNumber": "0512345678",
  "shopCode": "ABC123"
}
```

| Field | Type | Required | Rules |
|---|---|---|---|
| Name | string | Yes | 2-100 characters |
| MobileNumber | string | Yes | Format: `05XXXXXXXX` |
| ShopCode | string | Yes | Exactly 6 chars, uppercase alphanumeric `^[A-Z0-9]{6}$` |

**Success Response (200):**

```
{
  "pinId": "twilio-pin-id-string",
  "maskedMobileNumber": "0512****78"
}
```

**Possible Errors:**

- 400: Invalid shop code, shop owner not approved
- 404: Shop owner not found
- 409: Mobile already registered

---

### POST /api/auth/register/technician

**Content-Type:** `application/json`

```
{
  "name": "اسم الفني",
  "mobileNumber": "0512345679",
  "regionId": "region-guid",
  "cityId": "city-guid",
  "districtId": "district-guid-or-null",
  "postalCode": "12345"
}
```

| Field | Type | Required | Rules |
|---|---|---|---|
| Name | string | Yes | 2-100 characters |
| MobileNumber | string | Yes | Format: `05XXXXXXXX` |
| RegionId | string | Yes | Must exist |
| CityId | string | Yes | Must belong to region, must have SalesMan |
| DistrictId | string | No | If provided, must belong to city |
| PostalCode | string | Yes | Exactly 5 digits `^\d{5}$` |

**Success Response (200):**

```
{
  "pinId": "twilio-pin-id-string",
  "maskedMobileNumber": "0512****79"
}
```

**Possible Errors:**

- 400: City not found, no salesman, invalid district
- 409: Mobile already registered

## POST /api/auth/register/verify

Verifies OTP for **all 3 registration types** (shared endpoint).

**Content-Type:** `application/json`

```
{
  "pinId": "the-pin-id-from-registration",
  "otp": "123456"
}
```

| Field | Type | Required | Rules |
|---|---|---|---|
| PinId | string | Yes | Not empty |
| Otp | string | Yes | Exactly 6 digits |

**Success Response (200):**

```
{
  "id": "new-user-id",
  "name": "user name",
  "mobileNumber": "05XXXXXXXX",
  "userType": 1,
  "registrationStatus": 1
}
```

**UserType values:** 1=ShopOwner, 2=Seller, 3=Technician, 4=SalesMan, 5=ZoneManager, 6=SystemAdmin
**RegistrationStatus values:** 1=PendingSalesman, 2=PendingZoneManager, 3=Approved, 4=Rejected

**Possible Errors:**

- 400: Invalid/expired/used OTP, max attempts exceeded (5), registration data not found

## POST /api/auth/resend-otp

Resends OTP for a pending registration or login.

**Content-Type:** `application/json`

```
{
  "mobileNumber": "0512345678"
}
```

**Success Response (200):**

```
{
  "pinId": "new-pin-id",
  "maskedMobileNumber": "0512****78"
}
```

**Rate Limits:**

- 30-second cooldown between resends
- Max 3 OTP requests per mobile within 15 minutes

**Possible Errors:**

- 400: No OTP found for mobile
- 429: Too soon (< 30 seconds) or too many requests (> 3 per 15 min)

---

## 4.3 Login Endpoints (Public — No Auth Required)

### POST /api/auth/login

Sends OTP for login.

**Content-Type:** `application/json`

```
{
  "mobileNumber": "0500000010"
}
```

**Success Response (200):**

```
{
  "pinId": "twilio-pin-id",
  "maskedMobileNumber": "0500****10"
}
```

**Possible Errors:**

- 404: User not found
- 403: User rejected, user not approved (pending), user disabled

---

### POST /api/auth/login/verify

Verifies login OTP and returns JWT tokens.

**Content-Type:** `application/json`

```
{
  "pinId": "pin-id-from-login",
  "otp": "123456"
}
```

**Success Response (200):**

```
{
  "token": "eyJhbGciOiJIUzI1NiIs...",
  "refreshToken": "random-64-byte-string",
  "expiresIn": 3600,
  "refreshTokenExpiration": "2026-03-03T10:30:00Z",
  "user": {
    "id": "user-id",
    "name": "محمود حجازي",
    "mobileNumber": "0500000010",
    "userType": 4,
    "registrationStatus": 3
  }
}
```

**Possible Errors:**

- 400: Invalid/expired OTP
- 403: User not approved

---

## 4.4 Token Management Endpoints (Auth Required)

### POST /api/auth/refresh-token

**Headers:** `Authorization: Bearer {access-token}`
**Content-Type:** `application/json`

```
{
  "refreshToken": "the-refresh-token-string"
}
```

**Success Response (200):** Same format as login/verify response (new tokens).

**Behavior:**

- Old refresh token is automatically revoked
- Expired/revoked tokens are cleaned up
- Returns new access + refresh token pair

**Possible Errors:**

- 401: Invalid/expired/revoked refresh token

- 403: User disabled

---

### POST /api/auth/revoke-token

Logout — revokes the refresh token.

**Headers:** `Authorization: Bearer {access-token}`
**Content-Type:** `application/json`

```
{
  "refreshToken": "the-refresh-token-string"
}
```

**Success Response (200):**

```
{
  "message": "تم تسجيل الخروج بنجاح"
}
```

**Possible Errors:**

- 401: Invalid/inactive refresh token

---

## 4.5 Approval Endpoints (Auth Required — SalesMan or ZoneManager Only)

### GET /api/approvals/pending?page=1&pageSize=20

Returns paginated list of users pending approval.

**Headers:** `Authorization: Bearer {salesman-or-zm-token}`

**Query Parameters:**

| Param | Default | Range |
|---|---|---|
| page | 1 | Min: 1 |
| pageSize | 20 | 1-50 (clamped) |

**What Each Role Sees:**

| Approver Role | Sees Users With Status | Filter |
|---|---|---|
| SalesMan only | PendingSalesman | AssignedSalesManId = current user |
| ZoneManager only | PendingZoneManager | User's city's region's ZoneManagerId = current user |
| Dual-role (SM+ZM) | Both PendingSalesman AND PendingZoneManager | Combined queue |

**Success Response (200):**

```
{
  "items": [
    {
      "id": "user-id",
      "name": "محمد أحمد",
      "mobileNumber": "0512345678",
      "userType": 1,
      "registrationStatus": 1,
      "registeredAt": "2026-02-24T10:30:00Z",
      "storeName": "متجر الجودة",
      "vat": "300000000000003",
      "crn": "1234567890",
      "shopImageUrl": "/uploads/shops/image.jpg",
      "shopCode": null,
      "regionName": "الرياض",
      "cityName": "الرياض",
      "districtName": null,
      "street": "شارع الملك فهد",
      "buildingNumber": 1234,
      "postalCode": "12345",
      "subNumber": 1,
      "shopOwnerName": null,
      "assignedSalesManName": "محمود حجازي"
    }
  ],
  "totalCount": 5,
  "page": 1,
  "pageSize": 20,
  "totalPages": 1,
  "hasNextPage": false,
  "hasPreviousPage": false
}
```

**Possible Errors:**

- 401: Not authenticated
- 403: User does not have SalesMan or ZoneManager role

---

## POST /api/approvals/approve

**Headers:** `Authorization: Bearer {salesman-or-zm-token}`
**Content-Type:** `application/json`

```
{
  "userId": "the-user-id-to-approve"
}
```

**Approval Rules:**

| Current Status | Approver Must Be | Result |
|---|---|---|
| PendingSalesman | The assigned SalesMan | Status changes to PendingZoneManager |
| PendingZoneManager | The region's ZoneManager | Status changes to Approved |

**On Final Approval (ZoneManager):**

- ShopOwner receives a generated 6-character ShopCode (e.g., "A3X9K2")
- Welcome WhatsApp message sent (fire-and-forget)

**Success Response (200):**

```
{
  "message": "تمت الموافقة بنجاح"
}
```

**Possible Errors:**

- 400: User not found, user not pending, region has no zone manager
- 403: Not authorized to approve (wrong salesman/zone manager)

## POST /api/approvals/reject

**Headers:** `Authorization: Bearer {salesman-or-zm-token}`
**Content-Type:** `application/json`

```
{
  "userId": "the-user-id-to-reject",
  "reason": "سبب الرفض - المستندات غير مكتملة"
}
```

| Field | Required | Rules |
|---|---|---|
| UserId | Yes | Must exist, must be pending |
| Reason | Yes | 1-500 characters |

**Success Response (200):**

```
{
  "message": "تم الرفض بنجاح"
}
```

**Possible Errors:**

- 400: User not found, not pending, reason too long
- 403: Not authorized to reject

# 5. Test Flows (Step-by-Step)

## Flow 1: Complete ShopOwner Registration + Approval + Login

This is the **primary end-to-end flow**. Follow each step in order.

**Step 1 — Get location IDs**

```
GET /api/lookup/regions
```

Pick a region ID (e.g., Riyadh). Note it.

```
GET /api/lookup/regions/{regionId}/cities
```

Pick a city ID. Note it.

**Step 2 — Register ShopOwner**

```
POST /api/auth/register/shop-owner
Content-Type: multipart/form-data

StoreName: متجر اختبار
OwnerName: صاحب المتجر
MobileNumber: 0555111222
VAT: 300000000100003
CRN: 1234500001
RegionId: {regionId from step 1}
CityId: {cityId from step 1}
ShopImage: {upload a .jpg or .png file, under 5MB}
NationalAddress.BuildingNumber: 100
NationalAddress.Street: شارع التحلية
NationalAddress.PostalCode: 12345
NationalAddress.SubNumber: 1
```

**Expected:** 200 with `pinId` and `maskedMobileNumber`.
**Save:** the `pinId` value.

**Step 3 — Verify OTP**

```
POST /api/auth/register/verify
{
  "pinId": "{pinId from step 2}",
  "otp": "{get from database or logs in mock mode}"
}
```

**Expected:** 200 with user details, `registrationStatus: 1` (PendingSalesman).
**Save:** the `id` (userId).

**Step 4 — Login as the SalesMan who covers that city**

Check which SalesMan is assigned to the city you chose (see seeded data in Section 3). For example, if you chose الرياض city, the SalesMan is محمود حجازي (mobile 0500000010).

```
POST /api/auth/login
{ "mobileNumber": "0500000010" }
```

Get pinId, then:

```
POST /api/auth/login/verify
{ "pinId": "{pinId}", "otp": "{from DB/logs}" }
```

**Save:** the `token` (JWT).

**Step 5 — Check pending queue as SalesMan**

```
GET /api/approvals/pending?page=1&pageSize=20
Authorization: Bearer {salesman-token}
```

**Expected:** The ShopOwner from Step 2 should appear in the list.

**Step 6 — Approve as SalesMan**

```
POST /api/approvals/approve
Authorization: Bearer {salesman-token}
{ "userId": "{userId from step 3}" }
```

**Expected:** 200, user moves to PendingZoneManager.

**Step 7 — Login as the ZoneManager for that region**

For Riyadh region, ZoneManager is فرحان ممدوح (mobile 0500000002).

```
POST /api/auth/login
{ "mobileNumber": "0500000002" }
```

Verify OTP, get token.

**Step 8 — Approve as ZoneManager**

```
POST /api/approvals/approve
Authorization: Bearer {zm-token}
{ "userId": "{userId from step 3}" }
```

**Expected:** 200, user is now Approved. ShopCode generated.

**Step 9 — Login as the newly approved ShopOwner**

```
POST /api/auth/login
{ "mobileNumber": "0555111222" }
```

Verify OTP.

**Expected:** 200 with JWT tokens, `registrationStatus: 3` (Approved).

---

## Flow 2: Seller Registration (Requires Approved ShopOwner)

**Prerequisite:** Complete Flow 1 first. Note the ShopCode from the approval response or database.

**Step 1 — Register Seller**

```
POST /api/auth/register/seller
{
  "name": "بائع اختبار",
  "mobileNumber": "0555111333",
  "shopCode": "{ShopCode from Flow 1}"
}
```

**Step 2 — Verify OTP**

```
POST /api/auth/register/verify
{ "pinId": "{pinId}", "otp": "{from DB/logs}" }
```

**Step 3 — Approve via SalesMan then ZoneManager** (same as Flow 1, Steps 4-8)

---

## Flow 3: Technician Registration

**Step 1 — Get location IDs** (same as Flow 1, Step 1)

**Step 2 — Register Technician**

```
POST /api/auth/register/technician
{
  "name": "فني اختبار",
  "mobileNumber": "0555111444",
  "regionId": "{regionId}",
  "cityId": "{cityId}",
  "districtId": null,
  "postalCode": "54321"
}
```

**Step 3 — Verify and Approve** (same pattern)

---

## Flow 4: Rejection Flow

Follow Flow 1 Steps 1-4 (register + login as SalesMan), then:

```
POST /api/approvals/reject
Authorization: Bearer {salesman-token}
{
  "userId": "{userId}",
  "reason": "المستندات غير مكتملة"
}
```

Then verify the rejected user **cannot login**:

```
POST /api/auth/login
{ "mobileNumber": "{rejected-user-mobile}" }
```

**Expected:** 403 Forbidden.

---

## Flow 5: Token Refresh and Revoke

**Step 1 — Login and get tokens** (any approved user)

**Step 2 — Refresh token**

```
POST /api/auth/refresh-token
Authorization: Bearer {access-token}
{ "refreshToken": "{refresh-token-from-login}" }
```

**Expected:** New access token + new refresh token.

**Step 3 — Try old refresh token**

```
POST /api/auth/refresh-token
Authorization: Bearer {new-access-token}
{ "refreshToken": "{OLD-refresh-token}" }
```

**Expected:** 401 (old token was revoked).

**Step 4 — Revoke (logout)**

```
POST /api/auth/revoke-token
Authorization: Bearer {access-token}
{ "refreshToken": "{current-refresh-token}" }
```

**Expected:** 200, token revoked.

---

## Flow 6: Dual-Role Approval (SalesMan + ZoneManager)

Login as a dual-role user (e.g., نعيم عوض, mobile 0500000007 — SalesMan + ZoneManager for Western Region).

```
GET /api/approvals/pending?page=1&pageSize=20
Authorization: Bearer {dual-role-token}
```

**Expected:** Combined queue showing BOTH PendingSalesman users (assigned to them as SM) AND PendingZoneManager users (in their managed region).

---

# 6. Validation Rules

## 6.1 Mobile Number

- Format: `05XXXXXXXX` (exactly 10 digits starting with 05)
- Regex: `^05\d{8}$`
- Must be unique per user

## 6.2 VAT Number

- Exactly 15 digits
- Must start with 3 and end with 3
- Regex: `^3\d{13}3$`
- Must be unique

## 6.3 CRN (Commercial Registration Number)

- Exactly 10 digits
- Regex: `^\d{10}$`
- Must be unique

## 6.4 Shop Code

- Exactly 6 characters
- Uppercase alphanumeric only
- Regex: `^[A-Z0-9]{6}$`

## 6.5 Postal Code

- Exactly 5 digits
- Regex: `^\d{5}$`

## 6.6 OTP

- Exactly 6 digits
- Expires after 5 minutes
- Max 5 verification attempts
- Rate limit: max 3 OTP requests per mobile in 15 minutes
- Resend cooldown: 30 seconds

## 6.7 Shop Image

- Allowed formats: JPG (.jpg, .jpeg) and PNG (.png) only
- Max file size: 5 MB

## 6.8 Name Fields

- Minimum: 2 characters
- Maximum: 100 characters (OwnerName, Seller Name, Technician Name)
- StoreName maximum: 150 characters

## 6.9 National Address

- BuildingNumber: integer, greater than 0
- Street: 1-100 characters
- PostalCode: exactly 5 digits
- SubNumber: integer, greater than 0

## 6.10 Rejection Reason

- Required for rejection
- 1-500 characters

---

# 7. Error Codes Reference

## Error Response Format

All errors follow the ProblemDetails format:

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "extensions": {
    "error": [
      {
        "code": "Auth.MobileAlreadyRegistered",
        "description": "رقم الجوال مسجل مسبقاً"
      }
    ]
  }
}
```

## Complete Error Code Table

### Authentication Errors

| Code | HTTP Status | Arabic Description |
|---|---|---|
| Auth.MobileAlreadyRegistered | 409 | رقم الجوال مسجل مسبقاً |
| Auth.VatAlreadyExists | 409 | رقم الضريبة مسجل مسبقاً |
| Auth.CrnAlreadyExists | 409 | السجل التجاري مسجل مسبقاً |
| Auth.UserNotFound | 404 | المستخدم غير موجود |
| Auth.UserNotApproved | 403 | المستخدم غير معتمد |
| Auth.UserRejected | 403 | تم رفض طلب التسجيل |
| Auth.UserDisabled | 403 | تم تعطيل حساب المستخدم |
| Auth.CityNotFound | 400 | المدينة غير موجودة |
| Auth.DistrictNotFound | 400 | الحي غير موجود |
| Auth.InvalidShopCode | 400 | رمز المتجر غير صالح |
| Auth.ShopOwnerNotApproved | 400 | صاحب المتجر غير معتمد |

## OTP Errors

| Code | HTTP Status | Arabic Description |
|---|---|---|
| Auth.OtpNotFound | 400 | رمز التحقق غير موجود |
| Auth.OtpExpired | 400 | انتهت صلاحية رمز التحقق |
| Auth.OtpInvalid | 400 | رمز التحقق غير صحيح |
| Auth.OtpAlreadyUsed | 400 | رمز التحقق مستخدم مسبقاً |
| Auth.TooManyOtpRequests | 429 | تم تجاوز الحد الأقصى لطلبات رمز التحقق |
| Auth.OtpResendTooSoon | 429 | يرجى الانتظار قبل إعادة إرسال رمز التحقق |
| Auth.MaxVerificationAttempts | 400 | تم تجاوز الحد الأقصى لمحاولات التحقق |

## Token Errors

| Code | HTTP Status | Arabic Description |
|---|---|---|
| Auth.InvalidRefreshToken | 401 | رمز التحديث غير صالح |
| Auth.RefreshTokenExpired | 401 | انتهت صلاحية رمز التحديث |
| Auth.RefreshTokenRevoked | 401 | تم إبطال رمز التحديث |

## Approval Errors

| Code | HTTP Status | Arabic Description |
|---|---|---|
| Approval.UserNotPendingApproval | 400 | المستخدم ليس في حالة انتظار الموافقة |
| Approval.NotAuthorizedToApprove | 403 | غير مصرح لك بالموافقة |
| Approval.NoZoneManagerForRegion | 400 | لا يوجد مدير منطقة للمنطقة |

### Lookup Errors

| Code | HTTP Status | Arabic Description |
|---|---|---|
| Lookup.RegionNotFound | 404 | المنطقة غير موجودة |
| Lookup.CityNotFound | 404 | المدينة غير موجودة |
| Lookup.DistrictNotFound | 404 | الحي غير موجود |

## 8. Role-Based Access Matrix

| Endpoint | No Auth | SystemAdmin | ZoneManager | SalesMan | ShopOwner | Seller | Technician |
|---|---|---|---|---|---|---|---|
| GET /api/lookup/* | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| POST /api/auth/register/* | Yes | - | - | - | - | - | - |
| POST /api/auth/register/verify | Yes | - | - | - | - | - | - |
| POST /api/auth/login | Yes | - | - | - | - | - | - |
| POST /api/auth/login/verify | Yes | - | - | - | - | - | - |
| POST /api/auth/resend-otp | Yes | - | - | - | - | - | - |
| POST /api/auth/refresh-token | - | Yes | Yes | Yes | Yes | Yes | Yes |
| POST /api/auth/revoke-token | - | Yes | Yes | Yes | Yes | Yes | Yes |
| GET /api/approvals/pending | - | - | Yes | Yes | No (403) | No (403) | No (403) |
| POST /api/approvals/approve | - | - | Yes | Yes | No (403) | No (403) | No (403) |
| POST /api/approvals/reject | - | - | Yes | Yes | No (403) | No (403) | No (403) |

# 9. Test Checklist

## 9.1 Registration Tests

### ShopOwner Registration

- □ Valid registration with all required fields
- □ Missing StoreName → 400
- □ Missing OwnerName → 400
- □ Invalid mobile format (e.g., "0612345678") → 400
- □ Duplicate mobile number → 409
- □ Duplicate VAT → 409
- □ Duplicate CRN → 409
- □ Invalid VAT format (not starting/ending with 3) → 400
- □ Invalid CRN format (not 10 digits) → 400
- □ Non-existent RegionId → 400
- □ Non-existent CityId → 400

- □ CityId not belonging to RegionId → 400
- □ City without assigned SalesMan → 400
- □ Invalid DistrictId (not in city) → 400
- □ Non-JPG/PNG image → 400
- □ Image > 5 MB → 400
- □ Missing image → 400
- □ BuildingNumber = 0 or negative → 400
- □ PostalCode not 5 digits → 400
- □ Empty Street → 400
- □ SubNumber = 0 or negative → 400
- □ Registration with DistrictId = null (valid) → 200

### Seller Registration

- □ Valid registration with approved ShopOwner code → 200
- □ Non-existent ShopCode → 400
- □ ShopCode from non-approved ShopOwner → 400
- □ Invalid ShopCode format (lowercase, wrong length) → 400
- □ Duplicate mobile → 409

### Technician Registration

- □ Valid registration → 200
- □ Non-existent CityId → 400
- □ City without SalesMan → 400
- □ Invalid PostalCode (not 5 digits) → 400
- □ DistrictId = null (valid) → 200

## 9.2 OTP Tests

- □ Valid OTP verification → 200
- □ Expired OTP (> 5 minutes) → 400
- □ Wrong OTP code → 400
- □ Already used OTP → 400
- □ 6th attempt (max 5) → 400
- □ Empty PinId → 400
- □ Resend OTP within 30 seconds → 429
- □ Resend OTP after 30 seconds → 200
- □ 4th OTP request in 15 minutes → 429

## 9.3 Login Tests

- □ Login with approved user → 200
- □ Login with non-existent mobile → 404
- □ Login with PendingSalesman user → 403
- □ Login with PendingZoneManager user → 403
- □ Login with Rejected user → 403
- □ Login with disabled user → 403

- □ Verify login OTP → 200 with JWT tokens

## 9.4 Token Tests

- □ Refresh with valid token → 200 (new tokens)
- □ Refresh with revoked token → 401
- □ Refresh with expired token → 401
- □ Refresh with invalid string → 401
- □ Old refresh token rejected after refresh → 401
- □ Revoke token → 200
- □ Revoke already-revoked token → 401

## 9.5 Approval Tests

- □ SalesMan sees only their assigned PendingSalesman users
- □ ZoneManager sees only their region's PendingZoneManager users
- □ Dual-role user sees combined queue
- □ SalesMan approves → status moves to PendingZoneManager
- □ Wrong SalesMan cannot approve → 403
- □ ZoneManager approves → status moves to Approved
- □ Wrong ZoneManager cannot approve → 403
- □ ShopOwner gets ShopCode on final approval
- □ Seller/Technician do NOT get ShopCode
- □ Rejection with reason → 200
- □ Rejection without reason → 400
- □ Rejected user cannot login → 403
- □ Pagination: page=1, pageSize=20 → correct
- □ Pagination: pageSize=100 → clamped to 50
- □ Non-SM/ZM user accessing approvals → 403
- □ Unauthenticated user → 401

## 9.6 Lookup Tests

- □ Get all regions → 8 regions returned
- □ Get cities by region → correct cities with regionId
- □ Get cities for invalid region → 404
- □ Get districts by city → districts or empty array
- □ Get districts for invalid city → 404

## 9.7 Edge Cases

- □ Arabic characters in names → accepted
- □ Very long StoreName (> 150 chars) → 400
- □ Concurrent registration with same mobile → one succeeds, one gets 409
- □ Register, don't verify, register again with same mobile → should succeed (no user created yet)
- □ Token expired (after 60 min) → 401 on protected endpoints

## 10. Known Limitations

| ID | Description | Impact |
|---|---|---|
| **C6** | JWT key and Twilio credentials are placeholders in appsettings.json | Must be set properly for staging/production |
| **H5** | AuthService is 690+ lines (god service) | Code complexity, not a testing issue |
| **P3** | OTP records are never cleaned up from database | Table grows indefinitely |
| **M6** | Rejected users cannot re-register | Mobile number is permanently blocked; no re-apply path |
| **Mock OTP** | In dev/staging, OTPs must be read from DB or logs | Tester needs DB access or log viewer |

---

## Appendix A: Quick Reference Card

### Mobile Format

`05XXXXXXXX` (10 digits, starts with 05)

### VAT Format

`3XXXXXXXXXXXXX3` (15 digits, starts and ends with 3)

### CRN Format

`XXXXXXXXXX` (10 digits)

### ShopCode Format

`XXXXXX` (6 uppercase alphanumeric)

### Postal Code Format

`XXXXX` (5 digits)

### OTP Format

`XXXXXX` (6 digits)

### JWT Token

- Access Token: 60-minute expiration
- Refresh Token: 7-day expiration
- Header format: `Authorization: Bearer {token}`

### Registration Status Flow

```
1 (PendingSalesman) → 2 (PendingZoneManager) → 3 (Approved)
                                                 or
1 (PendingSalesman) → 4 (Rejected)
2 (PendingZoneManager) → 4 (Rejected)
```

---

*End of QA Test Guide*