



COMPLETE BACKEND TESTING (POSTMAN GUIDE)

◆ STEP 0: PRE-CHECK (MANDATORY)

0.1 Start backend

```
uvicorn app.main:app --reload
```

0.2 Open Swagger once

```
http://127.0.0.1:8000/docs
```

Confirm:

- Server is running
 - No red errors
-



STEP 1: AUTHENTICATION (LOGIN)

Request

POST

```
http://127.0.0.1:8000/auth/login
```

Headers


Content-Type: application/json

Body (raw → JSON)

```
{
  "email": "admin@crm.com",
  "password": "admin123"
}
```

✅ Expected Response (200)

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs... ",
  "token_type": "bearer"
}
```

 **Copy the access_token**
You'll use it in all next steps.

STEP 2: SET TOKEN IN POSTMAN (ONCE)

In Postman:

- Go to **Authorization**
- Type: **Bearer Token**
- Paste token

- Save

👉 Now all requests in this collection are authenticated.

STEP 3: LEADS MODULE (ADMIN + AGENT)

3.1 Create Lead

POST

`/leads`

Body:

```
{  
  "business_name": "Royal Spice Restaurant",  
  "email": "royal@spice.com",  
  "industry": "Restaurant",  
  "city": "Sikar"  
}
```

✅ Expected:

```
{  
  "id": 1,  
  "status": "NEW"  
}
```

3.2 Duplicate Lead (Validation Test)

Send same request again.

 Expected:

400 Lead already exists

3.3 List Leads

GET

/leads

 Expected:

- List with created lead
 - Status = NEW
-



STEP 4: AI EMAIL GENERATION

4.1 Generate AI Email

POST

/ai/email?lead_id=1

✓ Expected:

```
{  
  "subject": "...",  
  "body": "..."  
}
```

✗ If lead_id invalid → 404

STEP 5: OUTREACH (STATUS CHANGE → CONTACTED)

5.1 Send Outreach

POST

`/outreach/send?lead_id=1`

✓ Expected:

```
{  
  "message": "Email sent successfully",  
  "lead_status": "CONTACTED"  
}
```

5.2 Verify Status Change

GET

/leads

✓ Status must now be:

CONTACTED

STEP 6: LEAD REPLY (INTERESTED)

6.1 Simulate Reply

POST

/replies/1

Body:

```
{  
  "reply": "Yes, I'm interested"  
}
```

✓ Expected:

```
{  
  "status": "INTERESTED"  
}
```

STEP 7: PROPOSAL GENERATION

7.1 Create Proposal

POST

`/proposals/1`

✓ Expected:

```
{  
  "proposal": "..."  
}
```

Lead status → `PROPOSAL_SENT`

STEP 8: PAYMENTS (ADMIN ONLY)

8.1 Create Payment Link

POST

`/payments/create/1`

✓ Expected:

```
{  
  "payment_url": "https://checkout.stripe.com/..."  
}
```

8.2 Pay Using Stripe Test Card

Open payment URL.

Use:

4242 4242 4242 4242

Any future date

Any CVC

8.3 Webhook Check

Stripe CLI terminal:

checkout.session.completed

Backend:

Lead status → PAID

STEP 9: ROLE TESTING (AGENT)

9.1 Login as AGENT

agent@crm.com

agent123

9.2 Try payment

POST

/payments/create/1

✗ Expected:

403 Insufficient permissions

👑 STEP 10: USER MANAGEMENT (ADMIN ONLY)

10.1 List Users

GET

/users

ADMIN → 

AGENT → ✗ 403

FINAL BACKEND HEALTH CHECK

Feature	Status
---------	--------


Auth	
------	---


JWT	
-----	---


RBAC	
------	---

Leads	
-------	---

AI 

Outreach 

Payments 

Webhook 

BONUS (PROFESSIONAL TIP)

Create **Postman Collection**:

CRM AutoPilot - Backend Tests

This is a **portfolio asset**.

WHAT NEXT?

You are **DONE** with backend verification.

Next powerful steps:

- 1 Build **Next.js Admin Dashboard**
- 2 Deploy to **Render**
- 3 Write **README + Architecture Diagram**
- 4 Start **Project 2 (Payroll System)**

👉 Tell me what you want next 