📘 Pishonserv Automation Backend — API Documentation

Version 1.0
Prepared for Backend Developer

This backend will power the automation between:

- Botpress
- Website (Pishonserv)
- Zoho CRM
- Vendors
- WhatsApp API
- Payment Provider (Paystack)

The backend must expose simple REST APIs that Botpress can call, and also send events back to Botpress through incoming webhooks.

All requests and responses should be JSON.

🔐 1. Authentication

Use **API Key (Bearer token)** in headers:

```
Authorization: Bearer <API_KEY>
Content-Type: application/json
```

You can generate and securely store the API key.

🏡 2. Property Management Endpoints

These endpoints help Botpress retrieve property and vendor details from the Pishonserv database.

2.1 Get Property Details

GET

```
/property/{property_id}
```

Response Example

```
{
  "status": true,
  "data": {
    "property_id": "P-2489",
    "name": "3 Bedroom Duplex",
    "location": "Lekki Phase 1",
    "price": 120000000,
    "bedrooms": 3,
    "type": "Duplex",
    "vendor_id": "V-1001",
    "available": true,
    "images": ["..."]
  }
}
```

2.2 Get Similar Properties

POST

```
/property/similar
```

Request Body

```
{
  "location": "Lekki",
  "price_range": { "min": 100000000, "max": 150000000 },
  "type": "Duplex",
  "bedrooms": 3
}
```

Response

```
{
  "status": true,
  "similar": [
    { "property_id": "P-1111", "name": "..." },
    { "property_id": "P-2222", "name": "..." }
  ]
}
```

👤 3. Vendor Communication

Botpress does NOT communicate with vendors directly.
Vendors reply via WhatsApp/SMS → Backend → Botpress.

3.1 Send Availability Request to Vendor

POST

```
/vendor/request
```

Body

```
{
  "vendor_id": "V-1001",
  "property_id": "P-2489",
  "conversation_id": "C-8891",
  "customer_question": "Is this still available?"
}
```

Backend Must Do:

- Fetch vendor contact
- Send message via WhatsApp API, SMS, or email
- Log notification status

Response

```
{ "status": true, "message": "vendor_notified" }
```

3.2 Vendor Reply Webhook (Vendor → Backend)

POST

```
/vendor/reply
```

Body

```
{
  "vendor_id": "V-1001",
  "conversation_id": "C-8891",
  "raw_message": "It's available. Call me on 08022233344."
}
```

Backend Must:

1. Store raw vendor message
2. **Sanitize** message (remove phone numbers/emails)
3. Forward sanitized reply to Botpress webhook

Example sanitized reply:

```
"It's available. [contact removed]"
```

Response

```
{ "status": true }
```

💬 4. Conversations

4.1 Create New Conversation Record

POST

```
/conversation/create
```

Body

```
{
  "conversation_id": "C-8891",
  "customer_id": "L-122",
  "vendor_id": "V-1001",
  "property_id": "P-2489",
  "channel": "whatsapp"
}
```

Response

```
{ "status": true }
```

💳 5. Payment Handling

5.1 Generate Payment Link

POST

```
/payment/initiate
```

Body

```
{
  "conversation_id": "C-8891",
  "property_id": "P-2489",
  "amount": 50000,
  "customer_name": "John Doe",
  "customer_email": "john@example.com"
}
```

Response

```
{
  "status": true,
  "payment_link": "https://paystack.com/pay/xyza123"
}
```

5.2 Payment Webhook (Paystack → Backend)

POST

`/payment/webhook`

Backend Must:

- Validate payment with Paystack API
- Update transaction record
- Forward event to Botpress → `/botpress/payment-success`
- Email vendor + customer confirmation

Botpress Notification Format

```
POST /botpress/payment-success
{
  "conversation_id": "C-8891",
  "payment_status": "success",
  "amount": 50000
}
```

🤖 6. Backend → Botpress Webhooks

Botpress will provide a URL like:

```
https://botpress.cloud/api/v2/bots/<bot_id>/incoming
```

Backend must hit this URL to send events.

6.1 Notify Botpress of Vendor Reply

POST

```
/botpress/vendor-reply
```

Body

```
{
  "type": "vendor_reply",
  "conversation_id": "C-8891",
  "message": "Yes, it's available."
}
```

6.2 Notify Botpress of Payment Success

POST

```
/botpress/payment-success
```

Body

```
{
  "type": "payment_success",
  "conversation_id": "C-8891",
  "amount": 50000
}
```

🗄️ 7. Database Structure (Recommended)

Developer may use relational or NoSQL.

Recommended tables:

```
properties
vendors
conversations
vendor_messages
customer_messages
payments
logs
```

🔒 8. Security Requirements

- All endpoints must require API key
- Sanitize vendor replies before sending to Botpress
- Validate all Paystack events
- Log all message traffic
- Use HTTPS only
- Add rate limiting

🚀 9. Notes for Backend Developer

- No UI required
- Pure API only
- Keep endpoints simple
- Return clean JSON only
- Botpress handles conversation logic
- Backend handles:
    - vendor comms
    - database
    - payment verification
    - sanitization
    - notifying Botpress