

# AI Dialer Pro - Backend API









Complete backend system for AI-powered Sales & Recruitment Dialer with emotion detection, real-time monitoring, and intelligent automation.

## Table of Contents






- [Features](#)
- [Tech Stack](#)
- [Prerequisites](#)
- [Installation](#)
- [Configuration](#)
- [Database Setup](#)
- [Running the Application](#)
- [API Documentation](#)
- [Architecture](#)
- [Deployment](#)




## Features

### Core Features





-  **Automated Calling Engine** - Twilio-powered outbound calling
-  **Real-time Transcription** - OpenAI Whisper integration
-  **Emotion Detection** - AI-powered sentiment analysis
-  **Intent Scoring** - NLP-based lead qualification
-  **Voice Personas** - Multiple AI voice personalities
-  **Custom Voice Cloning** - ElevenLabs integration
-  **Knowledge Base** - FAQ-powered responses
-  **DNC Management** - Compliance-first architecture

### Advanced Features

-  **Whisper Mode** - Agent monitoring without customer awareness
-  **Seamless Handoff** - Warm transfer with full context
-  **Objection Handlers** - Real-time battle cards
-  **Competitor Intel** - Automatic detection & response
-  **Best Time Prediction** - ML-powered call scheduling

-  **Auto-Retry Logic** - Smart retry based on outcomes
-  **CSAT Integration** - Post-call satisfaction surveys
-  **Script Optimization** - AI recommendations

## Analytics & Reporting

-  **Executive Dashboard** - KPIs and ROI metrics
-  **Cost per Lead** - Real-time cost tracking
-  **Conversion Funnels** - Visual analytics
-  **Emotion Analytics** - Aggregate sentiment tracking

## Tech Stack

- **Runtime:** Node.js 18+
- **Framework:** Express.js
- **Database:** PostgreSQL 15
- **Cache/Queue:** Redis 7
- **Telephony:** Twilio
- **Voice AI:** ElevenLabs
- **Speech-to-Text:** OpenAI Whisper
- **NLP:** OpenAI GPT-4
- **WebSockets:** ws
- **Job Queue:** Bull

## Prerequisites

Before you begin, ensure you have:

- Node.js (v18 or higher)
- PostgreSQL (v15 or higher)
- Redis (v7 or higher)
- Twilio Account (with phone number)
- OpenAI API Key
- ElevenLabs API Key (optional for custom voices)

## Installation

### 1. Clone Repository

```
bash
```

```
git clone https://github.com/your-org/ai-dialer-backend.git
cd ai-dialer-backend
```

## 2. Install Dependencies

```
bash

npm install
```

## 3. Environment Setup

```
bash

cp .env.example .env
```

Edit `.env` with your credentials.

## 4. Database Setup

```
bash

# Create database
createdb ai_dialer

# Run migrations
npm run migrate

# Seed sample data (optional)
npm run seed
```



## Configuration

### Required Environment Variables

```
bash
```

```
# Twilio (Required)
TWILIO_ACCOUNT_SID=your_account_sid
TWILIO_AUTH_TOKEN=your_auth_token
TWILIO_PHONE_NUMBER=+1234567890
```

```
# OpenAI (Required)
OPENAI_API_KEY=sk-your-key
```

```
# Database (Required)
DB_HOST=localhost
DB_PORT=5432
DB_NAME=ai_dialer
DB_USER=postgres
DB_PASSWORD=your_password
```

```
# Redis (Required)
REDIS_HOST=localhost
REDIS_PORT=6379
```

## Optional Integrations

```
bash

# ElevenLabs (Voice Cloning)
ELEVENLABS_API_KEY=your_key

# Emotion Detection
EMOTION_API_KEY=your_key

# CRM Integrations
SALESFORCE_CLIENT_ID=your_id
HUBSPOT_API_KEY=your_key
```

## Database Setup

### Automatic Setup (Recommended)

```
bash

npm run migrate
```

### Manual Setup

```
sql
```

```
-- Create database
```

```
CREATE DATABASE ai_dialer;
```

```
-- Run the schema from server.js createTables()
```

```
-- Or use the migration files
```

## Database Schema

Key tables:

- `organizations` - Multi-tenant support
- `users` - User authentication
- `campaigns` - Campaign management
- `contacts` - Lead/candidate database
- `calls` - Call records with AI metadata
- `call_events` - State machine tracking
- `scripts` - Conversation templates
- `voice_personas` - AI voice configurations
- `knowledge_base` - FAQ storage
- `dnc_registry` - Do Not Call list
- `audit_logs` - Compliance tracking



## Running the Application

### Development Mode

```
bash
```

```
npm run dev
```

Server runs on: `http://localhost:3000`

### Production Mode

```
bash
```

```
npm start
```

### Using Docker

```
bash
```

```
# Start all services
docker-compose up -d
```

```
# View logs
docker-compose logs -f api
```

```
# Stop services
docker-compose down
```

Access:

- API: <http://localhost:3000>
- pgAdmin: <http://localhost:5050>
- Redis Commander: <http://localhost:8081>



## API Documentation

### Authentication

All API endpoints require authentication (except health check).

```
bash

POST /api/v1/auth/login
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "password"
}
```

Response:

```
json

{
  "success": true,
  "token": "jwt_token_here",
  "user": { ... }
}
```

### Campaigns

#### Create Campaign

```
bash
```

```
POST /api/v1/campaigns
Authorization: Bearer <token>
```

```
{
  "organization_id": "uuid",
  "name": "Q4 Sales Outreach",
  "type": "sales",
  "voice_persona": "professional",
  "auto_retry": true,
  "best_time_enabled": true,
  "emotion_detection": true
}
```

## Get Campaigns

```
bash
```

```
GET /api/v1/campaigns?organization_id=uuid&status=active
```

## Contacts

### Bulk Upload

```
bash
```

```
POST /api/v1/contacts/bulk
```

```
{
  "organization_id": "uuid",
  "campaign_id": "uuid",
  "contacts": [
    {
      "first_name": "John",
      "phone": "+1234567890",
      "email": "john@example.com",
      "company": "TechCorp"
    }
  ]
}
```

## Calls

### Start Call

```
bash
```

```
POST /api/v1/calls/start
```

```
{  
  "organization_id": "uuid",  
  "campaign_id": "uuid",  
  "contact_id": "uuid"  
}
```

## Get Call History

```
bash
```

```
GET /api/v1/calls?organization_id=uuid&status=completed&limit=50
```

## Complete Call

```
bash
```

```
POST /api/v1/calls/complete
```

```
{  
  "call_id": "uuid",  
  "status": "completed",  
  "outcome": "scheduled",  
  "duration": 245,  
  "transcript": "Full transcript...",  
  "emotion": "interested",  
  "intent_score": 0.85,  
  "csat_score": 4.5  
}
```

## DNC Management

### Check DNC

```
bash
```

```
POST /api/v1/dnc/check
```

```
{  
  "organization_id": "uuid",  
  "phone": "+1234567890"  
}
```

### Add to DNC



bash

POST /api/v1/dnc/add

```
{  
  "organization_id": "uuid",  
  "phone": "+1234567890",  
  "reason": "User requested"  
}
```

## Knowledge Base

### Query KB

bash

POST /api/v1/knowledge/query

```
{  
  "organization_id": "uuid",  
  "question": "What is your pricing?"  
}
```

Response:

json

```
{  
  "success": true,  
  "answer": "Our pricing starts at $99/month",  
  "confidence": 0.92,  
  "should_fallback": false  
}
```

## Analytics

### Dashboard Stats

bash

GET /api/v1/analytics/dashboard?organization\_id=uuid

## ROI Calculator

bash

```
GET /api/v1/analytics/roi?organization_id=uuid&campaign_id=uuid
```

## ML Features

### Best Time Prediction

```
bash
```

```
GET /api/v1/ml/best-time/:contact_id
```

### Script Optimization

```
bash
```

```
GET /api/v1/ml/optimize-script/:script_id
```

## WebSocket Events

Connect to: `ws://localhost:3000`

Subscribe to call updates:

```
json

{
  "type": "subscribe_call",
  "call_id": "uuid"
}
```

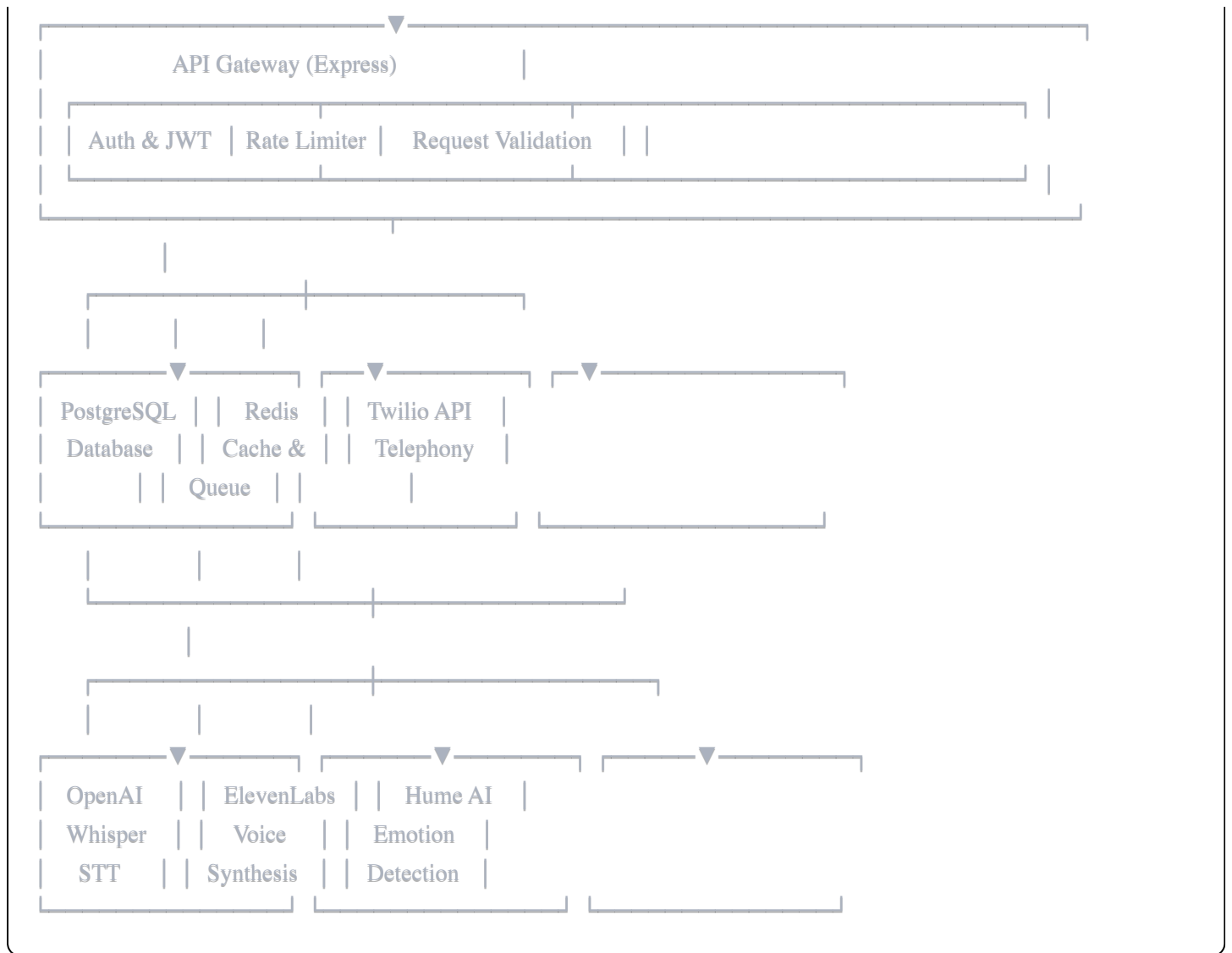
Receive real-time events:

- `call_update` - Status changes
- `transcript_update` - Live transcription
- `emotion_detected` - Emotion changes
- `competitor_mentioned` - Battle card triggers
- `handoff_request` - Agent notification



## Architecture

Frontend (React)  
WebSocket + REST API Communication



## Components

1. **API Server** - Express REST API + WebSocket
2. **Call Queue Worker** - Background job processor
3. **State Machine** - Call flow orchestration
4. **AI Integration Layer** - OpenAI, ElevenLabs, Hume AI
5. **Telephony Handler** - Twilio webhooks
6. **Real-time Engine** - Redis Pub/Sub + WebSockets

## Security

- JWT-based authentication
- Rate limiting on all endpoints
- Input validation with Joi
- SQL injection prevention (parameterized queries)
- XSS protection with Helmet
- PII encryption at rest (AES-256)

- HTTPS enforced in production
- CORS configured
- Audit logging for compliance



## Monitoring

### Health Check

```
bash
```

```
GET /health
```

### Metrics

- Request latency
- Error rates
- Call success rates
- Credits consumed
- Queue depth

### Logging

Uses Winston for structured logging:

- `logs/app.log` - Application logs
- `logs/error.log` - Error logs
- Console output in development



## Deployment

### Heroku

```
bash
```

```
heroku create ai-dialer-api
```

```
heroku addons:create heroku-postgresql:hobby-dev
```

```
heroku addons:create heroku-redis:hobby-dev
```

```
heroku config:set TWILIO_ACCOUNT_SID=xxx
```

```
git push heroku main
```

### AWS EC2

```
bash
```

```
# Install dependencies
```

```
sudo apt update
```

```
sudo apt install nodejs npm postgresql redis-server
```

```
# Clone and setup
```

```
git clone <repo>
```

```
cd ai-dialer-backend
```

```
npm install
```

```
npm run migrate
```

```
pm2 start server.js --name ai-dialer-api
```

## Docker

```
bash
```

```
docker build -t ai-dialer-api .
```

```
docker run -p 3000:3000 --env-file .env ai-dialer-api
```



## Testing

```
bash
```

```
# Run all tests
```

```
npm test
```

```
# Run with coverage
```

```
npm run test:coverage
```

```
# Run specific test
```

```
npm test -- campaigns.test.js
```



## Contributing

1. Fork the repository
2. Create feature branch ((`git checkout -b feature/amazing-feature`))
3. Commit changes ((`git commit -m 'Add amazing feature'`))
4. Push to branch ((`git push origin feature/amazing-feature`))
5. Open Pull Request



## License

MIT License - see LICENSE file for details



## Support

- Documentation: <https://docs.ai-dialer.com>
- Email: [support@ai-dialer.com](mailto:support@ai-dialer.com)
- Slack: <https://ai-dialer.slack.com>

## Roadmap

- ☐ Multi-language support (Hindi, Spanish)
- ☐ Video calling capabilities
- ☐ Advanced analytics dashboard
- ☐ Mobile SDK
- ☐ Zapier integration
- ☐ Chrome extension

---

Built with  by AI Dialer Team