Notification System - FastAPI + RabbitMQ + PostgreSQL

Python: 3.9+ | FastAPI: 0.95.0 | Docker: Latest

A scalable notification system supporting email, SMS, and in-app notifications with queue-based processing and retry mechanisms.

Features

- ✅ Multiple notification types (Email/SMS/In-App)
- 🚀 Asynchronous processing with RabbitMQ
- 🔄 Automatic retries for failed notifications
- 🐘 PostgreSQL data persistence
- 🐳 Dockerized services

Prerequisites

- Docker and Docker Compose
- Python 3.9+
- PyCharm (recommended) or any Python IDE

Setup Instructions

1. Clone the Repository

Bash

```
git clone https://github.com/Premkumarreddy-datascience/Notification_service_application.git
cd Notification_service_application
```

2. Project Structure

```
notification_system/
├── api/
|   ├── app/
|   |   ├── __init__.py
|   |   ├── main.py        # FastAPI application
|   |   ├── models.py      # Database models
|   |   ├── schemas.py     # Pydantic schemas
|   |   └── database.py    # DB connection
|   ├── requirements.txt
|   └── Dockerfile
├── worker/
|   ├── worker.py          # Notification processor
|   ├── requirements.txt
|   └── Dockerfile
├── docker-compose.yml     # Service definitions
└── .env.example           # Environment template
```

3. Environment Setup

1. **Copy the example env file:**

bash

```
cp .env.example .env
```

2. **Update .env with your credentials:**

ini

```
DATABASE_URL=postgresql://postgres:postgres@db:5432/notification_db

RABBITMQ_URL=amqp://admin:admin@rabbitmq:5672/
```

4. Start Services with Docker

bash

```
docker-compose up -d --build
```

This will start:

- PostgreSQL database
- RabbitMQ server
- FastAPI application
- Worker service

5. Initialize Database

bash

```
docker-compose exec api python -c "from app.database import Base, engine;
Base.metadata.create_all(bind=engine)"
```

6. Test the API

Send a Notification

bash

```
curl -X POST "http://localhost:8000/notifications/" \

-H "Content-Type: application/json" \

-d '{"user_id": 1, "title": "Test", "message": "Hello World"}'
```

Get User Notifications

bash

```
curl http://localhost:8000/users/1/notifications/
```

Development Setup (PyCharm)

1. **Mark Sources Root**:
   - Right-click api/ → Mark Directory as → Sources Root
2. **Configure Python Interpreter**:
   - Use Python 3.9+ virtual environment
3. **Run Configurations**:
   - For api/app/main.py:
     - Environment variables:

```ini
DATABASE_URL=postgresql://postgres:postgres@localhost:5432/notification_db
RABBITMQ_URL=amqp://admin:admin@localhost:5672/
```

Assumptions

1. **SMTP Configuration**:
   - The worker assumes an SMTP server is available
   - Update SMTP credentials in .env for email notifications
2. **SMS Service**:
   - Currently uses mock implementation
   - Replace send_sms() in worker.py with real provider (Twilio, etc.)
3. **Database**:
   - PostgreSQL is used as the primary datastore
   - Tables are auto-created on first run

Monitoring

- **RabbitMQ Management**: http://localhost:15672 (admin/admin)
- **API Docs**: http://localhost:8000/docs

Troubleshooting

Issue: Import errors
Solution: Ensure __init__.py exists in all package directories and PyCharm sources root is set correctly

Issue: Docker compose fails
Solution: Check logs with `docker-compose logs -f`