OTA Integration Documentation

Overview

The OTA (Online Travel Agency) Integration Service provides seamless connectivity between your hostel management system and major booking platforms like Booking.com and HostelWorld. This service handles real-time inventory synchronization, rate management, and reservation processing.

Features

- **Multi-OTA Support**: Booking.com and HostelWorld integration
- **Real-time Synchronization**: Inventory and rates sync automatically
- **Webhook Processing**: Handle incoming reservations instantly
- **Queue Management**: Reliable message processing with Redis
- **Health Monitoring**: Built-in health checks and metrics
- **Error Handling**: Automatic retries and fallback mechanisms
- **Security**: Webhook signature verification and rate limiting

Architecture

```
┌─────────────────┐  ┌─────────────────┐  ┌─────────────────┐
│   Booking.com   │  │   HostelWorld   │  │   Other OTAs    │
└────────┬────────┘  └────────┬────────┘  └────────┬────────┘
         │                    │                    │
         └────────────────────┼────────────────────┘
                              │
                   ┌──────────┴──────────┐
                   │ OTA Integration Service │
                   │    (Port 3003)      │
                   └──────────┬──────────┘
                              │
                   ┌──────────┴──────────┐
                   │    Redis Queue      │
                   │  (Message Broker)   │
                   └──────────┬──────────┘
                              │
                   ┌──────────┴──────────┐
                   │ Hostel Management   │
                   │      System         │
                   └─────────────────────┘
```

API Endpoints

Health Check

```http
GET /health
```

Returns service health status and OTA connection status.

Inventory Sync

```http
```

```
POST /api/ota/sync-inventory
Content-Type: application/json

{
  "roomId": 1,
  "date": "2024-01-15",
  "available": 10,
  "price": 50.00,
  "currency": "USD"
}
```

Rate Sync

```http
POST /api/ota/sync-rates
Content-Type: application/json

{
  "roomId": 1,
  "dateFrom": "2024-01-15",
  "dateTo": "2024-01-20",
  "rates": [
    {"date": "2024-01-15", "price": 50.00, "currency": "USD"},
    {"date": "2024-01-16", "price": 55.00, "currency": "USD"}
  ]
}
```

Webhooks

- `POST /api/webhooks/booking` - Booking.com webhooks
- `POST /api/webhooks/hostelworld` - HostelWorld webhooks

Statistics

```http
GET /api/ota/stats
```

Returns synchronization statistics and performance metrics.

Configuration

Environment Variables

Server Configuration

- `PORT`: Service port (default: 3003)
- `NODE_ENV`: Environment (development/production)

Redis Configuration

- `REDIS_HOST`: Redis server host
- `REDIS_PORT`: Redis server port (default: 6379)
- `REDIS_PASSWORD`: Redis password (optional)

Booking.com Configuration

- `BOOKING_API_KEY`: Booking.com API key
- `BOOKING_USERNAME`: Booking.com username
- `BOOKING_PASSWORD`: Booking.com password
- `BOOKING_HOTEL_ID`: Your property ID on Booking.com
- `BOOKING_WEBHOOK_SECRET`: Webhook signature secret

HostelWorld Configuration

- `HOSTELWORLD_API_KEY`: HostelWorld API key
- `HOSTELWORLD_API_SECRET`: HostelWorld API secret
- `HOSTELWORLD_PROPERTY_ID`: Your property ID on HostelWorld
- `HOSTELWORLD_WEBHOOK_SECRET`: Webhook signature secret

Setup Instructions

Local Development

1. **Clone and Navigate**

```bash
git clone https://github.com/RasPutinnn/hostel-business-strategy.git
cd hostel-business-strategy
```

2. **Run Setup Script**

```bash
```

```
chmod +x scripts/ota-setup/setup-ota.sh
./scripts/ota-setup/setup-ota.sh
```

3. **Configure Environment**

```bash
cp services/ota-integration-service/.env.example services/ota-integration-service/.env
# Edit .env with your actual OTA credentials
```

4. **Start Services**

```bash
docker-compose up -d ota-integration redis
```

5. **Verify Installation**

```bash
curl http://localhost:3003/health
```

Production Deployment

1. **Set Environment Variables**

```bash
export BOOKING_API_KEY="your_booking_api_key"
export HOSTELWORLD_API_KEY="your_hostelworld_api_key"
# ... other variables
```

2. **Deploy to Kubernetes**

```bash
chmod +x scripts/ota-setup/deploy-ota.sh
./scripts/ota-setup/deploy-ota.sh --namespace production --tag latest
```

Testing

Unit Tests

```bash
cd services/ota-integration-service
npm test
```

Integration Tests

```bash
npm run test:integration
```

Load Testing

```bash
# Install artillery first: npm install -g artillery
artillery run tests/load-test.yml
```

Monitoring

Health Checks

The service provides comprehensive health checks:

- Service availability
- OTA API connectivity
- Redis connection
- Queue status

Metrics

Available metrics include:

- Request count and response times
- Sync success/failure rates
- Queue length and processing times
- OTA API response times

Logging

Structured logging with different levels:

- `ERROR`: Service errors and failures
- `WARN`: Recoverable issues and warnings
- `INFO`: General information and successful operations
- `DEBUG`: Detailed debugging information

Webhook Configuration

Booking.com Webhooks

1. **Login to Booking.com Partner Hub**
2. **Navigate to Connectivity Settings**
3. **Add Webhook URL**: `https://your-domain.com/api/webhooks/booking`
4. **Set Secret**: Use the value from `BOOKING_WEBHOOK_SECRET`
5. **Select Events**: reservation_created, reservation_modified, reservation_cancelled

HostelWorld Webhooks

1. **Login to HostelWorld Partner Portal**
2. **Go to Integration Settings**
3. **Add Webhook URL**: `https://your-domain.com/api/webhooks/hostelworld`
4. **Set Secret**: Use the value from `HOSTELWORLD_WEBHOOK_SECRET`
5. **Enable Events**: booking_created, booking_updated, booking_cancelled

Troubleshooting

Common Issues

Service Not Starting

```bash
# Check logs
docker-compose logs ota-integration

# Common causes:
# - Missing environment variables
# - Redis connection issues
# - Port conflicts
```

OTA API Connection Issues

```bash
```

```
# Test API connectivity
curl http://localhost:3003/api/ota/stats

# Check API credentials in .env file
# Verify API endpoints are accessible
```

Webhook Not Receiving Data

```bash
# Check webhook configuration in OTA admin panels
# Verify webhook URLs are accessible from internet
# Check webhook signature verification
```

Queue Processing Issues

```bash
# Check Redis connection
docker-compose exec redis redis-cli ping

# Monitor queues
curl http://localhost:3003/api/ota/stats
```

Performance Tuning

Queue Concurrency

Adjust queue concurrency in environment variables:

```env
QUEUE_CONCURRENCY_BOOKING=10
QUEUE_CONCURRENCY_HOSTELWORLD=10
QUEUE_CONCURRENCY_RATES=5
```

Rate Limiting

Configure rate limiting:

```env
```

```
RATE_LIMIT_WINDOW_MS=900000  # 15 minutes
RATE_LIMIT_MAX_REQUESTS=200  # requests per window
```

Redis Optimization

```bash
# Monitor Redis memory usage
docker-compose exec redis redis-cli info memory

# Configure Redis maxmemory policy
# Set appropriate persistence settings
```

Security Considerations

Webhook Security

- Always verify webhook signatures
- Use HTTPS for webhook endpoints
- Implement rate limiting
- Log and monitor webhook requests

API Security

- Store credentials securely (use secrets management)
- Rotate API keys regularly
- Monitor API usage and quotas
- Implement proper error handling to avoid information leakage

Network Security

- Use firewalls to restrict access
- Implement VPN for internal communication
- Regular security updates
- Monitor for suspicious activities

API Rate Limits

Booking.com

- Standard: 1000 requests/hour
- Burst: 10 requests/second
- Webhook: No specific limits

HostelWorld

- Standard: 500 requests/hour
- Burst: 5 requests/second
- Webhook: 100 requests/minute

Support and Maintenance

Regular Tasks

- Monitor service health and performance
- Review and rotate API credentials
- Update dependencies and security patches
- Backup configuration and logs
- Test webhook endpoints periodically

Scaling Considerations

- Horizontal scaling: Add more service replicas
- Vertical scaling: Increase resource limits
- Queue optimization: Tune concurrency settings
- Database optimization: Monitor and optimize queries

Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Add tests for new features
5. Run the test suite
6. Submit a pull request

License

This project is licensed under the MIT License - see the LICENSE file for details.

Changelog

v1.0.0 (Initial Release)

- Booking.com integration
- HostelWorld integration
- Basic inventory and rate sync
- Webhook processing
- Queue management with Redis
- Health monitoring
- Docker and Kubernetes support