

GARUDAHOKI WhatsApp Bot - LiveChatInc Integration Guide

Panduan Integrasi dengan LiveChatInc (platform.text.com)

Overview

Panduan ini akan membantu Anda mengintegrasikan bot WhatsApp GARUDAHOKI dengan LiveChatInc untuk handover otomatis ke agen manusia (Beatrice) ketika customer memilih opsi “Bicara dengan Agen”.

Langkah 1: Setup Authentication LiveChatInc

A. Akses Developer Console

1. Login ke Platform Text

...

2. Kunjungi <https://platform.text.com/console>

3. Login dengan akun LiveChatInc yang sudah ada

4. Navigate ke “Apps” section

...

5. Buat New Application

...

6. Klik “Create App”

7. App Details:

- App Name: “GARUDAHOKI WhatsApp Bot”
- App Type: “Server-side app”
- Description: “WhatsApp bot integration for customer handover”

8. Klik “Create App”

...

B. Konfigurasi OAuth 2.1

1. Setup OAuth Credentials

...

2. Di app dashboard, pilih “Authorization”

3. OAuth Settings:

- Grant Type: Authorization Code
- Redirect URI: <https://yourdomain.com/auth/livechat/callback>
- Scopes: pilih yang diperlukan:
 - chats-all:rw (read/write chats)
 - agents-all:rw (manage agents)

- webhooks-all:rw (webhook management)

4. Save configuration

...

5. Dapatkan Credentials

Catat informasi berikut:

- Client ID: [your_client_id]
- Client Secret: [your_client_secret]
- License ID: [your_license_id] (dari account settings)

C. Setup Personal Access Token (untuk Testing)

1. Generate PAT

...

2. Di Developer Console, pilih "Tools" > "Personal Access Tokens"

3. Klik "Create Token"

4. Token Settings:

- Name: "GARUDAHOKI Bot Token"
- Scopes: chats:rw, agents:rw

5. Copy token yang dihasilkan

...



Langkah 2: Konfigurasi Webhook LiveChatInc

A. Setup Webhook untuk Incoming Chats

1. Buat Webhook di Developer Console

...

2. Di app dashboard, pilih "Webhooks"

3. Klik "Create Webhook"

4. Webhook Configuration:

- Name: "GARUDAHOKI Bot Webhook"
- URL: https://yourdomain.com/webhook/livechat
- Events: pilih events yang diperlukan:
 - incoming_chat
 - chat_transferred
 - agent_status_changed

5. Save webhook

...

6. Verify Webhook

...

7. Test webhook dengan sample payload

8. Verify endpoint responds dengan status 200

9. Check webhook status di dashboard

...

B. Setup Agent Assignment

1. Konfigurasi Agen Beatrice

- ...
- 2. Di LiveChat dashboard, pilih "Agents"
- 3. Find agent "Beatrice"
- 4. Note Agent ID untuk konfigurasi bot
- 5. Verify agent status dan availability
- ...

Langkah 3: Update Environment Variables

A. Tambahkan LiveChatInc Credentials

Edit file `.env` dan tambahkan konfigurasi berikut:

```
# LiveChatInc Integration
LIVECHAT_LICENSE_ID=your_license_id
LIVECHAT_CLIENT_ID=your_client_id
LIVECHAT_CLIENT_SECRET=your_client_secret
LIVECHAT_REDIRECT_URI=https://yourdomain.com/auth/livechat/callback

# Personal Access Token (untuk testing)
LIVECHAT_PAT=your_personal_access_token

# Agent Configuration
LIVECHAT_AGENT_BEATRICE_ID=beatrice_agent_id
LIVECHAT_DEFAULT_GROUP_ID=0

# API Configuration
LIVECHAT_API_VERSION=v3.5
LIVECHAT_API_BASE_URL=https://api.livechatinc.com
```

B. Restart Application

```
# Restart bot application
pm2 restart garudahoki-bot

# Verify environment variables loaded
pm2 logs garudahoki-bot --lines 20
```

Langkah 4: Test Integration

A. Test API Connection

1. Test Authentication

```
bash
# Test dengan Personal Access Token
curl -X GET "https://api.livechatinc.com/v3.5/configuration/webhooks" \
```

```
-H "Authorization: Bearer YOUR_PAT" \
-H "X-API-Version: 3.5"
```

2. Test Agent List

```
bash
# Get list of agents
curl -X GET "https://api.livechatinc.com/v3.5/configuration/agents" \
-H "Authorization: Bearer YOUR_PAT" \
-H "X-API-Version: 3.5"
```

B. Test Bot Handover Flow

1. Test via WhatsApp

...

2. Kirim pesan ke bot WhatsApp
3. Pilih menu “b” atau “Bicara dengan Agen”
4. Jelaskan masalah singkat
5. Verify bot mengirim data ke LiveChatInc
6. Check di LiveChat dashboard apakah chat baru muncul

...

7. Verify di Admin Panel

...

8. Login ke admin panel: <https://yourdomain.com/admin>
9. Check “Chat Logs” untuk melihat handover history
10. Verify integration status di dashboard

...



Langkah 5: Monitoring & Analytics

A. Setup Monitoring

1. LiveChat Analytics

...

2. Di LiveChat dashboard, pilih “Analytics”
3. Monitor metrics:

- Chat volume dari WhatsApp
- Response time agen Beatrice
- Customer satisfaction

...

4. Bot Analytics

...

5. Di admin panel bot, check:
 - Total handovers ke LiveChat
 - Success rate integrasi

- Error logs

...

B. Setup Alerts

1. Webhook Failure Alerts

...

2. Configure monitoring untuk webhook failures
3. Setup email alerts untuk integration errors
4. Monitor API rate limits

...



Langkah 6: Advanced Configuration

A. Custom Chat Routing

1. Setup Routing Rules

```
javascript
// Contoh routing berdasarkan keyword
const routingRules = {
  'deposit': 'agent_finance_id',
  'withdraw': 'agent_finance_id',
  'bonus': 'agent_promo_id',
  'technical': 'agent_tech_id',
  'default': 'beatrice_agent_id'
};
```

2. Implement Smart Routing

...

3. Analyze customer message content
4. Route to appropriate agent based on keywords
5. Fallback to Beatrice for general inquiries

...

B. Chat Context Transfer

1. Transfer Chat History

```
javascript
// Include chat context when creating LiveChat session
const chatContext = {
  whatsapp_number: customerPhone,
  initial_message: customerMessage,
  bot_interaction_history: chatHistory,
  customer_timezone: 'Asia/Jakarta'
};
```

2. Custom Fields

...

3. Setup custom fields di LiveChat:

- WhatsApp Number

- Bot Interaction Count
- Last Bot Response
- Customer Segment (VIP, Regular, New)
- ...

API Integration Code Examples

A. Create Chat Session

```
// Function untuk membuat chat session di LiveChatInc
async function createLiveChatSession(customerData, message) {
  const payload = {
    chat: {
      users: [{
        name: customerData.name || 'WhatsApp Customer',
        email: `${customerData.phone}@whatsapp.customer`,
        session_fields: [{
          whatsapp_number: customerData.phone,
          source: 'GARUDAHOKI WhatsApp Bot'
        }]
      }],
      properties: {
        source: {
          type: 'whatsapp'
        },
        initial_message: {
          text: message
        }
      }
    }
  };

  try {
    const response = await fetch(`${LIVECHAT_API_BASE_URL}/v3.5/agent/action/
start_chat`, {
      method: 'POST',
      headers: {
        'Authorization': `Bearer ${LIVECHAT_PAT}`,
        'Content-Type': 'application/json',
        'X-API-Version': '3.5'
      },
      body: JSON.stringify(payload)
    });

    const result = await response.json();
    return result;
  } catch (error) {
    console.error('LiveChat integration error:', error);
    throw error;
  }
}
```

B. Transfer to Specific Agent

```
// Function untuk transfer chat ke agen Beatrice
async function transferToAgent(chatId, agentId, message) {
  const payload = {
    target: {
      type: 'agent',
      ids: [agentId]
    },
    force: false
  };

  try {
    const response = await fetch(`${LIVECHAT_API_BASE_URL}/v3.5/agent/action/transfer_chat`, {
      method: 'POST',
      headers: {
        'Authorization': `Bearer ${LIVECHAT_PAT}`,
        'Content-Type': 'application/json',
        'X-API-Version': '3.5'
      },
      body: JSON.stringify({
        chat_id: chatId,
        ...payload
      })
    });

    return await response.json();
  } catch (error) {
    console.error('Transfer to agent error:', error);
    throw error;
  }
}
```

C. Send Message to Chat

```
// Function untuk mengirim pesan ke LiveChat
async function sendMessageToLiveChat(chatId, message, authorId) {
  const payload = {
    chat_id: chatId,
    event: {
      type: 'message',
      text: message,
      author_id: authorId
    }
  };

  try {
    const response = await fetch(`${LIVECHAT_API_BASE_URL}/v3.5/agent/action/send_event`, {
      method: 'POST',
      headers: {
        'Authorization': `Bearer ${LIVECHAT_PAT}`,
        'Content-Type': 'application/json',
        'X-API-Version': '3.5'
      },
      body: JSON.stringify(payload)
    });

    return await response.json();
  } catch (error) {
    console.error('Send message error:', error);
    throw error;
  }
}
```

Troubleshooting

Common Issues

1. Authentication Errors

Error: "Invalid access token"

Solution:

- Verify PAT is correct dan belum expired
- Check scopes include required permissions
- Regenerate token jika perlu

2. Webhook Not Receiving Events

Error: Webhook tidak menerima events dari LiveChat

Solution:

- Verify webhook URL accessible via HTTPS
- Check webhook configuration di Developer Console
- Test endpoint manually dengan curl

3. Agent Not Available

Error: "Agent Beatrice not available"

Solution:

- Check agent status di LiveChat dashboard

- Verify agent ID correct
- Setup fallback agent atau queue

4. Rate Limiting

Error: "Too many requests"

Solution:

- Implement rate limiting di bot
- Use exponential backoff untuk retries
- Monitor API usage di Developer Console

Debug Steps

1. Check API Logs

```
```bash
Check bot logs
pm2 logs garudahoki-bot | grep -i livechat

Check specific error patterns
grep -i "livechat error" /path/to/logs/bot.log
```
```

1. Test API Endpoints

```
```bash
Test authentication
curl -X GET "https://api.livechatinc.com/v3.5/configuration/license" \
-H "Authorization: Bearer YOUR_PAT"

Test webhook
curl -X POST "https://yourdomain.com/webhook/livechat" \
-H "Content-Type: application/json" \
-d '{"test": "webhook"}'
```
```

1. Verify Environment Variables

```
```bash
Check environment variables
env | grep LIVECHAT

Test configuration
node -e "console.log(process.env.LIVECHAT_LICENSE_ID)"
```
```



Best Practices

A. Security

1. Token Management

- Store tokens securely dalam environment variables
- Rotate tokens regularly
- Use minimum required scopes
- Monitor token usage

2. Data Privacy

- Encrypt sensitive customer data
- Comply dengan GDPR/privacy regulations
- Log minimal necessary information
- Implement data retention policies

B. Performance

1. API Optimization

- Implement caching untuk agent status
- Use batch operations when possible
- Monitor API rate limits
- Implement circuit breaker pattern

2. Error Handling

- Implement retry logic dengan exponential backoff
- Handle network timeouts gracefully
- Provide fallback mechanisms
- Log errors untuk monitoring

C. User Experience

1. Smooth Handover

- Provide clear transition messages
- Transfer complete chat context
- Set proper expectations untuk response time
- Follow up on handover success

2. Agent Experience

- Provide rich customer context
- Include chat history dan bot interactions
- Set up proper notifications
- Enable easy escalation paths

Support & Resources

Documentation Links

- LiveChatInc API Docs: <https://platform.text.com/docs>
- Authentication Guide: <https://platform.text.com/docs/authorization>
- Webhook Setup: <https://platform.text.com/docs/guides/api-guides>
- Developer Console: <https://platform.text.com/console>

Community Support

- Discord: Join developer community
- GitHub: Contribute dan report issues
- Email: developers@text.com
- Twitter: Follow @LiveChatInc untuk updates

Testing Tools

- Postman Collection: Import LiveChatInc API collection

- Webhook Tester: Use webhook.site untuk testing
- API Explorer: Test endpoints di Developer Console

 **Integration Complete! Bot WhatsApp GARUDAHOKI sekarang terhubung dengan LiveChatInc untuk handover seamless ke agen Beatrice.**
