

Construction CRM Webhook Integration Guide

Overview

This guide explains how to integrate your Construction CRM Platform with n8n (or any other automation tool) to automatically create change orders via webhook.

Webhook Endpoint

URL: `https://your-domain.com/api/webhooks/change-orders`

Method: `POST`

Content-Type: `application/json`

Authentication

The webhook uses a secret key for authentication. Your webhook secret is stored in the `.env` file:

```
WEBHOOK_SECRET=34d7412dfa4d1c54106d4c5129c6a312061a3d8b50966dbe6000124cbece9890
```

Important: Keep this secret secure and never commit it to version control.

Request Payload

Required Fields

Field	Type	Description
<code>webhookSecret</code>	string	Your webhook authentication secret
<code>projectId</code>	string	UUID of the project (must exist in database)
<code>title</code>	string	Title of the change order
<code>proposedCost</code>	number	Proposed cost for the change order

Optional Fields

Field	Type	Default	Description
description	string	""	Detailed description of the change order
type	string	"Scope"	Type of change order (see Types section)
priority	string	"Normal"	Priority level (see Priority section)
scheduleImpact	number	null	Number of days impact on schedule
reason	string	""	Reason for the change
justification	string	""	Justification for the change
requestedByEmail	string	Project Manager	Email of user requesting the change
attachments	array	[]	Array of attachment URLs or file references

Types

Valid values for the `type` field:

- `Scope` - Scope change
- `Schedule` - Schedule change
- `Budget` - Budget change
- `Design` - Design change
- `Unforeseen` - Unforeseen conditions
- `ClientRequested` - Client requested change
- `RegulatoryCompliance` - Regulatory compliance change
- `Other` - Other type of change

Priority Levels

Valid values for the `priority` field:

- `Low` - Low priority
- `Normal` - Normal priority (default)
- `High` - High priority
- `Critical` - Critical priority

Example Request

Using cURL

```
curl -X POST https://your-domain.com/api/webhooks/change-orders \
-H "Content-Type: application/json" \
-d '{
  "webhookSecret": "your-secret-key-here",
  "projectId": "550e8400-e29b-41d4-a716-446655440000",
  "title": "Additional HVAC Units Required",
  "description": "Client requested 3 additional HVAC units for the third floor",
  "type": "ClientRequested",
  "priority": "High",
  "proposedCost": 45000,
  "scheduleImpact": 7,
  "reason": "Client expansion requirements",
  "justification": "Additional cooling capacity needed for increased occupancy",
  "requestedByEmail": "john@doe.com",
  "attachments": []
}'
```

Using JavaScript/Node.js

```
const axios = require('axios');

const payload = {
  webhookSecret: 'your-secret-key-here',
  projectId: '550e8400-e29b-41d4-a716-446655440000',
  title: 'Additional HVAC Units Required',
  description: 'Client requested 3 additional HVAC units for the third floor',
  type: 'ClientRequested',
  priority: 'High',
  proposedCost: 45000,
  scheduleImpact: 7,
  reason: 'Client expansion requirements',
  justification: 'Additional cooling capacity needed for increased occupancy',
  requestedByEmail: 'john@doe.com',
  attachments: []
};

axios.post('https://your-domain.com/api/webhooks/change-orders', payload)
  .then(response => {
    console.log('Success:', response.data);
  })
  .catch(error => {
    console.error('Error:', error.response.data);
  });
```

Using n8n

1. Add HTTP Request Node

- Method: POST
- URL: https://your-domain.com/api/webhooks/change-orders
- Authentication: None (using webhook secret in body)
- Body Content Type: JSON

2. Configure JSON Body:

```
{
  "webhookSecret": "{{${env.WEBHOOK_SECRET}}",
  "projectId": "{{${json.projectId}}",
  "title": "{{${json.title}}",
  "description": "{{${json.description}}",
  "type": "{{${json.type}}",
  "priority": "{{${json.priority}}",
  "proposedCost": {{${json.proposedCost}},
  "scheduleImpact": {{${json.scheduleImpact}},
  "reason": "{{${json.reason}}",
  "justification": "{{${json.justification}}",
  "requestedByEmail": "{{${json.requestedByEmail}}"}
}
```

1. Store Webhook Secret in n8n:

- Go to Settings > Credentials
- Create a new credential or use environment variables
- Set `WEBHOOK_SECRET` with your secret key

Response Format

Success Response (201 Created)

```
{
  "success": true,
  "message": "Change order created successfully",
  "changeOrder": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "changeOrderNumber": "CO-2025-0001",
    "title": "Additional HVAC Units Required",
    "status": "Draft",
    "proposedCost": 45000,
    "project": {
      "name": "Downtown Office Complex",
      "projectNumber": "PRJ-2024-001"
    },
    "requestedBy": {
      "firstName": "John",
      "lastName": "Doe",
      "email": "john@doe.com",
      "role": "Admin"
    }
  }
}
```

Error Responses

401 Unauthorized - Invalid Webhook Secret

```
{
  "error": "Invalid webhook secret"
}
```

400 Bad Request - Missing Required Fields

```
{
  "error": "Missing required fields",
  "required": ["projectId", "title", "proposedCost"]
}
```

404 Not Found - Project Not Found

```
{
  "error": "Project not found"
}
```

500 Internal Server Error

```
{
  "error": "Failed to process webhook",
  "message": "Detailed error message"
}
```

Getting Project IDs

To get a list of valid project IDs for your webhook:

1. **Via API:** Make an authenticated GET request to `/api/projects`
2. **Via Dashboard:** Navigate to the Projects page in your CRM
3. **Via Database:** Query the `Project` table directly

Example API call:

```
curl -H "Cookie: your-session-cookie" \
https://your-domain.com/api/projects
```

Testing the Webhook

Test Endpoint Availability

You can test if the webhook is accessible by making a GET request:

```
curl https://your-domain.com/api/webhooks/change-orders
```

This will return documentation about the endpoint.

Test with Sample Data

Replace `PROJECT_ID` with an actual project ID from your database:

```
curl -X POST https://your-domain.com/api/webhooks/change-orders \
-H "Content-Type: application/json" \
-d '{
  "webhookSecret": "34d7412dfa4d1c54106d4c5129c6a312061a3d8b50966dbe6000124cbe-
ce9890",
  "projectId": "PROJECT_ID",
  "title": "Test Change Order",
  "proposedCost": 1000
}'
```

Security Best Practices

1. **Rotate Webhook Secret Regularly:** Update your `WEBHOOK_SECRET` in `.env` periodically
2. **Use HTTPS:** Always use HTTPS in production to encrypt webhook payloads
3. **Validate Requests:** The webhook automatically validates:
 - Webhook secret authentication
 - Required field presence
 - Project existence
 - User email (if provided)
4. **Monitor Logs:** Check server logs for webhook activity and errors
5. **Rate Limiting:** Consider implementing rate limiting if needed

Troubleshooting

Common Issues

1. **“Invalid webhook secret”**
 - Verify the webhook secret matches the one in `.env`
 - Check for extra spaces or characters in the secret
2. **“Project not found”**
 - Ensure the project ID exists in your database
 - Verify you’re using the UUID, not the project number
3. **“Missing required fields”**
 - Check that all required fields are present in the payload
 - Ensure field names match exactly (case-sensitive)
4. **Connection Timeout**
 - Verify the URL is correct
 - Check that your server is running and accessible
 - Ensure no firewall is blocking the request

Support

For additional help:

- Check server logs for detailed error messages
- Review the API documentation at `/api/webhooks/change-orders` (GET request)
- Test with minimal payload first, then add optional fields

Changelog

Version 1.0 (2025-01-07)

- Initial webhook implementation
- Support for change order creation
- Webhook secret authentication
- Comprehensive validation and error handling