

Integração n8n + Kommo

Observações Iniciais

Este documento descreve a arquitetura e os detalhes técnicos da automação n8n com o CRM Kommo. Também criei um arquivo .md para documentar os exemplos em código de forma mais específica e completa.

1. Arquitetura do Workflow no n8n

- Utilização preferencial de Webhook.
- Fallback com Cron a cada 20 minutos (caso o Webhook falhe).
- Alternativamente, uso direto de Cron com intervalo de 15 a 30 minutos.

2. Sequência dos Nós no n8n

Webhook / Cron → Function Node (validação do payload) → HTTP Request (busca de lead no Kommo) → IF Node (validação de pagamento) → HTTP Request (mudança de estágio do lead) → Error Trigger (tratamento de falhas)

3. Autenticação no Kommo

- Método preferido: API-Key, o motivo é: API-key é suficiente para server-to-server sem interação humana.
- Alternativa: OAuth 2.0 (quando necessário acesso contínuo e renovável).

Armazenamento seguro:

- As credenciais são salvas nas configurações do n8n (criptografadas).

Token Refresh (caso OAuth):

- Renovação via Refresh Token, usando o nó OAuth2 ou função customizada.

4. Chamadas à API do Kommo

Endpoints utilizados:

- GET /api/v4/leads (Filtro: scoring >= 80 ou status pagamento_aprovado).
- PATCH /api/v4/leads/{id} (Atualiza estágio do lead).

Exemplo de payload para PATCH:

```
{
  "status_id": 12345,
  "pipeline_id": 67890
}
```

5. Paginação

- Utilizar parâmetros limit=50&page=1.
- Repetir chamadas enquanto _links.next for diferente de null.
- Loop implementado com Interval Node.

6. Rate Limit

- Resposta HTTP 429 Too Many Requests: aplicar backoff exponencial (2s → 4s → 8s), usando Wait Node.

7. Mapeamento de Dados

- Campos obrigatórios:
 - lead_id do webhook → usado no PATCH.
 - novo_status_id com base em regras de negócio.
- Campos opcionais:
 - notas: ex: "Lead movido por automação em {data}".

8. Tratamento de Erros

- Retry automático: Até 3 tentativas usando Error Trigger.
- Fallback: Após falha, registrar em Dead Letter Queue (Google Sheets, Redis, etc).
- Notificações: Slack ou Email com detalhes.

9. Segurança e Conformidade

- Sanitização de logs com Function Node (ex: console.log(lead_id)).
- Criptografia: SSL em todas as chamadas e uso de variáveis de ambiente.

10. Observabilidade

- Métricas: sucesso, tempo_execucao_ms, lead_id.
- Dashboard: uso do Grafana com logs JSON.

11. Teste e Validação

- Ambiente Sandbox: pipelines de teste no Kommo.
- Geração de leads de teste com scoring variado.
- Testes: happy path, lead inexistente, API-key inválida.

12. Extensibilidade

Função reutilizável para mover entidades:

```
const moverEntidade = (tipo, id, novoStatus) => {  
  const endpoints = {  
    'lead': '/api/v4/leads',  
    'negocio': '/api/v4/negocios',  
    'tarefa': '/api/v4/tasks'  
  };  
  return { endpoint: endpoints[tipo], body: { status_id: novoStatus } };  
};
```