

n8n Workflow Prompt Framework & Guidelines

Core Prompt Structure

1. Context Setting (Always Include)

Unset

"Create an n8n workflow that [primary objective]. This workflow should be production-ready with proper error handling and documentation."

2. Trigger Definition (Required)

Specify ONE of these trigger types clearly:

- **Manual Trigger:** "Manually triggered workflow for..."
- **Webhook:** "When receiving a webhook from [source]..."
- **Schedule:** "Run every [frequency] to..."
- **App Trigger:** "When [specific event] occurs in [app]..."
- **File/Email:** "When new [file/email] is received..."

3. Data Flow Specification (Critical)

Unset

"The workflow should:

1. [Input step] - describe data source and expected format
2. [Processing step] - describe transformations, conditions, loops
3. [Output step] - describe destination and required format
4. [Error handling] - describe fallback behavior"



Essential Components Checklist

Trigger Details

- ☐ **Trigger Type:** Clearly specify (webhook, schedule, manual, etc.)
- ☐ **Trigger Configuration:** Include specific settings (frequency, endpoints, etc.)
- ☐ **Expected Input Format:** Define data structure/schema

Data Processing

- ☐ **Field Mapping:** Specify which input fields map to which outputs
- ☐ **Data Transformations:** Detail any formatting, calculations, or modifications
- ☐ **Conditional Logic:** Define if/then scenarios with specific conditions
- ☐ **Loops/Iterations:** Specify when to process arrays or repeat actions

Integrations & Actions

- ☐ **Target Applications:** Name specific services (Discord, Airtable, etc.)
- ☐ **Required Operations:** Be specific (create record, send message, update row)
- ☐ **Authentication Needs:** Mention credential requirements
- ☐ **Data Format Requirements:** Specify expected formats for each integration

Error Handling & Quality

- ☐ **Error Scenarios:** Define what should happen when things fail
- ☐ **Retry Logic:** Specify retry attempts and conditions
- ☐ **Logging Requirements:** Define what should be logged and where
- ☐ **Validation Rules:** Specify data quality checks

Template Structures

Simple Linear Workflow

Unset

```
"Create a workflow that:  
- Triggers: [specific trigger]  
- Processes: [data transformation]  
- Outputs: [specific destination]  
- Handles errors by: [fallback action]"
```

Complex Multi-Path Workflow

Unset

```
"Create a workflow that:
- Triggers: [trigger details]
- Checks if: [condition]
  - If true: [path A actions]
  - If false: [path B actions]
- For each item: [loop logic]
- Finally: [completion action]
- Error handling: [fallback strategy]"
```

Integration-Heavy Workflow

Unset

```
"Create a workflow that connects:
- Source: [app] via [method] expecting [data format]
- Processing: [transformation rules]
- Destinations:
  - [App 1]: [specific action and data format]
  - [App 2]: [specific action and data format]
- Include proper authentication placeholders and error handling"
```

⚠ Common Pitfalls to Avoid

Vague Descriptions

❌ **Don't say:** "Process some data and send it somewhere" ✅ **Do say:** "Extract 'name', 'email', and 'phone' fields from webhook payload, validate email format, and create contact in HubSpot"

Missing Error Context

❌ **Don't say:** "Handle errors" ✅ **Do say:** "If API call fails, retry 3 times with 5-second delays, then log error to Google Sheets and send alert to Slack #alerts channel"

Unclear Data Flow

❌ **Don't say:** "Get data from Google Sheets" ✅ **Do say:** "Monitor Google Sheet 'Leads' for new rows, extract columns A-E (name, email, phone, company, status), filter for status='new'"

Advanced Prompt Modifiers

For Complex Logic

Unset

```
"Include branching logic that:  
- Evaluates [specific condition]  
- Routes to different paths based on [criteria]  
- Merges results using [merge strategy]"
```

For Data Transformation

Unset

```
"Transform data by:  
- Converting [field] from [format A] to [format B]  
- Calculating [formula] using [input fields]  
- Filtering records where [condition]  
- Grouping by [field] and aggregating [metrics]"
```

For Enterprise Features

Unset

```
"Include production-ready features:  
- Comprehensive error handling with retry logic  
- Detailed logging for audit trails  
- Data validation and sanitization  
- Rate limiting considerations  
- Credential management placeholders"
```

Example High-Quality Prompts

Example 1: E-commerce Order Processing

Unset

```
"Create an n8n workflow that processes new e-commerce orders:"
```

TRIGGER: Webhook from Shopify when new order is placed

INPUT: Order JSON with customer info, items, totals

PROCESSING:

1. Validate order data (required fields present, valid email format)
2. Check if customer exists in CRM (HubSpot)
3. If new customer: create contact record
4. Calculate shipping cost based on weight and location
5. Check inventory levels for each item

OUTPUTS:

- Create order record in Airtable with all details
- Send order confirmation email via SendGrid
- Post to Slack #orders channel with summary
- If low inventory: alert warehouse team via email

ERROR HANDLING:

- Invalid data: log error and send to dead letter queue
- API failures: retry 3x with exponential backoff
- Critical errors: alert operations team immediately

Include proper field mappings, authentication placeholders, and documentation sticky notes."

Example 2: Social Media Monitoring

Unset

"Create an n8n workflow for social media brand monitoring:

TRIGGER: Schedule (every 15 minutes)

PROCESSING:

1. Search Twitter API for mentions of brand keywords:
['@ourcompany', 'our product', '#ourhashtag']
2. Filter out retweets and replies older than 1 hour
3. Analyze sentiment using built-in sentiment analysis
4. Categorize mentions: support request, complaint, praise, general

ROUTING LOGIC:

- Negative sentiment + support keywords → create Zendesk ticket
- High engagement (>100 likes/retweets) → notify marketing team
- Influencer mentions (>10k followers) → add to influencer CRM
- All mentions → log to Google Sheets dashboard

OUTPUTS:

- Zendesk: Create ticket with tweet content, user info, sentiment score
- Slack: Post high-priority mentions to #social-alerts
- Google Sheets: Log all mentions with timestamp, sentiment, category
- Email: Daily digest to marketing team

Include rate limiting, duplicate detection, and comprehensive error handling."

Quality Indicators

A well-crafted prompt should result in workflows that include:

- Specific node types mentioned by name
- Proper expression syntax for data references
- Error handling nodes (IF, Try/Catch, etc.)
- Sticky notes with clear documentation
- Realistic placeholder values
- Proper connection mapping
- Authentication considerations
- Production-ready structure



Pro Tips

1. **Be Specific About Data Formats:** Always specify expected JSON structure, field names, and data types
2. **Define Success Criteria:** Clearly state what constitutes successful completion
3. **Include Edge Cases:** Mention scenarios like empty responses, API limits, or missing data
4. **Specify Timing:** Include details about delays, timeouts, and scheduling requirements
5. **Reference Real Tools:** Use actual application names and their specific API capabilities
6. **Think Production:** Consider monitoring, logging, and maintenance requirements

This framework ensures your prompts generate robust, importable n8n workflows that work in real-world scenarios.