

EspoCRM with N8N integration

Link: <https://github.com/traien/n8n-nodes-espocrm>

Alt: <https://www.npmjs.com/package/@traien/n8n-nodes-espocrm>

Em 24/08/20206

@traien/n8n-nodes-espocrm

[@traien/n8n-nodes-espocrm](#)

This is a community node for [n8n](#) that enables seamless integration with [EspoCRM](#), a powerful open-source CRM platform. With this node, you can automate your CRM operations and integrate EspoCRM with other services in your n8n workflows.

[n8n](#) is a [fair-code licensed](#) workflow automation platform.

Features

The EspoCRM node provides comprehensive access to EspoCRM's API functionality:

- **Entity Operations**
 - Create new records (Leads, Contacts, Accounts, etc.)
 - Retrieve entity details
 - Update records (full or partial updates)
 - Delete records
 - List/search entities with advanced filtering
 - Supported core entities: Account, Contact, Lead, Meeting, Task, Call, Opportunity, Case
- **Dynamic Entity Support**
 - Work with any entity type in your EspoCRM instance
 - Automatic field detection and validation
 - Custom field support

- **Advanced Filtering**

- Complex search queries
- Date-based filtering
- Relationship filters
- Array field operations
- Custom boolean filters

- **Performance Optimizations**

- Pagination support
- Selective field loading
- Skip total count for large datasets
- Skip duplicate verification checks on record creation (using `X-Skip-Duplicate-Check` header)

- **AI Agent Tooling**

- Dedicated `EspoCRM Tool` node that outputs an AI Tool connection for n8n Agents
- Configure which entities and operations the agent may call plus safe pagination defaults
- Agent requests are translated into authenticated EspoCRM REST calls automatically

Prerequisites

- Active EspoCRM instance (v6.0 or later recommended)
- API access enabled in EspoCRM
- API user with appropriate permissions
- n8n installation (v0.170.0 or later)

Installation

Follow these steps to install this package in your n8n instance:

1. Open your n8n instance
2. Go to Settings > Community Nodes
3. Select "Install"
4. Enter `@traien/n8n-nodes-espocrm`
5. Click "Install"

For a manual installation, you can use:

```
npm install @traien/n8n-nodes-espocrm
```

Or if you have installed n8n globally:

```
npm install -g @traien/n8n-nodes-espocrm
```

Configuration

API Authentication

1. In EspoCRM:
 - Navigate to Administration > API Users
 - Create a new API User
 - Generate an API Key
 - Note down the API Key and API Secret
2. In n8n:
 - Add new credentials of type 'EspoCRM API'
 - Enter your EspoCRM instance URL
 - Input the API Key and API Secret
 - Save the credentials

Usage Examples

Creating a Contact

Create a new contact record with customized field values:

1. Add an "EspoCRM" node to your workflow
2. Select "Contact" as the Resource
3. Choose "Create" as the Operation
4. Fill in the required fields:
 - First Name
 - Last Name
 - Email Address
5. Add any additional fields as needed
6. (Optional) Under **Options**, enable **Skip Duplicate Check** if you wish to bypass EspoCRM's duplicate verification rules
7. Connect to other nodes in your workflow

Creating a Meeting

1. Add an "EspoCRM" node to your workflow
2. Select `Meeting` as the Resource
3. Choose `Create` as the Operation
4. Fill in the required fields:
 - `Name`
 - `Start Date`
5. Optionally set `End Date`, `Status`, `Assigned User ID`, or parent (`parentType` + `parentId`)
6. Execute to create the meeting in EspoCRM

Working with Dynamic Entities

The Dynamic resource allows you to work with any entity type in your EspoCRM system:

1. Add an "EspoCRM" node to your workflow
2. Select "Dynamic" as the Resource
3. Choose your desired Operation (Create, Update, Get, etc.)
4. Select the Entity Type from the dropdown
5. The available fields will be automatically loaded based on the entity type
6. Complete the required fields and connect to your workflow

Advanced Filtering Example

```
{
  "operation": "getAll",
  "entityType": "Lead",
  "filterOptions": {
    "where": [
      {
        "type": "and",
        "value": [
          {
            "type": "equals",
            "field": "status",
            "value": "New"
          },
          {
            "type": "greaterThan",
            "field": "createdAt",
            "value": "2024-01-01"
          }
        ]
      }
    ]
  }
}
```

```
    }
  ],
  "orderBy": "createdAt",
  "order": "desc",
  "maxSize": 50
}
}
```

Using EspoCRM as an AI Agent Tool

The package now ships with an `EspoCRM Tool` node that emits an AI Tool connection compatible with n8n's Agent/LLM nodes. Use it to let an agent autonomously read and modify EspoCRM data:

1. Drop the **EspoCRM Tool** node into the same workflow as your Agent node and select your Espo credentials.
2. Provide a concise tool description plus the entity types/operations you want to expose (e.g. `Account,Contact,Opportunity`).
3. Connect the node to the Agent's **Tools** input. At runtime the agent can call EspoCRM by sending structured parameters that map to the API.

The tool expects JSON with the following shape:

```
{
  "entityType": "Contact",
  "operation": "getAll",
  "filters": {
    "where": [
      { "type": "equals", "field": "accountId", "value": "ACCOUNT_ID" }
    ],
    "orderBy": "createdAt",
    "order": "desc"
  },
  "limit": 25
}
```

Supported operations are `get`, `getAll`, `create`, `update`, and `delete`. For single-record actions include `recordId`; for creates/updates add a `data` object with Espo field names. List requests inherit the default/max limits you configure on the node, and you can set `returnAll: true` when the agent needs the full dataset.



i When invoking the tool from an AI Agent, `data` and `filters` can be provided either as native JSON objects or as JSON strings (for example `"{\\"name\\":\\"Acme\\"}"`). The node automatically parses strings into objects before sending the EspoCRM REST call. You can also override the requested operation by including `operation` or `operations` in the payload (for example `{ "operation": "getAll" }` or `{ "operations": ["get", "update"] }`); requested values are validated against the operations enabled on the node. When no override is provided, the operations selected in the node panel act as defaults.

Attachments & Documents

You can upload files to EspoCRM as `Attachment` records and then create a `Document` that references the uploaded file. You can also download attachments to binary output in n8n.

Upload Methods

The Attachment upload operation supports two input sources:

Option A: Binary Field (default)

Upload from a binary property on the input item. Use this when you have binary data from a previous node (e.g., HTTP Request, Read Binary File, Email trigger).

- Add an `EspoCRM` node
- Resource: `Attachment`
- Operation: `Upload`
- Input Source: `Binary Field`
- Fields:
 - `Binary Property`: name of the binary key on the incoming item (default `data`)
 - `Related Type`: usually `Document` (but can be any entity supported by your instance)
 - `Field`: usually `file` for Document's File field
 - `Role`: defaults to `Attachment` (other roles: Inline Attachment)
- The node derives `name`, `type`, and `size` from the binary; you can override `name` and `type` in Additional Fields
- Output contains the created Attachment `id`

Option B: Base64 Direct

Provide base64-encoded file content directly. Use this when you have base64 data from an API response, AI agent, or other source.

- Add an `EspoCRM` node
- Resource: `Attachment`

- Operation: `Upload`
- Input Source: `Base64 Direct`
- Fields:
 - `Base64 Data`: the base64-encoded file content (without data URI prefix)
 - `File Name`: file name with extension (e.g., `document.pdf`)
 - `MIME Type`: optional, auto-detected from file extension if not provided
 - `Related Type`: usually `Document`
 - `Field`: usually `file`
 - `Role`: defaults to `Attachment`
- Output contains the created Attachment `id`

Create a Document linked to the uploaded file

- Add a second `EspoCRM` node
- Resource: `Document`
- Operation: `Create`
- Fields:
 - `Name`: document name
 - `File ID`: reference the Attachment `id` (from the previous step)
 - Optional: `Publish Date` (date only is expected; we normalize inputs), `Status`, `File Name`, `Folder ID`, `Description`, `Assigned User ID`

Notes:

- EspoCRM may restrict allowed file types by extension/MIME. If you receive `403 Not allowed file type`, verify your instance settings and the file's extension/MIME.
- For attachment-multiple fields (e.g., `Note.attachments`), upload first, then create/update the parent entity with `attachmentsIds` including the returned attachment ID (remember to include existing IDs when updating to avoid unlinking).

Resources

- [EspoCRM API Documentation](#)
- [n8n Community Nodes Documentation](#)

Support

- For EspoCRM-specific issues: [EspoCRM Forum](#)
- For node-specific issues: Create an issue in the GitHub repository

- For n8n-related questions: [n8n Community](#)

Development

Local Development on macOS (npm)

For local development on macOS with n8n installed globally via npm:

1. **Install n8n globally** (if not already installed):

```
npm install -g n8n
```

2. **Clone and build the package:**

```
git clone https://github.com/traien/n8n-nodes-espocrm.git
cd n8n-nodes-espocrm
npm install
npm run build
```

3. **Link the package to n8n's custom nodes folder:**

```
# Create package.json in n8n's nodes folder if it doesn't exist
mkdir -p ~/.n8n/nodes
cd ~/.n8n/nodes
echo '{"name": "installed-nodes", "private": true}' > package.json

# Install the local package (creates a symlink)
npm install /path/to/your/n8n-espocrm
```

4. **Start n8n:**

```
n8n start
```

Access n8n at <http://localhost:5678>

5. **Development workflow:**

```
# After making changes to the source code:
cd /path/to/your/n8n-espocrm
npm run build

# Restart n8n to pick up changes (Ctrl+C to stop, then):
```

```
n8n start
```

6. **Watch mode** (optional - auto-rebuild on changes):

```
npm run dev
```

Note: You still need to restart n8n after rebuilds.

Docker Development

To run a local version of n8n with this node for development using Docker. Make sure to replace `/path/to/your/local/n8n-espocrm` with the absolute path to this repository on your machine.

```
docker run -it --rm \
  --name n8n \
  -p 5678:5678 \
  -e N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true \
  -e N8N_RUNNERS_ENABLED=true \
  -e N8N_LOG_LEVEL=debug \
  -e N8N_COMMUNITY_PACKAGES_ALLOW_TOOL_USAGE=true \
  -v n8n_data:/home/node/.n8n \
  -v /path/to/your/local/n8n-espocrm:/home/node/.n8n/custom/n8n-espocrm \
  n8nio/n8n
```

Useful Commands

Command	Description
<code>npm run build</code>	Compile TypeScript and copy icons
<code>npm run dev</code>	Watch mode for active development
<code>npm run lintfix</code>	Auto-fix ESLint issues
<code>npm run lint</code>	Check for linting errors

License

[MIT](#)

Revision #2

Created 2026-08-24 09:53:00 UTC by Administrador

