# RBAC-matcher Developer Runbook

## Introduction

RBAC-matcher is a Python-based tool for generating and analyzing organizational charts from Excel or CSV data. It enables natural language file selection, fuzzy matching, column-based filtering, and outputs interactive HTML org charts. The project utilizes FastMCP for agent tooling and can be run locally or via Docker.

## Prerequisites

* Python: 3.8+
* Docker & Docker Compose (recommended for deployment)
* Excel or CSV files with standard headers located in /data

## Environment Setup

Clone the repository:

git clone <your-repo-url>  
cd RBAC-matcher

Install dependencies using uv or pip:

uv sync  
# or, alternatively  
pip install -r requirements.txt

Set up your .env file:

* Copy .env.example to .env
* Fill in all required API keys and endpoints

## Running the Server

Method A: Python

uv run server.py  
# Or  
python server.py

The server listens on port 3000 by default.

Method B: Docker

docker compose build server  
docker compose up -d server

Your Excel or CSV files should be in ./data. Generated org charts are saved to ./output.

## Running the Client

uv run client.py  
# Or  
python client.py

When prompted, enter your natural language instruction (e.g., "Show the org chart for cost center Woodland"). Type exit to quit.

## File & Data Layout

* /data: Place all Excel or CSV files here. Must contain headers like:  
  Associate ID, Legal Last Name, Legal First Name, Reports To Legal Name, Reports To Manager ID, Cost Center ID, Cost Center Name, Job Title, Worker Type
* /output: Generated HTML org charts go here.

## Main Features & Tools

1. Org Chart Generation (xlsx\_to\_org\_chart)

* Selects files based on name or description, even with typos or partial matches.
* Generates interactive org charts.
* Filtering: Restrict by column using, for example:  
  filter="Cost Center Name=Woodland,Job Title=Manager"
* Chart opens automatically in your web browser.

2. Cost Center Discovery (list\_cost\_centers)

Lists all unique cost centers in a selected file:

list\_cost\_centers(file\_reference=

3. Fuzzy Matching

Provides suggestions if filename is incomplete or misspelled.

Did you mean 'AcmeStaff.xlsx'? Reply with proceed=yes to continue.

## Common Developer Tasks

* Add new data: Place your Excel or CSV file in /data.
* View available cost centers: Use the list\_cost\_centers tool.
* Generate a filtered org chart: Use xlsx\_to\_org\_chart with a filter string.
* Customize node appearance: Edit the save\_network\_html() function in server.py.
* Add more tools: Define additional @mcp.tool-decorated functions in server.py.

## Debugging

View logs:

docker compose logs -f server

Inspect data/output mounts:

docker compose exec server bash -lc 'ls -la /data && echo && ls -la /app/output'

## Maintenance & Tips

* Restart the server after modifying tools or dependencies.
* Store all credentials in the .env file.
* For production, refer to the deployment steps in README.Docker.md.
* Core functions for file selection and charting are in server.py.
* Adjust AI interaction rules in /personas/default as needed for prompt alignment.

## Support & Further Reading

* Read code comments in server.py for technical details.
* For customizing org chart visuals, see pyvis documentation.
* Consult Docker and FastMCP documentation as needed.

Welcome aboard! This runbook should get you running and productive with RBAC-matcher. For more details, please refer to the project’s README and in-file comments.