graphrag / **graphrag-chat**

Code Issues Pull requests Actions Projects Security Insights

3 stars 2 forks 0 watching Branches Activity Custom properties
Tags

Public repository

1 Branch 0 Tags Go to file t Go to file Add file Code

akollegger added general notes about the project to readme 19dd8c5 · last month

graphrag_chat	converted in_memory to a singleton....	last month
.env.template	initial draft of a mediated multi-agen...	last month
.gitignore	initial draft of a mediated multi-agen...	last month
.python-version	initial draft of a mediated multi-agen...	last month
.tool-versions	initial draft of a mediated multi-agen...	last month
README.md	added general notes about the proje...	last month
main.py	initial draft of a mediated multi-agen...	last month
neo4j.example.json	initial draft of a mediated multi-agen...	last month
pyproject.toml	initial draft of a mediated multi-agen...	last month
uv.lock	initial draft of a mediated multi-agen...	last month

README

GraphRAG Chat

Moderated multi-agent group chat.

- top-level agent routes questions to sub-agents by name or speciality
- top-level agent includes multi-memory delegation
- all memories saved to each memory implementation
- sub-agents can select which memory implementation to read from
- a Neo4j graph catalog is available for read/write to multiple knowledge graphs

Agents:

```
graphrag_chat_agent_v1 (root agent)
│
├── cypher_agent: direct read/write access to available knowledge graphs
│
├── agent_smith: Neo4j product specialist
│   └── (memory: in-memory service. TBD migrate to simple KG memory)
│
└── new_agent: Give them a fun name and topical expertise
      └── (memory: note the memory impl)
```

Developer Notes

1. Initialize the Python environment with `uv`

- Create a virtual environment:

```
uv venv
```

- Activate the virtual environment:

```
source .venv/bin/activate  # select the appropriate activate.* for your shell
```

- Install dependencies:

```
uv sync
```

2. Set up configuration files

- Copy the environment template and edit as needed:

```
cp .env.template .env
# Edit .env to set your environment variables
```

- Copy the Neo4j connections example and edit as needed:

```
cp neo4j.example.json neo4j.json
# Edit neo4j.json to configure your Neo4j connections
```

3. Run the agent using `adk`

- Start the agent web server:

```
adk web
```

You should now be able to access the agent locally. See below for more details or troubleshooting steps.