

Non-Markdown File Support

Overview

Add support for tracking non-markdown files in the knowledge graph. This provides a foundation for features like file analysis while immediately enabling useful capabilities like file tracking and linking.

Core Changes

File Change Scanner

- Remove markdown-only restriction in `_detect_mime_type`
- Return actual mime type for any file type
- Keep existing change detection logic

Sync Service

```
async def sync(self):
    async for change in self.scanner.find_changes():
        if change.type == "added" or change.type == "modified":
            if change.content_type == "text/markdown":
                # Existing markdown parsing flow
                await self.process_markdown(change.file_path)
            else:
                # Simple file entity creation
                await self.entity_service.create_entity(
                    file_path=change.file_path,
                    content_type=change.content_type,
                    checksum=change.checksum
                )
```

Read File Tool

Add a new tool that can handle any file type:

```
async def read_file(self, file_path: str) -> dict:
    entity = await self.entity_service.get_by_file_path(file_path)
    content_type = entity.content_type

    if content_type == "text/markdown":
        content = await self.file_service.read_file(file_path)
        return {
            "type": "text/markdown",
            "content": content
        }

    # Non-markdown files return as base64
    binary = await window.fs.readFile(file_path)
    return {
        "type": content_type,
```

```
"content": base64.b64encode(binary).decode('utf-8'),  
"encoding": "base64"  
}
```

Benefits

1. All files tracked in knowledge graph
2. Files can be targets of relations
3. Track moves and renames
4. Everything stays in git
5. Foundation for future analysis features

Implementation Steps

1. Update file change scanner
2. Add non-markdown entity creation
3. Add read_file tool
4. Update tests
5. Document new capabilities

Test Cases

1. Basic file tracking:
 - Add non-markdown file
 - Verify entity created
 - Check content type stored
2. File operations:
 - Move file
 - Rename file
 - Delete file
 - Verify entity updates
3. Read tool:
 - Read markdown
 - Read image
 - Read binary file
 - Verify correct content types
4. Relations:
 - Create relation to file
 - Move file
 - Verify relation updates

Future Capabilities

This change enables future features like:

- AI file analysis
- Batch processing
- Advanced search

- Content extraction

But keeps the current implementation simple and focused.

Observations

- [tech] Store non-markdown files in entity table [#implementation](#)
- [design] Skip markdown parsing for non-markdown files [#core](#)
- [feature] Enable linking to any file type [#usability](#)
- [principle] Keep initial implementation simple [#architecture](#)

Relations

- extends [sync/file-change-scanner](#)
- implements [models/knowledge](#)
- related_to [design/file-analysis-and-batch-processing-design](#)