Selected files

`1 printable files`

`MEMORY_SYNC_FIX_SUMMARY.md`

`MEMORY_SYNC_FIX_SUMMARY.md`

Memory Sync Fix Summary

Problem Description

You experienced a **memory agent disconnect** where:

- **Streamlit interface** showed 20 memories using `agno_memory.get_user_memories()`
- **Agent tools** showed 22 memories using `memory_manager.search_memories()`
- **2 new memories** were accepted and stored but not visible in Streamlit interface
- **Inconsistent memory access** between different parts of the system

Root Cause Analysis

The issue was caused by **inconsistent memory access interfaces**:

1. **StreamlitMemoryHelper.get_all_memories()** used:

   ```
   return self.agent.agno_memory.get_user_memories(user_id=self.agent.user_id)
   ```

2. **Agent memory tools** used:

   ```
   results = self.agno_memory.memory_manager.search_memories(
       query="", db=self.agno_memory.db, user_id=self.user_id,
       limit=None, similarity_threshold=0.0, search_topics=False
   )
   ```

Solution Implemented

1. Fixed StreamlitMemoryHelper Interface

File: `tools/streamlit_helpers.py`

Change: Updated `get_all_memories()` to use the same SemanticMemoryManager interface as agent tools:

```
def get_all_memories(self):
    """Get all memories using consistent SemanticMemoryManager interface"""
    if not self.memory_manager or not self.db:
        return []
    try:
        # Use same method as agent tools for consistency
        results = self.memory_manager.search_memories(
            query="",  # Empty query to get all memories
            db=self.db,
            user_id=self.agent.user_id,
            limit=None,  # Get all memories
            similarity_threshold=0.0,  # Very low threshold to get all
            search_topics=False,
        )
        # Extract just the memory objects from the (memory, score) tuples
        return [memory for memory, score in results]
    except Exception as e:
        st.error(f"Error getting all memories: {e}")
        return []
```

2. Added Memory Sync Monitoring

File: `tools/streamlit_helpers.py`

Added Methods:

- `sync_memory_to_graph()` - Sync individual memories to LightRAG
- `get_memory_sync_status()` - Check sync between local SQLite and LightRAG graph

3. Enhanced Streamlit Interface

File: `tools/paga_streamlit_agno.py`

Added Section: Memory Sync Status monitoring in the Memory Manager tab:

- Check sync status between local and graph systems
- Display memory counts from both systems
- Sync missing memories button
- Visual indicators for sync health

4. Improved Dual Storage Logging

File: `src/personal_agent/core/agno_agent.py`

Enhanced: `store_user_memory()` method with better sync status logging:

- Detailed logging of dual storage success/failure
- Better error handling for graph sync failures

- Clear status indicators in return messages

Key Features Added

1. Consistent Memory Access

- Both Streamlit and Agent tools now use the same underlying memory access method
- Eliminates interface mismatches that caused different memory counts

2. Memory Sync Monitoring

- Real-time sync status between local SQLite and LightRAG graph systems
- Visual indicators in Streamlit interface
- Automatic sync repair functionality

3. Enhanced Error Handling

- Graceful handling of graph sync failures
- Detailed logging for debugging sync issues
- Fallback mechanisms when graph system is unavailable

4. Dual Storage Architecture

- Maintains storage in both local SQLite (fast access) and LightRAG graph (relationships)
- Automatic synchronization between systems
- Knowledge graph building capabilities preserved

Testing

Created `test_memory_sync_fix.py` to verify:

1. Memory storage via agent's dual storage system
2. Memory retrieval consistency between interfaces
3. Sync status monitoring functionality
4. Error handling and fallback mechanisms

Usage Instructions

For Users:

1. **Check Sync Status:** Use the "🔍 Check Sync Status" button in Memory Manager tab
2. **Sync Missing Memories:** Use "🔄 Sync Missing Memories" if systems are out of sync

3. **Monitor Dual Storage:** Look for "✅ Local memory" and "✅ Graph memory" indicators when storing facts

For Developers:

1. **Consistent Interface:** Always use `memory_manager.search_memories()` for memory access
2. **Dual Storage:** Use `agent.store_user_memory()` for automatic dual storage
3. **Sync Monitoring:** Use `StreamlitMemoryHelper.get_memory_sync_status()` for sync checks

Files Modified

1. **`tools/streamlit_helpers.py`** - Fixed memory access interface consistency
2. **`tools/paga_streamlit_agno.py`** - Added sync monitoring UI
3. **`src/personal_agent/core/agno_agent.py`** - Enhanced dual storage logging
4. **`test_memory_sync_fix.py`** - Created test script (new file)
5. **`MEMORY_SYNC_FIX_SUMMARY.md`** - This documentation (new file)

Expected Results

After implementing these fixes:

- ✅ **Streamlit interface** and **agent tools** show the same memory count
- ✅ **New memories** are visible in both interfaces immediately
- ✅ **Dual storage** works consistently (local SQLite + LightRAG graph)
- ✅ **Sync monitoring** provides visibility into system health
- ✅ **Knowledge graphs** continue building with proper entity relationships

Verification Steps

1. Run `python test_memory_sync_fix.py` to verify the fix
2. Use Streamlit interface to add new memories
3. Check that both "📋 Load All Memories" and agent tools show same count
4. Monitor sync status in Memory Manager tab
5. Verify dual storage success messages when adding memories

The memory disconnect issue should now be resolved with consistent access across all interfaces while maintaining your knowledge graph building capabilities.