API Fixes Implementation Summary

Date: 2025-11-13

All Priority 1 and Priority 2 fixes have been successfully implemented in app.py.

✅ FIXES COMPLETED

Priority 1: Critical Parser Fixes (100% Complete)

1. MediaAgents Parser Fixed ✅

File: app.py:380-424

Problem: API returns `{"response": [{"entityInfo": {...}}]}` but parser expected `{"mediaAgentList": [...]}`

Solution Implemented:

```
# Now checks for 'response' key first with entityInfo structure
ma_list = mediaagents_json.get("response", [])

# Then extracts ID and name from entityInfo
if "entityInfo" in ma_entry:
    entity_info = ma_entry.get("entityInfo", {})
    ma_id = entity_info.get("id")
    name = entity_info.get("name", "")
```

Backward Compatibility: Falls back to old structure if new one not found

Expected Result: MediaAgents will now be properly stored in database

---

2. Libraries Parser Fixed ✅

File: app.py:426-466

Problem: Same as MediaAgents - wrong JSON structure expected

Solution Implemented:

```
# Now checks for 'response' key first with entityInfo structure
lib_list = libraries_json.get("response", [])

# Then extracts ID and name from entityInfo
if "entityInfo" in lib_entry:
    entity_info = lib_entry.get("entityInfo", {})
    lib_id = entity_info.get("id")
    name = entity_info.get("name", "")
```

Expected Result: Libraries will now be properly stored in database

---

3. Storage Pools Fixed (Endpoint + Parser) ✅

File: - Endpoint: app.py:787-795 - Parser: app.py:468-503

Problem 1: Endpoint `/V4/StoragePool` returns 404 (V4 API not available)

Solution 1:

```
# Changed from:
response = requests.get(f"{base_url}/V4/StoragePool", headers=headers)
```

```
# To:
response = requests.get(f"{base_url}/StoragePool", headers=headers)
```

Problem 2: Parser expected `storagePools` key, API returns `storagePoolList`

Solution 2:

```
# Now uses correct key
pools_list = pools_json.get("storagePoolList", [])

# Capacity data is at pool_entry level, not inside storagePool object
total_cap = pool_entry.get("totalCapacity", "N/A")
free_space = pool_entry.get("totalFreeSpace", "N/A")  # Fixed: was 'freeSpace'
```

Expected Result: Storage pools with full capacity data will be stored

---

Priority 2: Performance Fix (100% Complete)

4. Jobs Timeout Fixed ✅

File: app.py:742-750

Problem: Request times out - API returns 1 year of jobs by default

Solution Implemented:

```
# Added time filter parameter (86400 seconds = 24 hours)
response = requests.get(
    f"{base_url}/Job?completedJobLookupTime=86400",
    headers=headers,
    timeout=30
)
```

Expected Result: Jobs from last 24 hours retrieved without timeout

---

Priority 3: Endpoint Research (Implemented with Fallback)

5. Events Endpoint Updated ⚠️

File: app.py:817-833

Problem: `/Event` returns 404

Solution Implemented:

```
# Try new endpoint first
response = requests.get(f"{base_url}/CommServ/Event?level=Critical", headers=headers)

# Fallback to old endpoint if new one fails
if response.status_code != 200:
    response = requests.get(f"{base_url}/Event?level=Critical", headers=headers)
```

Status: Needs testing - may not be available in this Commvault version

---

📊 Implementation Statistics

Fix	Priority	Status	File Location	Lines Changed
MediaAgents Parser	P1	✅ Complete	app.py:380-424	~45 lines
Libraries Parser	P1	✅ Complete	app.py:426-466	~40 lines
Storage Pools Endpoint	P1	✅ Complete	app.py:787-795	~2 lines
Storage Pools Parser	P1	✅ Complete	app.py:468-503	~36 lines
Jobs Timeout	P2	✅ Complete	app.py:742-750	~1 line
Events Endpoint	P3	⚠️ Needs Test	app.py:817-833	~17 lines

Total Lines Modified: ~141 lines **Files Modified**: 1 (app.py)

🧪 Testing

Test Script Created

File: test_fixes.py

What it tests: 1. MediaAgents endpoint + parser 2. Libraries endpoint + parser 3. Storage Pools endpoint + parser 4. Jobs with time filter 5. Events with new endpoint

How to run:

```
python test_fixes.py
```

Expected output: - Verification of each fix - JSON response structure validation - Success/failure summary - Test output files saved as `test_fixed_*.json`

📈 Expected Improvement

Before Fixes:

Endpoint	Status	Issue
MediaAgents	❌ No data	Wrong parser
Libraries	❌ No data	Wrong parser
Storage Pools	❌ 404 Error	Wrong endpoint

Endpoint	Status	Issue
Jobs	❌ Timeout	Too much data
Events	❌ 404	Wrong endpoint

Success Rate: 7/14 endpoints (50%)

After Fixes:

Endpoint	Expected Status	Fix Applied
MediaAgents	✅ Working	Parser updated
Libraries	✅ Working	Parser updated
Storage Pools	✅ Working	Endpoint + parser fixed
Jobs	✅ Working	Time filter added
Events	⚠️ Testing	Endpoint changed

Expected Success Rate: 11/14 endpoints (78%) minimum

🎯 How to Verify Fixes

Method 1: Run Test Script

```
# Test all fixes at once
python test_fixes.py
```

```
# Check output files
ls test_fixed_*.json
```

Method 2: Use the Web App

```
# Start the Flask app
python app.py

# In browser: http://localhost:5000
# 1. Select these checkboxes:
#    - MediaAgents
#    - Libraries
#    - Storage Pools
#    - Jobs
# 2. Click "Fetch Data from Commvault"
# 3. Check results - should see success counts
# 4. Go to Infrastructure Dashboard
# 5. Verify MediaAgents, Libraries, Storage Pools sections show data
```

Method 3: Check Database

```
# After fetching data, check database
python -c "
import sqlite3
db = sqlite3.connect('Database/commvault.db')
cursor = db.cursor()

print('Checking database after fixes...')
print()

cursor.execute('SELECT COUNT(*) FROM mediaagents')
count = cursor.fetchone()[0]
print(f'MediaAgents: {count} records')

cursor.execute('SELECT COUNT(*) FROM libraries')
count = cursor.fetchone()[0]
print(f'Libraries: {count} records')

cursor.execute('SELECT COUNT(*) FROM storage_pools')
```

```
count = cursor.fetchone()[0]
print(f'Storage Pools: {count} records')

cursor.execute('SELECT COUNT(*) FROM jobs')
count = cursor.fetchone()[0]
print(f'Jobs: {count} records')
"
```

Expected Results: - MediaAgents: Should show 10+ records - Libraries: Should show multiple records - Storage Pools: Should show multiple records - Jobs: Should show jobs from last 24 hours

---

📝 Code Changes Summary

1. MediaAgents Parser (app.py:380-424)

Before:

```
ma_list = mediaagents_json.get("mediaAgentList", [])
ma_info = ma_entry.get("mediaAgent", ma_entry)
ma_id = ma_info.get("mediaAgentId")
```

After:

```
ma_list = mediaagents_json.get("response", [])
if "entityInfo" in ma_entry:
    entity_info = ma_entry.get("entityInfo", {})
    ma_id = entity_info.get("id")
```

2. Libraries Parser (app.py:426-466)

Before:

```
lib_list = libraries_json.get("libraryList", [])
lib_info = lib_entry.get("library", lib_entry)
lib_id = lib_info.get("libraryId")
```

After:

```
lib_list = libraries_json.get("response", [])
if "entityInfo" in lib_entry:
    entity_info = lib_entry.get("entityInfo", {})
    lib_id = entity_info.get("id")
```

3. Storage Pools Endpoint (app.py:788)

Before:

```
response = requests.get(f"{base_url}/V4/StoragePool", headers=headers)
```

After:

```
response = requests.get(f"{base_url}/StoragePool", headers=headers)
```

4. Storage Pools Parser (app.py:468-503)

Before:

```
pools_list = pools_json.get("storagePools", [])
total_cap = pool_info.get("totalCapacity", "N/A")
free_space = pool_info.get("freeSpace", "N/A")
```

After:

```
pools_list = pools_json.get("storagePoolList", [])
total_cap = pool_entry.get("totalCapacity", "N/A")
free_space = pool_entry.get("totalFreeSpace", "N/A")
```

5. Jobs Endpoint (app.py:743)

Before:

```
response = requests.get(f"{base_url}/Job", headers=headers)
```

After:

```
response = requests.get(f"{base_url}/Job?completedJobLookupTime=86400", headers=heade
```

6. Events Endpoint (app.py:817-833)

Before:

```
response = requests.get(f"{base_url}/Event?level=Critical", headers=headers)
```

After:

```
response = requests.get(f"{base_url}/CommServ/Event?level=Critical", headers=headers)
# With fallback to old endpoint
```

🔍 Troubleshooting

If MediaAgents/Libraries still show no data:

1. Check test output file: `test_fixed_MediaAgents.json`
2. Verify JSON structure matches expected format
3. Look for `response` array with `entityInfo` objects
4. Check database: `SELECT * FROM mediaagents LIMIT 5;`

If Storage Pools still fail:

1. Verify endpoint returns 200 status
2. Check test output file: `test_fixed_Storage_Pools.json`
3. Look for `storagePoolList` key
4. Verify capacity fields: `totalCapacity`, `totalFreeSpace`

If Jobs still timeout:

1. Try increasing time window: `completedJobLookupTime=172800` (48 hours)
2. Or decrease: `completedJobLookupTime=43200` (12 hours)
3. Check if any jobs exist in that time range

If Events still fail:

1. Check if `/CommServ/Event` endpoint exists in your version
2. Try without filters: `/CommServ/Event`
3. May not be available in all Commvault versions
4. Alternative: Use Alerts endpoint instead

📚 Related Documentation

- API_FIXES_RESEARCH.md - Detailed research findings
- API_TEST_RESULTS.md - Original test results
- test_fixes.py - Verification test script

✅ Next Steps

1. **Test the fixes**: `bash python test_fixes.py`
2. **Verify with web app**: `bash python app.py # Visit http://localhost:5000`
3. **Check database**: `bash # Use the SQL command above to verify data`
4. **Review test output files**: `bash # Check test_fixed_*.json files for actual API responses`
5. **If all fixes work**: Update QUICK_WINS_SUMMARY.md with new success rates
6. **If Events endpoint fails**: Remove from UI or mark as "Not Available"

🎉 Success Criteria

All Priority 1 & 2 fixes are considered successful if:

- ✅ MediaAgents: Shows 10+ records in database
- ✅ Libraries: Shows multiple records in database
- ✅ Storage Pools: Shows pools with capacity data
- ✅ Jobs: Returns jobs from last 24 hours without timeout
- ⚠️ Events: Works (bonus) or gracefully fails

Minimum target: 4/5 fixes working (80% success rate)

Expected: All 5 fixes working (100% success rate for Priority 1 & 2)

Implementation Complete! ✅

All confirmed fixes have been applied to app.py. Ready for testing!