

FastMCP Cloud Deployment Fixes

Issue Diagnosis

Root Cause: Missing `fastapi` dependency prevented cloud deployment

Error: `No module named 'fastapi'`

Applied Fixes

1. ☒ CRITICAL: Added FastAPI Dependency

File: `pyproject.toml`

- Added `"fastapi>=0.104.0"` to dependencies list
- This was blocking all cloud deployments

2. ☒ Fixed Async/Sync Decorator Mismatch

File: `src/koeln_presse/utils.py`

- Updated `retry_on_failure()` decorator to support both async and sync functions
- Added async wrapper with `asyncio.sleep()` for async functions
- Maintained sync wrapper with `time.sleep()` for sync functions
- Automatic detection using `asyncio.iscoroutinefunction()`

3. ☒ Fixed Dockerfile Healthcheck

File: `Dockerfile`

- Changed from `requests` (not in dependencies) to `httpx` (already installed)
- Updated healthcheck command to use `httpx`

4. ☒ Updated Documentation

File: `README.md`

- Added `fastapi` to installation instructions

Verification

Local Testing Results

- ☒ Health Check: `{"status":"healthy","service":"fastmcp-koeln-presse","version":"1.0.0"}`
- ☒ Latest Tool: Returns press items successfully
- ☒ Manifest: Returns proper MCP tool definitions
- ☒ Server Startup: No errors (except port conflict from duplicate instance)

Test Commands

```
# Health check
curl http://localhost:8000/health

# Latest press releases
curl -X POST http://localhost:8000/tools/latest -H "Content-Type:
application/json" -d '{"n":2}'

# Manifest
curl http://localhost:8000/manifest

# Search
curl -X POST http://localhost:8000/tools/search -H "Content-Type:
application/json" -d '{"query":"Köln","limit":5}'
```

Cloud Deployment Instructions

1. Push Changes to Git

```
git add .
git commit -m "Fix: Add missing fastapi dependency and async decorator support"
git push origin main
```

2. FastMCP Cloud Configuration

Build Command:

```
pip install -e .
```

Start Command:

```
python -m koeln_presse.server
```

Environment Variables (optional):

```
CACHE_TTL=300
HTTP_TIMEOUT=8
HTTP_RETRIES=3
PORT=8000
```

3. Expected Behavior

- ☒ Server should start without errors
- ☒ Health endpoint returns healthy status
- ☒ All 4 MCP tools functional:
 - `koeln.presse.latest`
 - `koeln.presse.search`
 - `koeln.presse.get`
 - `koeln.presse.categories`

Technical Details

Dependencies Now Include

- `fastapi>=0.104.0` ← **NEW (CRITICAL)**
- `fastmcp>=0.1.0`
- `httpx>=0.27.0`
- `lxml>=4.9.0`
- `pydantic>=2.0.0`
- `python-dateutil>=2.8.0`
- `uvicorn>=0.20.0`
- `typing-extensions>=4.5.0`

Async Pattern Fix

Before:

```
def retry_on_failure(...):
    def decorator(func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            # Only sync support, breaks with async
            time.sleep(wait_time)
```

After:

```
def retry_on_failure(...):
    def decorator(func):
        # Auto-detect and use appropriate wrapper
        if asyncio.iscoroutinefunction(func):
            return async_wrapper # Uses await asyncio.sleep()
        else:
            return sync_wrapper # Uses time.sleep()
```

Warnings Fixed

- Pydantic V2 deprecation warning about `schema_extra` → `json_schema_extra`
(Non-blocking, can be fixed later)

Next Steps

1. Commit and push all changes
2. Trigger new deployment in fastMCP Cloud
3. Monitor deployment logs for successful startup
4. Test all endpoints via cloud URL

Status: ☒ All critical issues resolved

Local Testing: ☒ Passed

Ready for Cloud Deploy: ☒ Yes