Contents

1   CouchDB MCP Server Setup

This document explains how to use the CouchDB MCP (Model Context Protocol) server with Claude Code.

1.1   Overview

The CouchDB MCP server provides Claude Code with direct access to your CouchDB databases, enabling: - Querying documents - Searching content - Retrieving attachments - Executing views - Listing databases and documents

1.2 Files Created

1. **couchdb_mcp_server.py** - The MCP server implementation
2. **.mcp.json** - MCP server configuration
3. **.env** - Environment variables (contains credentials - DO NOT commit!)

1.3 Configuration

1.3.1 Environment Variables

The `.env` file contains your CouchDB credentials:

```
COUCHDB_URL=http://localhost:5984
COUCHDB_USER=admin
COUCHDB_PASSWORD=SU2orange!
COUCHDB_DATABASES=skol_dev,skol_taxa_dev
```

IMPORTANT: Add `.env` to your `.gitignore` to avoid committing credentials!

1.3.2 MCP Configuration

The `.mcp.json` file configures the MCP server for Claude Code:

```
{
  "mcpServers": {
    "couchdb": {
      "type": "stdio",
      "command": "/usr/bin/python3",
      "args": ["/data/piggy/src/github.com/piggyatbaqaqi/skol/couchdb_mcp_server
      "env": {
        "COUCHDB_URL": "${COUCHDB_URL:-http://localhost:5984}",
        "COUCHDB_USER": "${COUCHDB_USER}",
        "COUCHDB_PASSWORD": "${COUCHDB_PASSWORD}",
        "COUCHDB_DATABASES": "${COUCHDB_DATABASES:-skol_dev,skol_taxa_dev}"
      }
    }
  }
}
```

1.4 Usage

1.4.1 Starting the MCP Server

The MCP server is automatically loaded when you start Claude Code in this directory.

To verify it's running, use the `/mcp` command in Claude Code:

`/mcp`

This will show all configured MCP servers and their status.

1.4.2 Available Tools

The CouchDB MCP server provides the following tools:

1.4.2.1 1. `list_databases` List all available CouchDB databases.

Example:

`List all CouchDB databases`

1.4.2.2 2. `get_document` Get a specific document by ID.

Parameters: - database - Database name (e.g., "skol_dev") - doc_id - Document ID

Example:

`Get document "doc123" from skol_dev database`

1.4.2.3 3. `query_documents` Query documents with pagination.

Parameters: - database - Database name - `limit` - Maximum documents to return (default: 100) - `skip` - Number of documents to skip (default: 0) - `include_docs` - Include full document content (default: true)

Example:

`Show me the first 10 documents from skol_taxa_dev`

1.4.2.4 4. `search_documents` Search for documents containing specific text.

Parameters: - database - Database name - search_text - Text to search for - `field` - (Optional) Specific field to search in - `limit` - Maximum results (default: 50)

Example:

`Search for "Amanita" in skol_dev database`

1.4.2.5 5. `get_attachment` Get an attachment from a document.

Parameters: - database - Database name - doc_id - Document ID - attachment_name - Attachment filename

Example:

```
Get the "species.txt" attachment from document "doc123" in skol_dev
```

1.4.2.6 6. `list_attachments` List all attachments for a document.

Parameters: - database - Database name - doc_id - Document ID

Example:

```
List all attachments for document "doc123" in skol_dev
```

1.4.2.7 7. `execute_view` Execute a CouchDB view.

Parameters: - database - Database name - design_doc - Design document name (without "_design/" prefix) - view_name - View name - key - (Optional) Filter by specific key - limit - Maximum results (default: 100)

Example:

```
Execute the "by_species" view in the "taxonomy" design doc
```

1.4.2.8 8. `count_documents` Count documents in a database.

Parameters: - database - Database name - doc_type - (Optional) Filter by document type field

Example:

```
How many documents are in skol_taxa_dev?
```

1.5 Example Interactions

1.5.1 Query Your Data

```
Show me the first 5 documents from skol_dev
```
```
Search for documents containing "mycology" in skol_taxa_dev
```
```
Get document "species_001" from skol_dev
```

1.5.2 Work with Attachments

```
List all attachments for document "doc_abc123"
```
```
Get the content of "description.txt" attachment from document "doc_abc123"
```

1.5.3 Database Information

How many documents are in each database?

List all available CouchDB databases

1.6 Resources

The MCP server exposes databases as resources with URIs:

- couchdb://skol_dev - The skol_dev database
- couchdb://skol_taxa_dev - The skol_taxa_dev database

You can reference these URIs in your interactions with Claude Code.

1.7 Troubleshooting

1.7.1 Server Not Starting

1. **Check Python path:**

   which python3

   Update the command field in .mcp.json if needed.

2. **Check MCP SDK installation:**

   pip3 show mcp

3. **Check CouchDB connection:**

   curl http://localhost:5984

1.7.2 Authentication Errors

Verify your credentials in .env: - Ensure COUCHDB_USER and COUCHDB_PASSWORD are correct - Test credentials: bash   curl -u admin:SU2orange! http://localhost:5984/_all_dbs

1.7.3 Database Not Found

Check that the database exists:

curl -u admin:SU2orange! http://localhost:5984/_all_dbs

Update COUCHDB_DATABASES in .env if needed.

1.8 Security Notes

1. **Never commit .env file** - It contains sensitive credentials
2. **Add to .gitignore:**

   .env

3. **Use environment variables** for production deployments
4. **Restrict CouchDB access** - Use firewalls and network security
5. **Use HTTPS** for remote CouchDB instances

1.9 Advanced Configuration

1.9.1 Remote CouchDB

To connect to a remote CouchDB instance, update .env:

```
COUCHDB_URL=https://your-instance.cloudant.com
COUCHDB_USER=your_username
COUCHDB_PASSWORD=your_password
```

1.9.2 Multiple Databases

Add more databases to the COUCHDB_DATABASES list (comma-separated):

```
COUCHDB_DATABASES=skol_dev,skol_taxa_dev,another_db,yet_another_db
```

1.9.3 Custom Python Environment

If using a conda environment or virtualenv, update the command in .mcp.json:

```
"command": "/home/piggy/miniconda3/envs/skol/bin/python3"
```

1.10 Development

To modify the MCP server:

1. Edit couchdb_mcp_server.py
2. Restart Claude Code to reload the server
3. Test with /mcp command

1.11 Dependencies

- Python 3.7+
- mcp - Model Context Protocol SDK
- couchdb - CouchDB Python client

Install dependencies:

```
pip3 install mcp couchdb
```

1.12 Support

For issues or questions: - Check the MCP documentation - Review Claude Code MCP guide - Check CouchDB logs for connection issues

1.13 Next Steps

1. Try the example interactions above
2. Explore your databases with natural language queries
3. Create custom views for frequently-used queries
4. Integrate with your existing PySpark workflows

Happy querying! 🚀