

MongoDB Tools Documentation

This document provides detailed information about the MongoDB tools available in the DevOps MCP Server.

Overview

The MongoDB integration allows you to interact with your MongoDB server through the MCP protocol. It provides tools for retrieving database information, collections, documents, and server status.

Configuration

The MongoDB integration can be configured using the following environment variables:

- **MONGODB_URI**: The connection URI for your MongoDB server (default: `mongodb://localhost:27017`)
- **MONGODB_TIMEOUT**: Timeout for API calls in seconds (default: `10`)

Available Tools

The MongoDB integration provides the following tools:

Database Tools

list_mongodb_databases

List all MongoDB databases.

Parameters:

- **limit** (optional): Maximum number of databases to return (default: 100, max: 500)

Example:

```
list_mongodb_databases()  
list_mongodb_databases(50)
```

get_mongodb_database_stats

Get statistics for a MongoDB database.

Parameters:

- **database** (required): Database name

Example:

```
get_mongodb_database_stats("mydb")
```

list_mongodb_collections

List all collections in a MongoDB database.

Parameters:

- **database** (required): Database name
- **limit** (optional): Maximum number of collections to return (default: 100, max: 500)

Example:

```
list_mongodb_collections("mydb")  
list_mongodb_collections("mydb", 50)
```

get_mongodb_collection_stats

Get statistics for a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name

Example:

```
get_mongodb_collection_stats("mydb", "users")
```

count_mongodb_documents

Count documents in a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name
- **filter** (optional): Filter criteria in JSON format

Example:

```
count_mongodb_documents("mydb", "users")  
count_mongodb_documents("mydb", "users", '{"active": true}')
```

Collection Tools

find_mongodb_documents

Find documents in a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name
- **filter** (optional): Filter criteria in JSON format
- **projection** (optional): Fields to include or exclude in JSON format
- **sort** (optional): Sort criteria in JSON format
- **limit** (optional): Maximum number of documents to return (default: 100, max: 500)
- **skip** (optional): Number of documents to skip (default: 0)

Example:

```
find_mongodb_documents("mydb", "users")
find_mongodb_documents("mydb", "users", '{"age": {"$gt": 18}}',
'{"name": 1, "email": 1}', '{"name": 1}', 50, 0)
```

get_mongodb_document

Get a document by ID from a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name
- **document_id** (required): Document ID

Example:

```
get_mongodb_document("mydb", "users", "507f1f77bcf86cd799439011")
```

aggregate_mongodb_documents

Run an aggregation pipeline on a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name
- **pipeline** (required): Aggregation pipeline in JSON format
- **limit** (optional): Maximum number of documents to return (default: 100, max: 500)

Example:

```
aggregate_mongodb_documents("mydb", "users", ' [{"$match": {"age": {"$gt": 18}}}, {"$group": {"_id": "$city", "count": {"$sum": 1}}}]')
```

get_mongodb_distinct_values

Get distinct values for a field in a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name
- **field** (required): Field name
- **filter** (optional): Filter criteria in JSON format
- **limit** (optional): Maximum number of values to return (default: 100, max: 500)

Example:

```
get_mongodb_distinct_values("mydb", "users", "city")  
get_mongodb_distinct_values("mydb", "users", "city", '{"active": true}')
```

get_mongodb_indexes

Get indexes for a MongoDB collection.

Parameters:

- **database** (required): Database name
- **collection** (required): Collection name

Example:

```
get_mongodb_indexes("mydb", "users")
```

Information Tools

get_mongodb_server_info

Get MongoDB server information.

Example:

```
get_mongodb_server_info()
```

get_mongodb_server_status

Get MongoDB server status.

Example:

```
get_mongodb_server_status()
```

get_mongodb_build_info

Get MongoDB build information.

Example:

```
get_mongodb_build_info()
```

get_mongodb_host_info

Get MongoDB host information.

Example:

```
get_mongodb_host_info()
```

get_mongodb_server_parameters

Get MongoDB server parameters.

Example:

```
get_mongodb_server_parameters()
```

get_mongodb_replica_set_status

Get MongoDB replica set status.

Example:

```
get_mongodb_replica_set_status()
```

get_mongodb_sharding_status

Get MongoDB sharding status.

Example:

```
get_mongodb_sharding_status()
```

get_mongodb_current_operations

Get MongoDB current operations.

Parameters:

- **limit** (optional): Maximum number of operations to return (default: 100, max: 500)

Example:

```
get_mongodb_current_operations()  
get_mongodb_current_operations(50)
```

Available Resources

The MongoDB integration provides the following resources:

- **mongodb://info**: Get MongoDB server information
- **mongodb://status**: Get MongoDB server status
- **mongodb://databases**: List all MongoDB databases
- **mongodb://db/{database}**: Get information about a MongoDB database
- **mongodb://db/{database}/collections**: List all collections in a MongoDB database
- **mongodb://db/{database}/collection/{collection}**: Get information about a MongoDB collection

PROF

Examples

Working with Databases and Collections

```
# List all databases  
list_mongodb_databases()  
  
# Get statistics for a database  
get_mongodb_database_stats("mydb")  
  
# List all collections in a database  
list_mongodb_collections("mydb")
```

```
# Get statistics for a collection
get_mongodb_collection_stats("mydb", "users")
```

Working with Documents

```
# Find documents in a collection
find_mongodb_documents("mydb", "users", '{"age": {"$gt": 18}}')

# Get a document by ID
get_mongodb_document("mydb", "users", "507f1f77bcf86cd799439011")

# Run an aggregation pipeline
aggregate_mongodb_documents("mydb", "users", '[{"$group": {"_id":
"$city", "count": {"$sum": 1}}}]')

# Get distinct values for a field
get_mongodb_distinct_values("mydb", "users", "city")
```

Working with Server Information

```
# Get server information
get_mongodb_server_info()

# Get server status
get_mongodb_server_status()

# Get current operations
get_mongodb_current_operations()
```