

PostgreSQL Tools Documentation

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

Overview

The PostgreSQL integration allows you to interact with your PostgreSQL server through the MCP protocol. It provides tools for retrieving database information, tables, executing queries, and monitoring server status.

Configuration

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

- **POSTGRES_HOST**: The host of your PostgreSQL server (default: `localhost`)
- **POSTGRES_PORT**: The port of your PostgreSQL server (default: `5432`)
- **POSTGRES_USER**: The username for your PostgreSQL server (default: `postgres`)
- **POSTGRES_PASSWORD**: The password for your PostgreSQL server (default: empty)
- **POSTGRES_TIMEOUT**: Timeout for API calls in seconds (default: `10`)

Available Tools

The PostgreSQL integration provides the following tools:

Database Tools

`list_postgresql_databases`

List all PostgreSQL databases.

Parameters:

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

Example:

```
list_postgresql_databases()  
list_postgresql_databases(50)
```

`get_postgresql_database_info`

Get information about a PostgreSQL database.

Parameters:

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

Example:

```
get_postgresql_database_info("mydb")
```

list_postgresql_schemas

List all schemas in a PostgreSQL database.

Parameters:

- **database** (optional): Database name (uses the current connection if not specified)
- **limit** (optional): Maximum number of schemas to return (default: 100, max: 500)

Example:

```
list_postgresql_schemas("mydb")  
list_postgresql_schemas("mydb", 50)
```

list_postgresql_tables

List all tables in a PostgreSQL schema.

Parameters:

- **database** (optional): Database name (uses the current connection if not specified)
- **schema** (optional): Schema name (default: "public")
- **limit** (optional): Maximum number of tables to return (default: 100, max: 500)

Example:

```
list_postgresql_tables("mydb")  
list_postgresql_tables("mydb", "public", 50)
```

get_postgresql_table_info

Get information about a PostgreSQL table.

Parameters:

- **table** (required): Table name
- **database** (optional): Database name (uses the current connection if not specified)
- **schema** (optional): Schema name (default: "public")

Example:

```
get_postgresql_table_info("users", "mydb", "public")
```

Query Tools

execute_postgresql_query

Execute a SQL query on a PostgreSQL database.

Parameters:

- **query** (required): SQL query to execute
- **params** (optional): Query parameters in JSON format
- **database** (optional): Database name (uses the current connection if not specified)
- **limit** (optional): Maximum number of rows to return (default: 100, max: 500)

Example:

```
execute_postgresql_query("SELECT * FROM users WHERE age > 18")  
execute_postgresql_query("SELECT * FROM users WHERE age > $1", "[18]",  
"mydb")
```

get_postgresql_table_data

Get data from a PostgreSQL table.

Parameters:

- **table** (required): Table name
- **columns** (optional): Comma-separated list of columns to select
- **where** (optional): WHERE clause
- **params** (optional): Query parameters for WHERE clause in JSON format
- **order_by** (optional): ORDER BY clause
- **limit** (optional): Maximum number of rows to return (default: 100, max: 500)
- **database** (optional): Database name (uses the current connection if not specified)
- **schema** (optional): Schema name (default: "public")

Example:

```
get_postgresql_table_data("users")  
get_postgresql_table_data("users", "id,name,email", "age > $1", "[18]",  
"name ASC", 50, "mydb", "public")
```

explain_postgresql_query

Explain a SQL query on a PostgreSQL database.

Parameters:

- **query** (required): SQL query to explain
- **params** (optional): Query parameters in JSON format
- **database** (optional): Database name (uses the current connection if not specified)

Example:

```
explain_postgresql_query("SELECT * FROM users WHERE age > 18")
explain_postgresql_query("SELECT * FROM users WHERE age > $1", "[18]",
"mydb")
```

Information Tools

get_postgresql_server_info

Get PostgreSQL server information.

Example:

```
get_postgresql_server_info()
```

get_postgresql_server_stats

Get PostgreSQL server statistics.

Example:

```
get_postgresql_server_stats()
```

get_postgresql_active_queries

Get active queries on the PostgreSQL server.

Parameters:

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

Example:

```
get_postgresql_active_queries()
get_postgresql_active_queries(50)
```

get_postgresql_slow_queries

Get slow queries on the PostgreSQL server.

Parameters:

- **min_duration** (optional): Minimum query duration in milliseconds (default: 1000)
- **limit** (optional): Maximum number of queries to return (default: 100, max: 500)

Example:

```
get_postgresql_slow_queries()  
get_postgresql_slow_queries(2000, 50)
```

get_postgresql_table_bloat

Get table bloat information on the PostgreSQL server.

Parameters:

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

Example:

```
get_postgresql_table_bloat()  
get_postgresql_table_bloat(50)
```

get_postgresql_index_usage

Get index usage information on the PostgreSQL server.

Parameters:

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

Example:

```
get_postgresql_index_usage()  
get_postgresql_index_usage(50)
```

Available Resources

The PostgreSQL integration provides the following resources:

- `postgresql://info`: Get PostgreSQL server information
- `postgresql://stats`: Get PostgreSQL server statistics
- `postgresql://databases`: List all PostgreSQL databases
- `postgresql://db/{database}`: Get information about a PostgreSQL database
- `postgresql://db/{database}/schemas`: List all schemas in a PostgreSQL database
- `postgresql://db/{database}/schema/{schema}`: Get information about a PostgreSQL schema
- `postgresql://db/{database}/schema/{schema}/tables`: List all tables in a PostgreSQL schema
- `postgresql://db/{database}/schema/{schema}/table/{table}`: Get information about a PostgreSQL table

Examples

Working with Databases and Tables

```
# List all databases
list_postgresql_databases()

# Get information about a database
get_postgresql_database_info("mydb")

# List all schemas in a database
list_postgresql_schemas("mydb")

# List all tables in a schema
list_postgresql_tables("mydb", "public")

# Get information about a table
get_postgresql_table_info("users", "mydb", "public")
```

—
PROF

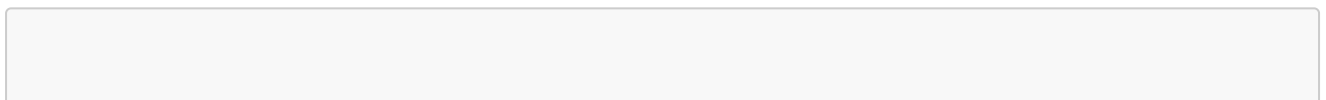
Working with Queries

```
# Execute a SQL query
execute_postgresql_query("SELECT * FROM users WHERE age > 18")

# Get data from a table
get_postgresql_table_data("users", "id,name,email", "age > 18")

# Explain a SQL query
explain_postgresql_query("SELECT * FROM users WHERE age > 18")
```

Working with Server Information



```
# Get server information
get_postgresql_server_info()

# Get server statistics
get_postgresql_server_stats()

# Get active queries
get_postgresql_active_queries()

# Get slow queries
get_postgresql_slow_queries()

# Get table bloat information
get_postgresql_table_bloat()

# Get index usage information
get_postgresql_index_usage()
```