OPEN SOURCE · APACHE 2.0 · TESTGEN 5.189.4

DataOps TestGen MCP

Why you should care: your AI assistant is smart but blind to your data quality. This fixes that. Ask in plain English; TestGen profiles the tables, writes the tests, runs them, and hands back the failing rows. **122 tools** (117 in open source) plus MCP resources and prompts, in Claude, Claude Code, Cursor, Copilot, or any MCP client. Free, like TestGen itself.

CONNECT

- **Endpoint** `https://<host>/mcp`
- **OAuth 2.1** paste the URL, sign in via browser
- **Token** header `Bearer <PAT>` (Enterprise)
- **Local** `http://localhost:<port>/mcp` (plain http)
- **Verify** "List the projects I have access to"

1 DISCOVER · 2 PROFILE · 3 HYGIENE · 4 TEST · 5 CONTRACT · 6 RUN · 7 MONITOR · 8 SCORE · 9 OPERATE

1 Discover your estate

Map projects, connections, table groups, tables, and columns. Sample real rows. Find columns anywhere. Rebuild DDL from profiles. Manage who belongs to a project.

`get_data_inventory` `list_projects` `get_project` ● `create_project` ENT
● `update_project` `list_project_members` ENT
● `set_project_membership` ENT `list_connections` `get_connection`
● `create_connection` ENT ● `update_connection` ENT `test_connection`
`list_table_groups` `get_table_group` ● `create_table_group`
● `update_table_group` ● `delete_table_group` `preview_table_group`
`list_tables` `get_table` `get_table_sample` `list_columns` `get_column`
`search_columns` `generate_create_table_script` ● `update_catalog_metadata`

2 Profile your data

Run profiling, compare runs, trend metrics over time, and catch schema drift. Drill to per-column stats, frequent values, and patterns.

● `run_profiling` ● `cancel_profiling_run` `list_profiling_runs`
`get_profiling_run` `list_profiling_summaries` `compare_profiling_runs`
`get_profiling_trends` `get_schema_history` `get_column_frequent_values`
`get_column_patterns`

3 Triage hygiene issues

Review what profiling flagged: bad blanks, type drift, PII, stale dates. Confirm, dismiss, or mute each finding.

`list_hygiene_issues` `search_hygiene_issues` `get_hygiene_issue`
● `update_hygiene_issue`

4 Generate & manage tests

Tests write themselves from the profile. Full CRUD, bulk toggles, portable export between suites, dry-run validation for custom SQL, and suites that push results into DataOps Observability.

● `generate_tests` `list_tests` `get_test` ● `create_test` ● `update_test`
● `bulk_update_tests` ● `delete_tests` `export_tests` ● `import_tests`
`validate_custom_test` `get_current_values` `list_test_types` `get_test_type`
`list_test_suites` `get_test_suite` ● `create_test_suite`
● `update_test_suite` ● `delete_test_suite` ● `create_test_note`
`list_test_notes` ● `update_test_note` ● `delete_test_note`

5 Write data contracts

Import a supplier's ODCS file or edit the draft. Freeze a version, diff two versions, and export the one in force for the other team.

● `import_contract` `get_contract_metadata` ● `update_contract_metadata`
● `freeze_contract_version` `list_contract_versions` `diff_contract_versions`
`export_contract` ● `discard_contract_draft`

6 Run tests, investigate failures

Execute suites. Summarize and trend failures. Compare runs for regressions. Pull the exact offending rows from the source database.

● `run_tests` ● `cancel_test_run` `list_test_runs` `get_test_run`
`compare_test_runs` `get_failure_summary` `get_failure_trend`
`list_test_results` `search_test_results` `list_test_result_history`
● `update_test_result` ● `bulk_update_test_results` `get_source_data`
`get_source_data_query`

7 Monitor tables

Anomaly monitors per table group: freshness, volume, and schema changes without writing a test. Add metric monitors on your own SQL and choose static, historical, or predictive thresholds.

● `enable_monitors` ● `disable_monitors` `get_monitor_settings`
● `update_monitor_settings` `get_monitor_summary` `list_monitors`
`list_monitored_tables` `list_monitor_events` `list_monitor_schema_changes`
● `create_metric_monitor` ● `update_metric_monitor` ● `delete_metric_monitor`
`validate_metric_expression` ● `set_monitor_static`
● `set_monitor_historical` ● `set_monitor_predictive`

8 Score quality

Total, CDE, Profiling, and Testing scores. Slice by quality dimension, business domain, or data product. Manage scorecards.

`get_quality_scores` `list_scorecards` `get_scorecard` ● `create_scorecard`
● `update_scorecard` ● `delete_scorecard`

9 Operate on a schedule

Recurring profiling and test runs. Alerts on failures, new hygiene issues, or a score dropping through a threshold, sent to the destinations you set up.

`list_schedules` `get_schedule` ● `create_profiling_schedule`
● `create_test_run_schedule` ● `update_schedule` ● `delete_schedule`
`list_notifications` `get_notification` ● `create_notification`
● `update_notification` ● `delete_notification`
`list_notification_destinations` `get_notification_destination`
● `create_notification_destination` ● `update_notification_destination`
● `delete_notification_destination`

PROMPTS TO TRY

```
> What data do I have and how healthy is it?
> Show me the source rows behind that failure
> Profile the sales tables, show new issues
> Import this ODCS file. Which terms fail?
> Generate tests, run them, sum up failures
> Email me if the score drops below 75
```

Built-in prompts (slash commands in most clients): `health_check` · `table_health` · `investigate_failures` · `hygiene_triage` · `profiling_overview` · `compare_runs`

● writes to TestGen • ENT Enterprise only • source-data & validation tools run read-only SQL • every tool runs with your TestGen role

 • © 2026 DataKitchen