MCP HARDENING RECORD

BLS OEWS MCP Testing Record

BLS wage data MCP hardened via retroactive live audit after a smoke test that said zero bugs

Skill
bls-oews-mcp v0.2.2
github.com/1102tools/bls-oews-mcp

Date
April 2026
1102tools.com

Executive Summary

This Model Context Protocol server exposes the BLS Occupational Employment and Wage Statistics (OEWS) API as callable tools for federal IGCE development, price analysis, and labor market research. It was hardened through a retroactive live audit with a real BLS API key after the sam-gov-mcp hardening demonstrated that mocked tests miss whole classes of bugs. The retroactive live audit surfaced 22 bugs including a P0 usability-breaker: the SOC code validator was rejecting "15-1252" (the exact format every BLS website and publication uses) and the smoke test had previously reported zero bugs. The MCP ships with 60 regression tests (55 offline plus 5 live-gated).

Metric	Value
MCP tools exposed	7
Total regression tests	60 (55 offline, 5 live-gated)
Audit rounds completed	5 (retroactive live audit)
P0 usability-breaking bugs found and fixed	1
P1 silent-wrong-data bugs found and fixed	10
P1 response-shape crash paths found and fixed	12
P2 validation gaps found and fixed	7
P3 cleanup items found and fixed	4
Current release	0.2.2
PyPI status	Published as `bls-oews-mcp`, auto-publishes via Trusted Publisher on tag push

What Was Tested

The MCP exposes 7 tools covering the BLS OEWS API surface. Testing covered all of them end-to-end.

Core: `get_wage_data`, `compare_metros`, `compare_occupations`, `igce_wage_benchmark`

Reference: `list_common_metros`, `list_common_soc_codes`, `detect_latest_year`

Each tool was exercised for argument validation, SOC code format normalization, state FIPS padding, response-shape guarantees against BLS's occasionally-inconsistent response shapes, error translation (including the REQUEST_PARTIALLY_PROCESSED status that BLS uses for partial data), and real-world data handling against the live production BLS v2 API with a real API key.

How It Was Tested

Testing discipline

The 0.1.1 smoke test for this MCP said "zero bugs." The retroactive live audit proved that was wrong. The hardening discipline established by sam-gov-mcp (run a live audit with a real API key, not just mocks) was applied retroactively here and surfaced 22 real bugs. Regression tests now invoke tools through `mcp.call_tool(name, kwargs)` the way a real MCP client does, and the live suite exercises the full stack against BLS's production v2 endpoint.

Audit rounds

Release	Context	Findings
0.2.0	Original hardening with mocks	Baseline validation
0.2.1	Cross-MCP `extra='forbid'` back-port	1 cross-fix
0.2.2	Full retroactive live audit with real BLS key: 5 audit rounds covering format validation, silent suppression, response-shape fuzzing, datatype mapping, validation gaps	22 real bugs

Live audit status

The retroactive audit in 0.2.2 used a real BLS v2 API key throughout. The repository includes 5 live-gated regression tests executable via `BLS_LIVE_TESTS=1 BLS_API_KEY=... pytest` covering real wage data retrieval, metro comparison, occupation comparison, IGCE wage benchmark, and the SOC format that previously failed validation ("15-1252" with dash). A BLS v2 key is free at `data.bls.gov/registrationEngine` and carries a 500-queries-per-day limit.

Issues Found and Fixed

Priority 0: Usability-breaking

One bug in this class, a hard blocker for new users.

Issue	Fix
SOC code validator rejected the standard BLS format "15-1252" (with dash). The regex required ASCII digits only, but every example on bls.gov and every BLS publication uses the dashed format (e.g. "15-1252" for Software Developers). Users pasting SOC codes directly from BLS got a hard "must contain only ASCII digits" error. Only the un-dashed "151252" worked.	Validator now accepts both `15-1252` and `151252`. Dash is stripped internally before forwarding. Regression tests cover both forms including mixed case and trailing whitespace.

Priority 1: Silent wrong data

Ten bugs in this class. Representative and signature items below.

Issue	Fix
`year=2023` (or any non-current year) returned ALL fields marked `suppressed: true`. BLS's public API only serves the current data year. Users requesting historical data thought the data was privacy-censored (a legitimate BLS suppression flag for low-observation cells) when in fact the API was just not serving that year at all.	Year tightened to `current ± 1` at the arg layer with an error message pointing to `bls.gov/oes/tables.htm` for historical data.
`occ_code="99-9999" returned 4 fully-formed "suppressed" benchmarks with `occ_title: "99-9999"`. The tool happily generated a benchmark table for a nonexistent SOC code.	`no_data` flag and `no_data_reason` field added when all values are null. `_title_warning` added to IGCE output when the SOC code is not in the known lookup.
Nonexistent state FIPS ("99") returned all-suppressed with no warning. Same failure mode.	State FIPS validated against known set; unknown codes raise actionable error.
Nonexistent metro code ("99999") returned all-suppressed with no warning. Same failure mode.	Metro codes validated; unknown codes raise actionable error.
Nonexistent industry ("999999") returned all-suppressed with no warning. Same failure mode.	Industry codes validated; unknown codes raise actionable error.
`compare_metros` silently accepted 2-digit state FIPS mixed with 5-digit MSAs. Returned 0 results without explanation.	Mixed-format input raises `ValueError` pointing at `compare_occupations(scope='state')`.
`compare_metros` with all-duplicate codes silently deduplicated to one and somehow returned 0 results.	Duplicates now flagged with a clear warning; single-code input is handled as a pass-through to `get_wage_data`.
Data-year field in the response was the API's latest regardless of the `year` parameter. No validation that the API actually returned the requested year.	Response year is now compared against the requested year; mismatch raises actionable error.

Issue	Fix
Short SOC like "15-125" (6 chars with dash) bypassed validation because length check was pre-strip. Valid SOCs are always 7 chars ("XX-XXXX").	Length check now post-normalization; 6-char input rejected.
Mixed metro and state codes in `compare_occupations` silently returned 0 results from scope confusion.	Codes checked for consistent scope; mixed input raises clear error.

Priority 1: Response-shape crashes

Twelve distinct crash paths in the BLS response parser from round 4 mock fuzzing. BLS's v2 API occasionally collapses single-element lists to dicts and returns partial-processed responses with non-standard status fields.

Issue	Fix
`series` returned as dict instead of list (XML-to-JSON collapse) → `TypeError: string indices must be integers`.	`_as_list` normalizer wraps `series`.
`data` returned as dict instead of list → `KeyError: 0`.	Same `_as_list` coercion.
Entry missing `value` field → `KeyError: 'value'`.	`_extract_first_data_entry` helper with `.get()` throughout.
Entry missing `year` field → `KeyError: 'year'`.	Guarded.
Series item missing `seriesID` → `KeyError: 'seriesID'`.	`_series_id_from` helper returns None and logs if missing.
`footnotes` as dict instead of list → `AttributeError: 'str' object has no attribute 'get'`.	`_safe_footnotes` helper normalizes.
`footnotes` as string → `AttributeError`.	Same helper.
Data array with None entries → `AttributeError: 'NoneType'`.	None entries filtered.
Series list with None entries → `TypeError: 'NoneType' object is not subscriptable`.	Same filtering.
`JSONDecodeError` unhandled. BLS returns HTML during maintenance windows and empty body during error states.	`_clean_error_body` helper catches and re-raises with API context.
`REQUEST_PARTIALLY_PROCESSED` status was silently treated as success. BLS returns partial results with messages like "Series not found" but the status is not `REQUEST_NOT_PROCESSED`; the code only checked that exact status.	All partial-processed responses now inspect the `message` field and surface per-series errors.
Int `seriesID` (non-string) caused `sid[-2:]` slice to crash.	`_coerce_str_digits` helper normalizes to string.

Helpers introduced as part of the response-shape fix pass: `_as_list`, `_coerce_str_digits`, `_validate_soc`, `_validate_industry`, `_validate_datatype`, `_validate_year`, `_ex-`

`tract_first_data_entry`, `_safe_footnotes`, `_series_id_from`, `_clean_error_body`, `_api_key_status`.

Priority 2: Validation gaps

Seven bugs in this class:

Issue	Fix
Single-digit state FIPS ("6" for California) was rejected. Users commonly know FIPS codes as 1 digit (CA=6, AK=2, not 06 / 02).	Auto-pad to 2 digits.
Newline, tab, carriage return in `occ_code` slipped through `strip()` because they were internal, not leading or trailing.	Control chars rejected before strip.
`OEWS_LATEST_FUTURE_YEAR = 2100` allowed years up to 2100 that will never have data.	Tightened to `current ± 1`.
`DATATYPE_LABELS["08"]` said "Hourly Median" but empirical BLS data shows dt=08 returns 25th percentile values. Mislabeled.	Relabel corrected after empirical verification.
`DATATYPE_LABELS` was missing labels for valid datatypes 07, 09, 10, 16. Users saw raw code as label.	All valid datatypes now have labels.
Bogus datatypes like "99" or "AA" were silently accepted and the API call was wasted, returning empty wages.	Validated against the known datatype set.
`igce_wage_benchmark` `burden_low > burden_high` silently produced nonsense "3.0x to 1.5x" range. `burden_low=-1.0` or 0 was accepted.	Reversed range raises actionable error. Negative and zero burdens rejected.

Priority 3: Cleanup items

Four items: `detect_latest_year` was silently swallowing all exceptions (a 429 rate-limit was becoming a misleading "no newer data available" message), the USER_AGENT was stale at `bls-oews-mcp/0.1.1`, `OEWS_CURRENT_YEAR` was a module-level constant that would not auto-update when BLS released a new data year, and there was no retry on 429. All resolved.

Response-shape defense

The eleven helpers introduced in 0.2.2 (listed above) now wrap every BLS response parsing path. BLS's v2 API occasionally returns shapes that do not match its documentation, especially single-element list-to-dict collapses and partial-processed responses. All variants now normalize cleanly with logged warnings when an unexpected shape is observed.

Test Coverage

The repo ships 60 regression tests. All 60 pass on every release cycle.

File	Purpose	Test count
`tests/test_validation.py`	Main regression suite covering every round finding, plus 5 live-gated integration tests	60
`tests/stress_test_live.py`	Retroactive live-audit scenario scripts for SOC format, suppressed-year detection, nonexistent-code detection, response-shape edge cases (retained for reproducibility)	N/A (scenario script)

Regression tests invoke tools through the FastMCP registry (`mcp.call_tool`). An autouse fixture resets the shared httpx client between tests.

Release History

Version	Focus	Outcome
0.1.1	Initial release (smoke tested, reported "zero bugs"; reality was 22+ lurking)	Baseline coverage
0.2.0	First hardening pass (mocks only)	Baseline validation
0.2.1	Cross-MCP `extra='forbid'` back-port from sam-gov-mcp 0.3.1	+1 regression test
0.2.2	Full retroactive live audit with real BLS key: 22 bugs fixed across 5 rounds; 60 regression tests including 5 live-gated	1 P0, 10 P1 silent-wrong-data, 12 P1 crash paths, 7 P2, 4 P3 resolved

Cross-MCP Context

This MCP is one of eight servers in the 1102tools federal-contracting MCP suite (`ecfr-mcp`, `federal-register-mcp`, `gsa-calc-mcp`, `gsa-perdiem-mcp`, `regulationsgov-mcp`, `sam-gov-mcp`, `usaspending-gov-mcp`, and this one). All eight were hardened under the same playbook. Patterns reused or established here:

- **"Smoke test said zero, live audit found everything" lesson** was codified here. The 0.1.1 smoke test claimed zero bugs; the live audit in 0.2.2 found 22. This is the clearest demonstration in the suite that mocked tests miss real bugs.
- **Response-shape defensive-parsing helpers** `_as_list`, `_extract_first_data_entry`, `_safe_footnotes` were exported to other MCPs that face similar XML-to-JSON collapse edge cases.
- **`_api_key_status` pattern** for warning the user when an API key is empty or whitespace was codified here.
- **`extra='forbid'` on every tool's pydantic arg model** was back-ported from sam-gov-mcp 0.3.1 in the 0.2.1 release.

What Was Not Tested

- **Rate-limit behavior beyond 500 queries per day.** The BLS v2 free tier is capped at 500 queries per day. The MCP surfaces 429s but does not implement client-side throttling.
- **Historical data years (prior-year and further back).** BLS's public API only serves current year. Users needing historical data are directed to `bls.gov/oes/tables.htm`.
- **Wage data during the annual BLS data release window.** BLS publishes new OEWS data roughly in April each year; the window when the new year's data appears and stabilizes has not been live-audited.
- **v1 (legacy) API endpoints.** This MCP uses v2 only.

Verification

All testing artifacts are in the repository. The methodology and fixes are reviewable commit-by-commit in git history. The regression test suite runs via `pytest` in the repo root and can be re-executed by anyone. The live suite runs with `BLS_LIVE_TESTS=1 BLS_API_KEY=... pytest` using a free BLS v2 API key.

> **Testing Methodology**
>
> Evaluators: James Jenrette, 1102tools, with Claude Code Opus 4.7 (1M context, max effort, Claude Max 20x subscription) during the hardening playbook execution.
>
> Testing in 0.2.2 spanned five rounds covering SOC format validation, silent-suppression detection, response-shape mock fuzzing (12 distinct crash paths), datatype-mapping correctness, and validation gap audit. The live regression suite runs against the production BLS v2 API when enabled with `BLS_LIVE_TESTS=1`.
>
> Test count: 60 regression tests. P0 usability-breaking bugs found and fixed: 1. P1 silent-wrong-data bugs found and fixed: 10. P1 response-shape crash paths found and fixed: 12. P2 validation gaps closed: 7. P3 cleanup items closed: 4. Total findings: 22. Current version: 0.2.2. PyPI: `bls-oews-mcp`.
>
> Source: github.com/1102tools/bls-oews-mcp. License: MIT.