



Rethinking Testing for MCP Servers

Puja Jagani · Lead of Open Source & Developer Advocacy · BrowserStack




A NEW KIND OF CLIENT



CLIENT BEFORE MCP

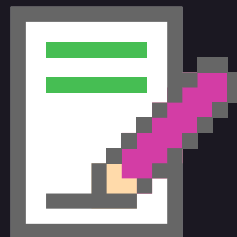
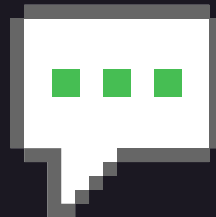
- Deterministic.
- Follows a script.
- You wrote every call.

LLM-DRIVEN CLIENT

- Non-deterministic.
 - Reads descriptions.
 - Calls tools in any order.
- 

CURRENT LANDSCAPE

- Absence of MCP testing guide.
- Happy path testing is prevalent.
- No mental model for how LLMs interact with servers.



THE TESTING MODEL



LAYER 1

Contract
Testing

Descriptions
match
behaviour



LAYER 2

Sequence
Testing

Tools handle
unexpected
order



LAYER 3

Error
Channel
Testing

Failures
surface to the
LLM





CONTRACT TESTING

Descriptions are not documentation. They are contracts.

CONTRACT TESTING



description: "Searches BrowserStack
docs"



TOOL DESCRIPTION



WHAT IT DOES

Searches across all BrowserStack product documentation

EXAMPLE INPUTS

"run parallel tests"
"upload app file"
"set capabilities"



SCOPE / CONSTRAINT

browserstack.com/docs –
Automate, App Automate,
Live,

NEXT STEPS

Returns { product, title,
url } → pass url to
get_page



HOW TO TEST IT?



Does it explain what the tool does?

Check descriptions contain structural markers.

Review descriptions like you review code.



TESTING STRUCTURAL MARKERS



```
FOR EACH tool in server.tools:
```

```
    ASSERT description is meaningful
```

```
        → length > 50 characters
```

```
    ASSERT description shows an example
```

```
        → contains a quoted string or "e.g."
```

```
    ASSERT description explains the output
```

```
        → contains "Returns" or "result"
```



02

SEQUENCE TESTING



LLMs don't read the before blocks

WHY SEQUENCE MATTERS?

```
Last login: Tue Jun 2 11:13:26 on ttya000
puja@browserstack.com@puja-CR6J761YD3 mcp-browserstack-docs % cd /Users/puja@browserstack.com/Documents/Di
tHub/mcp-selenium
puja@browserstack.com@puja-CR6J761YD3 mcp-selenium % node probe.mjs
--- Calling navigate WITHOUT calling start_browser first ---
Waiting for server response...
█
```

```
McpError -32001: Request timed out · { timeout: 60000 }
```

HOW TO TEST?



FOR EACH tool with a prerequisite:

IDENTIFY the dependency

→ what must be called first?

TEST the wrong sequence

→ call the tool without the prerequisite

ASSERT failure is clean, not silent

→ isError: true with a message

that tells the LLM what to call next



03



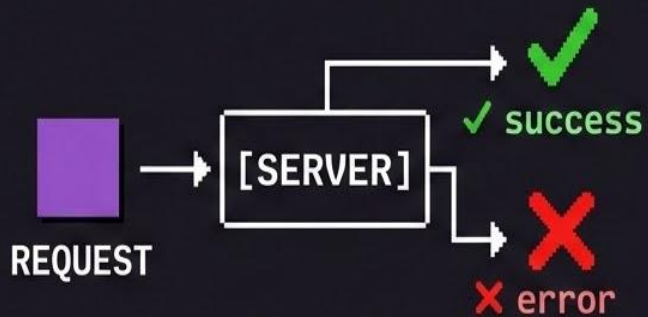
ERROR CHANNEL TESTING



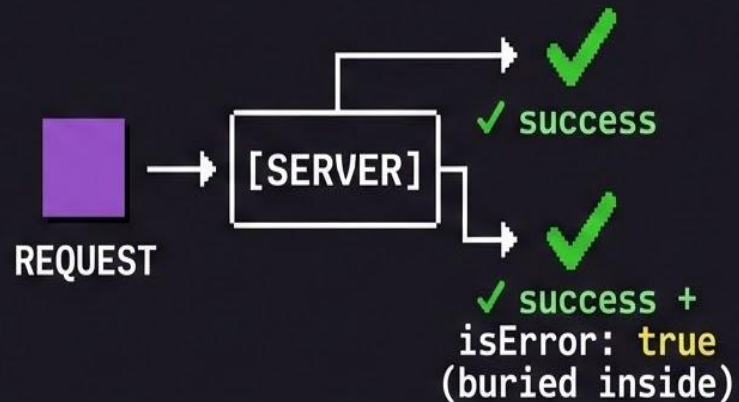
MCP has 2 error channels. Must test both equally.

TWO ERROR CHANNELS

WHAT YOU EXPECT



WHAT MCP ACTUALLY HAS



HOW TO TEST?



FOR EACH tool failure path:

TRIGGER the failure

→ empty query, invalid input,
missing prerequisite

ASSERT `isError` is explicitly `true`

ASSERT error message is actionable


→ tells LLM what to `do` next
→ not just what went wrong

TESTS IN ACTION


```
✓ search_form validation is successful and longer than 20 characters (0.26689sec)
✓ search_form validation accepts alphanumeric chars (0.136989sec)
✓ search_form validation contains an asterisk when (0.16683sec)
✓ get_page_validation_returns_response_object(0.126799sec)
✓ validate declares both 'question' and 'reason' as required (5.00000sec)
✓ every form declares at least one required input field (0.90409sec)
* Layer 2 - Request tests (177.864789sec)
* Layer 2 - Sequence tests
✓ search_docs with empty query returns (error) true (1.376189sec)
✓ get_page with a non-browserstack domain returns (error) true (0.710041sec)
✓ get_page with a browserstack url not under root returns (error) true (1165.46666sec)
✓ get_page with a browserstack browserstack-link url returns (error) true (170.857941sec)
✓ validate with empty question returns (error) true (11.498039sec)
✓ validate with empty reason returns (error) true (8.843437sec)
✓ correct sequence url returned by search_docs passes get_page schema validation (106.901799sec)
* Layer 2 - Sequence tests (1778.053779sec)
* Layer 3 - Error channel tests (1)
✓ search_docs / empty query - (error) true, never throws (0.58228sec)
✓ search_docs / query with characters to replace - (error) true, never throws (1.910431sec)
✓ get_page / malformed url string - (error) true, never throws (0.407942sec)
✓ get_page / non-browserstack domain - (error) true, never throws (1.21179sec)
✓ get_page / browserstack url not under docs - (error) true, never throws (0.714711sec)
✓ get_page / browserstack wrong url - (error) true, never throws (0.231619sec)
✓ validate / empty question field - (error) true, never throws (0.407979sec)
✓ validate / empty reason field - (error) true, never throws (0.438379sec)
✓ validate / both fields empty - (error) true, never throws (8.223379sec)
✓ search_docs re-rejects error tells the sim to try different search terms (8.872792sec)
✓ get_page invalid url error message tells the sim what a valid url looks like (8.426671sec)
✓ get_page wrong-domain error corroborates the required browserstack.com/docs format (300.316611sec)
✓ validate empty question error names the 'question' field (11.472799sec)
✓ validate empty reason error names the 'reason' field (8.866991sec)
✓ validate with valid inputs returns structured response with suggested_urls and suggested_info_for_profile (11.618984sec)
* Layer 3 - Error channel tests (11247.80079sec)
* 00000000
* 00111111
* 00000000
```

TESTS IN ACTION

```
✓ exactly three valid xss registered: search_desc, get_page, evaluate (0.369370ms)
✓ every valid description is non-empty and longer than 20 characters (0.000447ms)
✓ search_desc description contains "browserstack.com" (0.181870ms)
✓ search_desc description contains an example query (0.371800ms)
✓ get_page description contains "browserstack.com/docs" (0.36960ms)
✓ evaluate decides both 'question' and 'reason' as required fields (0.125500ms)
✓ every valid decision at least one required input field (0.181400ms)
✓ Layer 2 - Compact tests (0.000100ms)
▶ Layer 3 - Sequence tests
  ✓ search_desc with empty query returns isError: true (2.546000ms)
  ✓ get_page with a non-browserstack domain returns isError: true (0.741000ms)
  ✓ get_page with a browserstack URL not under /docs returns isError: true (8.496000ms)
  ✓ get_page with a fabricated browserstack-like URL (wrong TLD) returns isError: true (8.800000ms)
  ✓ evaluate with empty question returns isError: true (8.448700ms)
  ✓ evaluate with empty reason returns isError: true (8.800000ms)
  ✓ correct sequence: URL returned by search_desc passes get_page schema validation (0.000400ms)
✓ Layer 3 - Sequence tests (118.454700ms)
▶ Layer 3 - Error channel tests
  ✓ search_desc / empty query = isError: true, never throws (0.740000ms)
  ✓ search_desc / query with guaranteed no results = isError: true, never throws (2.738000ms)
  ✓ get_page / malformed URL string = isError: true, never throws (0.379000ms)
  ✓ get_page / non-browserstack domain = isError: true, never throws (8.800000ms)
  ✓ get_page / browserstack URL not under /docs = isError: true, never throws (8.799000ms)
  ✓ get_page / fabricated wrong TLD = isError: true, never throws (0.240417ms)
  ✓ evaluate / empty question field = isError: true, never throws (0.307000ms)
  ✓ evaluate / empty reason field = isError: true, never throws (0.307000ms)
  ✓ evaluate / both fields empty = isError: true, never throws (8.360000ms)
  ✓ search_desc no-results error tells the LHM to try different search terms (8.707000ms)
  ✓ get_page invalid URL error message tells the LHM what a valid URL looks like (8.24000ms)
  ✓ get_page wrong-domain error references the required browserstack.com/docs format (8.800000ms)
  ✓ evaluate empty question error names the 'question' field (8.240000ms)
  ✓ evaluate empty reason error names the 'reason' field (8.200000ms)
  ✓ evaluate with valid inputs returns 40 returned responses with suggested_queries and suggested_refs_by_page (8.270000ms)
✓ Layer 3 - Error channel tests (7.487700ms)
+ 22ms 37
+ 3/1ms 3
```



Test what
you control



THANK YOU!

puja@browserstack.com

puja.jagani93@gmail.com

[https://www.linkedin.com/
in/pujajagani/](https://www.linkedin.com/in/pujajagani/)



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

