

No2Bounce for Cursor

Validate emails without leaving your editor.

● VALID ● RISKY ● INVALID

Every marketing team knows the quiet cost of a bad email list: bounced campaigns, wasted sends, and a sender reputation that takes months to repair after one bad blast. Most teams do not catch it until the bounce report lands, and by then the damage is done. The No2Bounce MCP server moves that check earlier, into the conversation where you are already planning your outreach.

Connect Cursor to the No2Bounce API and check single addresses or bulk lists, get deliverability scores, and track jobs, all through plain-language requests.

DOCUMENT TITLE	No2Bounce for Cursor – MCP Integration Guide
SUBJECT	Connecting Cursor to the No2Bounce validation API via MCP
PREPARED FOR	Internal / Client Reference
DATE	August 31, 2026
STATUS	Final

TABLE OF CONTENTS

- 1 What MCP Does Here
- 2 What Cursor Can Check
- 3 Setup
- 4 Easy Setup — One-Prompt Installation
- 5 CTA & Package Info

1 What MCP Does Here

MCP, the Model Context Protocol, lets Cursor reach outside the editor and call real external tools on your behalf. Instead of switching to a browser tab mid-task, you ask the agent. The No2Bounce MCP server connects Cursor directly to the validation API, which matters most when you are seeding a test list, wiring up a signup form, or sanity-checking the addresses in a fixture file before they reach production.

2 What Cursor Can Check

- ▶ **Single address:** a deliverability score out of 100, a status (valid, invalid, risky), and a sub-status explaining why — confirmed mailbox, catch-all domain, role-based address, or likely bounce.
- ▶ **Bulk lists:** up to 10,000 addresses per request, returned as a downloadable CSV.
- ▶ **Long-running jobs:** poll a job by tracking ID and pull results whenever they're ready, useful for auditing an old list or validating before a CRM import.
- ▶ **Account status:** confirm your API key is active and check your remaining credit balance, so you're never caught mid-campaign.

3 Setup

Install the server with one npm command, grab your API key from the No2Bounce dashboard, and add a short block to Cursor's MCP config. Use `~/.cursor/mcp.json` to make it available in every project, or `.cursor/mcp.json` inside a repo to scope it to that project and commit it for the team.

```
npm install -g no2bounce-mcp

{
  "mcpServers": {
    "no2bounce": {
      "command": "no2bounce-mcp",
      "args": ["--api-key=YOUR_API_KEY"]
    }
  }
}
```

Open Settings, then MCP, and confirm the no2bounce entry shows as connected. The tools are then available to the agent in Composer. Worth noting: Cursor keeps a ceiling of roughly forty active tools across all servers combined, so if you run a long list of MCP servers, disable the ones you are not using that day.

There is no server to host, no build step, nothing else to maintain. Your API key stays local the whole time, passed as a command-line argument.

4 Easy Setup — One-Prompt Installation

The entire setup above can be handed to Cursor itself as a single prompt, no manual file editing required. Paste the prompt below, replace YOUR_API_KEY with your real token, and let the agent do the rest.

```
Install and set up the No2Bounce MCP server for me. Run "npm install -g no2bounce-mcp" globally, then open or create ~/.cursor/mcp.json and add a "no2bounce" entry under mcpServers using command "no2bounce-mcp" and args ["--api-key=YOUR_API_KEY"]. Show me the final config file content, then tell me to reload Cursor and confirm the no2bounce tools are available.
```

From there, it's just conversation. Paste an address, paste a list, or ask about your account, and let Cursor check it before you hit send.

5 CTA & Package Info

Get your API key → <https://app.no2bounce.com/api>

View on npm → <https://npmx.dev/package/no2bounce-mcp>

Published by no2bounce-com on npm → <https://www.npmjs.com/~no2bounce-com>