From Dashboard to AI Automation

Building an MCP Server for PurelyMail

Guilherme Fontes

September 23, 2025

Section 1

Building an MCP Server for PurelyMail

Problem: Manual email management friction

Solution: AI-powered automation via MCP

Result: Closer to Human-thinking-like account operations

The Problem: Death by Dashboard Clicks

- **Adding users** → 7 clicks
- **Managing app passwords** → 5 clicks
- **Domain configuration** → 10+ clicks
- **Repetitive tasks** → Mounting frustration

What is MCP?

Model Context Protocol = API to an API

`AI Assistant ↔ MCP Server ↔ PurelyMail API`

- **Standardized interface** for AI tool integration
- **Type-safe operations** with automatic validation
- **Natural language** commands for complex tasks

Solution Evolution

`Manual Clicks → Direct Scripts → MCP Abstraction`

The Discovery: - Found PurelyMail's OpenAPI specification - Automated client generation from swagger - Built systematic, reproducible methodology

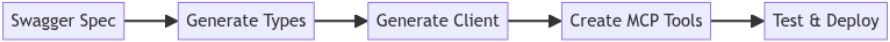
Discovery Process

Step 1: Found the OpenAPI specification

`https://news.purelymail.com/api/swagger-spec.js`

Step 2: Automated everything - Type generation from swagger - Client code generation - Tool definitions creation

Build Process



Commands:

```
npm run update:api    # Fetch + generate everything
npm run inspector     # Test tools locally
```

Code Examples

Tool Definition:

```
{
  name: "manage_user",
  description: "Create, delete, modify users",
  inputSchema: UserManagementSchema
}
```

Usage:

```
await client.manage_user({
  action: "Create User",
  userName: "newuser",
  domainName: "example.com"
});
```

Live Demo Time

What we'll see: - MCP tools in action - Real PurelyMail account operations - Behind-the-scenes code generation - Manual vs automated comparison

Switching to live demo...

Results & Impact

Time Saved: - Manual process: ~2 minutes per user - MCP automation: ~10 seconds per user - **12x speed improvement**

Beyond Speed: - Natural language interface - Transferable methodology

Key Takeaways

1. **Find OpenAPI specs** - they're a door for automation
2. **Generate, don't write** - let tools create boilerplate
3. **MCP abstraction** $>$ direct API scripting
4. **Automatic methodology** beats one-off solutions

Food for thought

Your turn: What manual processes are begging for this treatment?

who knows... maybe you will have a “shower-idea”...

Thank You

Questions?

GitHub Repository

All Resources