Setting up Atlassian MCP on Roo Code 🦘

or <<Setting up MCP servers on "any" MCP clients>> , really

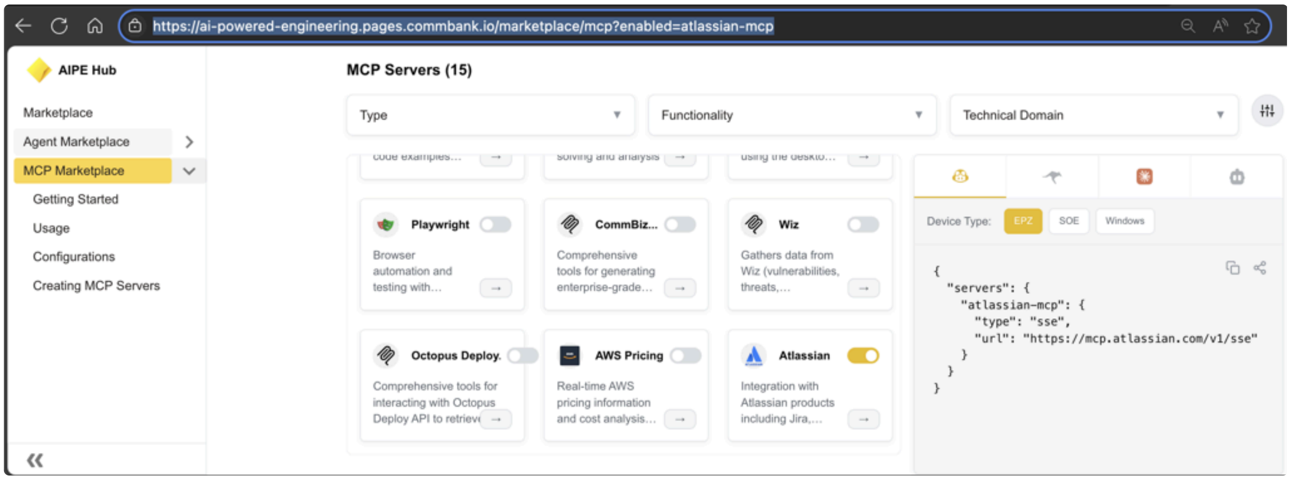
Introduction

Since MCP (Model Context Protocol) is a standardized protocol, MCP endpoints are designed to work universally across different clients. In theory, any MCP endpoint should function seamlessly whether you're using a command-line interface, a full GUI application, or an IDE plugin—as long as these clients adhere to the protocol's RFC specifications. This standardization means that if an Atlassian MCP endpoint works with Claude (which isn't surprising, given that Anthropic developed both Claude and the MCP standard), it should also be compatible with other MCP-compliant clients like Roo Code. However, theory doesn't always match reality. This article will walk you through overcoming the integration challenges when connecting Atlassian MCP with Roo Code, turning what should work in principle into a working solution in practice.

Use Case: Atlassian MCP with Roo Code

A quick look at our MCP Marketplace reveals the issue: there's no configuration snippet available for Atlassian with Roo Code (you'll see it's greyed out when you hover over it).

More telling is what's missing from the existing snippets—there's no authentication code included. This absence of authentication handling is exactly the challenge we need to solve.


https://ai-powered-engineering.pages.commbank.io/marketplace/mcp?enabled=atlassian-mcp

AIPE Hub
- Marketplace
- Agent Marketplace
- MCP Marketplace
  - Getting Started
  - Usage
  - Configurations
  - Creating MCP Servers

MCP Servers (15)

Type | Functionality | Technical Domain

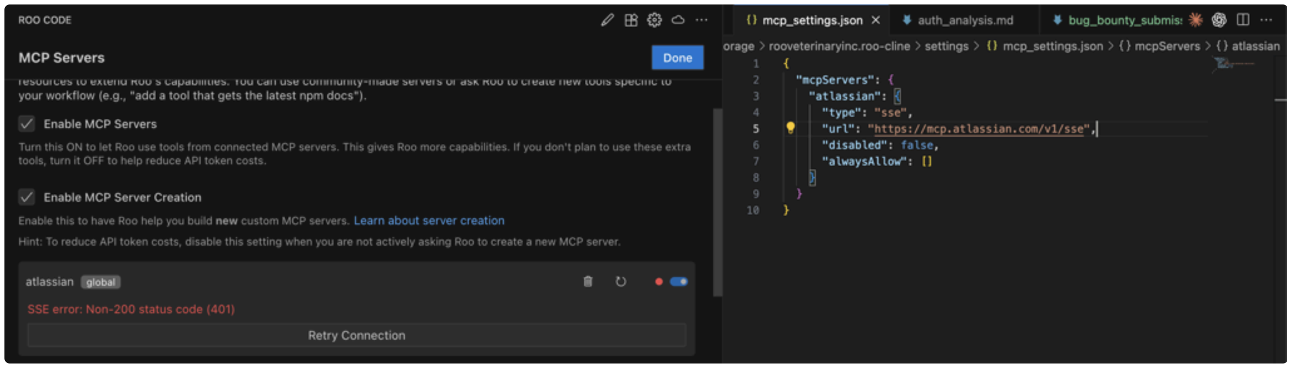
Playwright — Browser automation and testing with…
CommBiz… — Comprehensive tools for generating enterprise-grade…
Wiz — Gathers data from Wiz (vulnerabilities, threats,…
Octopus Deploy — Comprehensive tools for interacting with Octopus Deploy API to retrieve…
AWS Pricing — Real-time AWS pricing information and cost analysis…
Atlassian — Integration with Atlassian products including Jira,…

Device Type: EPZ SOE Windows

```
{
  "servers": {
    "atlassian-mcp": {
      "type": "sse",
      "url": "https://mcp.atlassian.com/v1/sse"
    }
  }
}
```

Check out our MCP Marketplace for more integrations!

Roo Code INCOMPLETE MCP Setup


ROO CODE

MCP Servers Done

resources to extend Roo's capabilities. You can use community-made servers or ask Roo to create new tools specific to your workflow (e.g., "add a tool that gets the latest npm docs").

✓ Enable MCP Servers

Turn this ON to let Roo use tools from connected MCP servers. This gives Roo more capabilities. If you don't plan to use these extra tools, turn it OFF to help reduce API token costs.

✓ Enable MCP Server Creation

Enable this to have Roo help you build **new** custom MCP servers. Learn about server creation

Hint: To reduce API token costs, disable this setting when you are not actively asking Roo to create a new MCP server.

atlassian global

SSE error: Non-200 status code (401)

Retry Connection

mcp_settings.json | auth_analysis.md | bug_bounty_submis

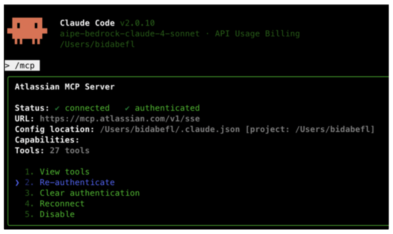
orage > rooveterinaryinc.roo-cline > settings > mcp_settings.json > mcpServers > atlassian

```
{
  "mcpServers": {
    "atlassian": {
      "type": "sse",
      "url": "https://mcp.atlassian.com/v1/sse",
      "disabled": false,
      "alwaysAllow": []
    }
  }
}
```

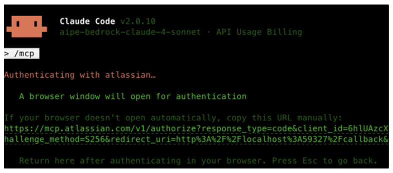
As expected, simply copying this snippet into Roo Code doesn't work—it fails due to authentication issues.
hence why it was greyed out in the marketplace...

Claude NORMAL MCP Setup

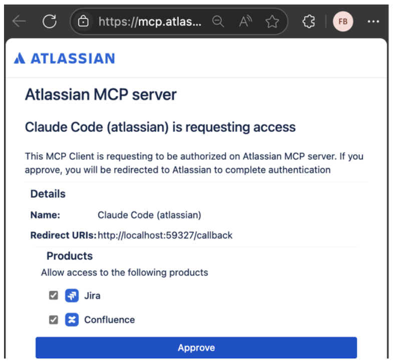
Now, if we configure it in Claude, we notice that Claude offers to authenticate!


```
Claude Code v2.0.10
aipe-bedrock-claude-4-sonnet · API Usage Billing
/Users/bidabefl

> /mcp

Atlassian MCP Server

Status: ✓ connected  ✓ authenticated
URL: https://mcp.atlassian.com/v1/sse
Config location: /Users/bidabefl/.claude.json [project: /Users/bidabefl]
Capabilities:
Tools: 27 tools

  1. View tools
❯ 2. Re-authenticate
  3. Clear authentication
  4. Reconnect
  5. Disable
```

1. Claude before/after setup


```
Claude Code v2.0.10
aipe-bedrock-claude-4-sonnet · API Usage Billing

> /mcp

Authenticating with atlassian…

   A browser window will open for authentication

If your browser doesn't open automatically, copy this URL manually:
https://mcp.atlassian.com/v1/authorize?response_type=code&client_id=6hlUAzcX
hallenge_method=S256&redirect_uri=http%3A%2F%2Flocalhost%3A59327%2Fcallback&

   Return here after authenticating in your browser. Press Esc to go back.
```

2. the Auth request


https://mcp.atlas…

ATLASSIAN

Atlassian MCP server

Claude Code (atlassian) is requesting access

This MCP Client is requesting to be authorized on Atlassian MCP server. If you approve, you will be redirected to Atlassian to complete authentication

Details

Name:	Claude Code (atlassian)
Redirect URIs:	http://localhost:59327/callback

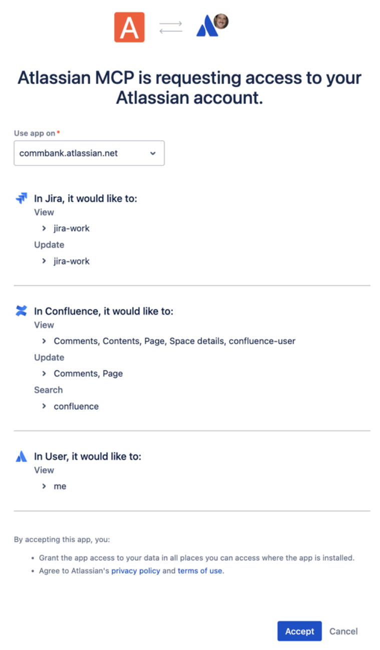
Products

Allow access to the following products

- ☑ Jira
- ☑ Confluence

Approve

3. Opens in the browser, and go on with the OAuth approval process



Atlassian MCP is requesting access to your Atlassian account.

Use app on *

commbank.atlassian.net

In Jira, it would like to:

View
- jira-work

Update
- jira-work

In Confluence, it would like to:

View
- Comments, Contents, Page, Space details, confluence-user

Update
- Comments, Page

Search
- confluence

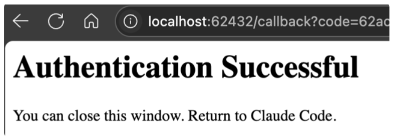
In User, it would like to:

View
- me

By accepting this app, you:

- Grant the app access to your data in all places you can access where the app is installed.
- Agree to Atlassian's privacy policy and terms of use.

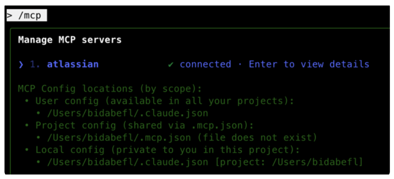
4. Select what access you're willing to grant



Authentication Successful

You can close this window. Return to Claude Code.

5. You're done with the browser



```
> /mcp

Manage MCP servers

❯ 1. atlassian            ✓ connected · Enter to view details

MCP Config locations (by scope):
 • User config (available in all your projects):
   • /Users/bidabefl/.claude.json
 • Project config (shared via .mcp.json):
   • /Users/bidabefl/.mcp.json (file does not exist)
 • Local config (private to you in this project):
   • /Users/bidabefl/.claude.json [project: /Users/bidabefl]
```

6. And Claude Caude can connect to Confluence and Jira!

The inner workings...

The URL shows the OAuth 2.0 authorization flow with PKCE (Proof Key for Code Exchange). Here are the variables:

`https://mcp.atlassian.com/v1/authorize`

- `response_type=code` - Indicates this is using the authorization code flow
- `client_id=6******************S` - The unique identifier for the application requesting access
- `code_challenge=03***************************AZHV4` - PKCE code challenge (SHA256 hash of a code verifier)
- `code_challenge_method=S256` - Specifies SHA256 is used for the code challenge
- `redirect_uri=http%3A%2F%2Flocalhost%3A59327%2Fcallback` - Where the user will be redirected after authorization (URL encoded: `http://localhost:59327/callback`)

- `state=_XMH77au3-****_DHg6gAT************rmr3zpHM` - Random string to prevent CSRF attacks and maintain state between request and callback

Roo Code TWEAKED / COMPLETE MCP Setup

Now, where is that bearer token ?

Claude Code uses Bearer token authentication to connect to mcp.atlassian.com

Using Charles to capture Claude request to the MCP, we can “steal” the bearer token we are after from the HTTP authorization header…

*And just like **R**obin H**oo**d, we’ll share it with… Roo*

Note you might also try using Wireshark, or NodeJS DEBUG to find this header, but a local Web Proxy might just be easier…

Install and configure Charles

As per Web Debugging with Charles

Make sure to enable SSL proxying and to trust Charles Root CA (that’ll’ intercept the TLS encrypted traffic and allow you to see the non-encrypted payload **and headers**)

Charles should forward the traffic to external proxies (Prisma , or Alpaca if you have it)

Shell Config

Point your Shell to Charles:

```
export {ALL,HTTP,HTTPS}_PROXY=http://localhost:8888
```

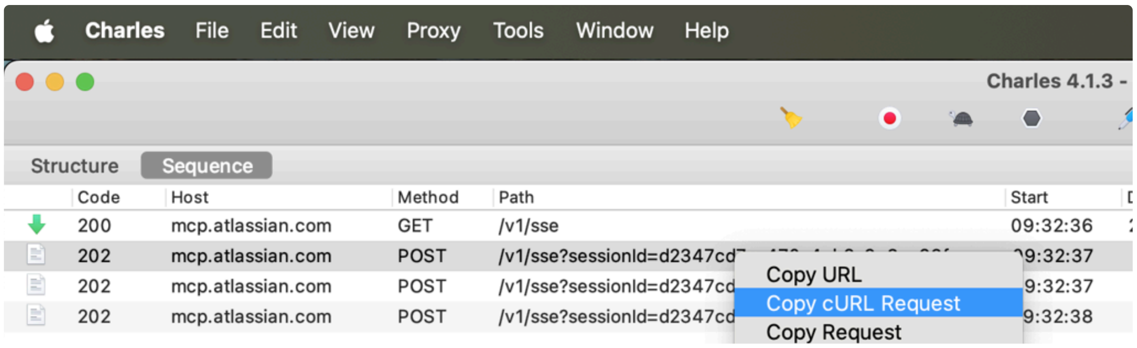
And trust the signing CA for NodeJS:

`export NODE_USE_SYSTEM_CA=1` if you have a NodeJS version `24+` ← Make sure you’ve have Charles RootCA trusted, in the Keychain Access (or Trust it)

or `export NODE_EXTRA_CA_CERTS="/Users/bidabefl/.config/cacert/cacert.pem"` ← This PEM needs to have the Base64 entry for your `Charles Root CA`

If you need to generate a PEM Certificate Store, check out: Group Internal Root CAs Certificate Store | Method 1 Automated

That’s it… SSL and Proxy setup



Code	Host	Method	Path	Start
200	mcp.atlassian.com	GET	/v1/sse	09:32:36
202	mcp.atlassian.com	POST	/v1/sse?sessionId=d2347cd	09:32:37
202	mcp.atlassian.com	POST	/v1/sse?sessionId=d2347cd	9:32:37
202	mcp.atlassian.com	POST	/v1/sse?sessionId=d2347cd	9:32:38

The request would look like

```
curl -H 'host: mcp.atlassian.com' \
     -H 'connection: keep-alive' \
     -H 'Authorization: Bearer
62*****************3:w**************8:S2************************vC' \
     -H 'mcp-protocol-version: 2025-06-18' \
     -H 'User-Agent: claude-code/2.0.10' \
     -H 'content-type: application/json' \
     -H 'accept: */*' \
     -H 'accept-language: *' \
     -H 'sec-fetch-mode: cors' \
     -H 'accept-encoding: br, gzip, deflate' \
     -H 'content-length: 54' \
     --data-binary '{"method":"notifications/initialized","jsonrpc":"2.0"}' \
     'https://mcp.atlassian.com/v1/sse?sessionId=d2347cd7-e470-4eb6-9a8c-63feb552dfe4'
```

The token appears to be composed of three colon-separated parts:

1. `62****************3` - Likely a user/client identifier
2. `w***********8` - Could be a session identifier or salt
3. `S2**********************vC` - Appears to be the main authentication token/signature

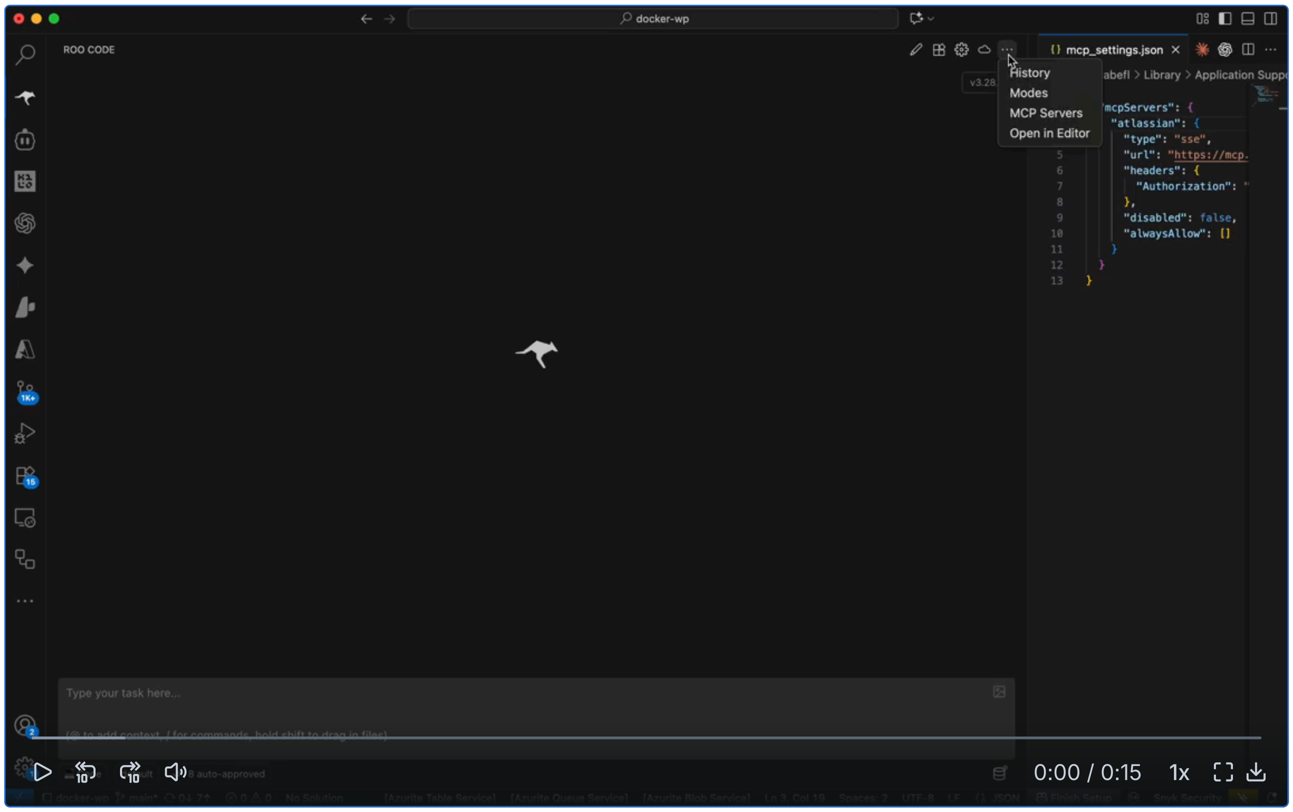
Let's give it to Roo Code

Add the Authorisation header to Roo following a conventional json formating.
- Three dots in the top right area for Roo Code
- MCP Servers / Edit Global MCP
- Add `"headers": {"Authorization": "Bearer Add:YOUR:Bearer"}`,

The end results should look like:

```
{
  "mcpServers": {
    "atlassian": {
      "type": "sse",
      "url": "https://mcp.atlassian.com/v1/sse",
      "headers": {
        "Authorization": "Bearer
62*****************3:w**************8:S2************************vC"
      },
      "disabled": false,
      "alwaysAllow": []
    }
  }
}
```

Is Roo happy ?



```
"mcpServers": {
  "atlassian": {
    "type": "sse",
    "url": "https://mcp.
    "headers": {
      "Authorization": "
    },
    "disabled": false,
    "alwaysAllow": []
  }
}
}
```

Yes, Roo is happy!

Okay that's not a roo, but I couldn't find a smiling roo...

You can now make use of Atlassian tools via the MCP server in Roo Code:

- atlassianUserInfo
- getAccessibleAtlassianResources
- getConfluenceSpaces
- getConfluencePage
- getPagesInConfluenceSpace
- getConfluencePageFooterComments
- getConfluencePageInlineComments
- getConfluencePageDescendants
- createConfluencePage
- updateConfluencePage
- createConfluenceFooterComment
- createConfluenceInlineComment
- searchConfluenceUsingCql
- getJiraIssue
- editJiraIssue
- createJiraIssue
- getTransitionsForJiraIssue
- transitionJiraIssue
- lookupJiraAccountId
- searchJiraIssuesUsingJql
- addCommentToJiraIssue
- getJiraIssueRemoteIssueLinks
- getVisibleJiraProjects
- getJiraProjectIssueTypesMetadata
- getJiraIssueTypeMetaWithFields
- search
- fetch

😢 *That is until your bearer token is invalidated or expires... then you can repeat the above and re-auth using Claude Code*

Withoutn surprise, the above methodology should work for most `HTTP` or `SSE` MCP endpoints ! (Not just Atlassian...)

🎉 *Helpful? Drop me a thanks on Achievers ! And if you've got knowledge to share, don't hold back - we all grow when we learn from each other* 💡