Git Repository & Jira MCP Connection Setup Guide

1. Git Repository Setup & Forking Instructions

Before proceeding with the MCP Connection Setup, ensure your project is connected to the correct Git repository.

Steps to Fork and Use the Repository:

1. Access the main repository link provided by your team (Bitbucket/GitHub).

2. Fork the repository by clicking the 'Fork' option (creates a copy under your account).

3. Clone the forked repository to your machine: git clone

4. Create a new branch for your feature/work: cd git checkout -b feature/

5. Make code changes, commit, and push to your forked repo: git add . git commit -m "Added MCP setup and code generation documentation" git push origin feature/

6. Create a Pull Request (PR) from your fork/branch to the main development branch to merge your changes.

2. Jira MCP Connection Setup Guide

This guide explains how to connect Jira with VS Code using the Atlassian MCP Server extension and enable automatic Jira story fetching and code generation.

Steps to Set Up the MCP Connection:

1. Install the 'Atlassian MCP Server' extension from the VS Code Marketplace.

2. Open Extension Settings and configure the required Atlassian-related links (Confluence/Jira base URLs, etc.).

3. Ensure mcp.json exists and configure company email, Atlassian API token, and base URL there. (Path: .vscode/mcp.json)

4. If mcp.json is not generated automatically, create it manually inside .vscode/mcp.json and add the necessary fields.

5. Install Docker Desktop and verify that containers can run on your machine.

6. Open the Command Palette (Ctrl + Shift + P) → 'Configure Tools' → Confirm MCP is listed and enable it.

7. Validate the setup by fetching Jira stories within VS Code, reviewing ticket content, and generating code snippets using MCP prompts.

3. Example mcp.json Configuration (fill with your values)

```
{
  "servers": {
    "mcp-atlassian": {
      "type": "stdio",
      "command": "docker",
      "args": [
        "run",
        "-i",
        "--rm",
        "-e", "CONFLUENCE_URL",
        "-e", "CONFLUENCE_USERNAME",
        "-e", "CONFLUENCE_API_TOKEN",
        "-e", "JIRA_URL",
        "-e", "JIRA_USERNAME",
        "-e", "JIRA_API_TOKEN",
        "ghcr.io/sooperset/mcp-atlassian:latest"
      ],
      "env": {
        "CONFLUENCE_URL": "your-org.atlassian.net",
        "CONFLUENCE_USERNAME": "your.email@company.com",
        "CONFLUENCE_API_TOKEN": "YOUR_API_TOKEN_HERE",
        "JIRA_URL": "your-org.atlassian.net",
        "JIRA_USERNAME": "your.email@company.com",
        "JIRA_API_TOKEN": "YOUR_API_TOKEN_HERE"
      }
    }
  }
}
```

4. Summary

• The Atlassian MCP Server connects with Jira through VS Code to enable story fetching and code generation.

5. Process Flow

1. MCP Setup – Initialize the MCP tool, configure settings, and connect.
2. Jira Connect – Authenticate and validate access to Jira/Confluence.
3. Story Fetch – Query the project and extract epics and stories.
4. Epic Analysis – Parse requirements, map features, and design APIs.
5. Code Generate – Generate files, add dependencies, and scaffold routes.
6. Validation – Run tests, verify APIs (Swagger), and confirm builds.