# Atlassian AI Assistant

🤖 **AI-powered code generation using Jira stories and Confluence documentation**

Transform your development workflow by connecting Jira requirements with Confluence team standards to generate intelligent, context-aware code.

## ✨ Features

### 🚀 **AI Code Generation from Jira Stories**

* Connect to your Jira instance and pull story details
* Generate TypeScript code based on story requirements
* Context-aware generation that follows your team’s coding standards

### 📚 **Confluence Integration**

* Automatically retrieves team coding standards from Confluence
* Incorporates API guidelines and architecture patterns
* Ensures generated code follows established team practices

### 🎯 **Smart Context Combination**

* Combines Jira story requirements with Confluence documentation
* Produces code that is both functionally correct and style-compliant
* Includes TODO comments for human review points

## 🛠️ Commands

| Command | Description |
| --- | --- |
| Atlassian AI: Generate Code from Jira Story | Generate code based on a Jira issue key |
| Atlassian AI: Show Demo | Display demo information and quick links |

## 📋 Requirements

* **VS Code** 1.101.0 or higher
* **Internet connection** for Atlassian API access
* **Atlassian Cloud account** with:
  + Jira project access
  + Confluence space access
  + Valid API token

## ⚙️ Setup

1. **Install the extension** (see Installation section below)
2. **Configure your Atlassian credentials** in the extension source
3. **Ensure you have demo content** in your Jira/Confluence instance
4. **Run the commands** via Command Palette (Ctrl+Shift+P)

## 🚀 Usage

1. **Open Command Palette** (Ctrl+Shift+P)
2. **Type “Atlassian”** to see available commands
3. **Select “Generate Code from Jira Story”**
4. **Enter a Jira issue key** (e.g., “SCRUM-1”)
5. **Watch the AI generate contextual code** based on your story and team standards

## 📦 Installation

### For Testing/Demo:

# Install from .vsix file  
code --install-extension atlassian-ai-assistant-0.0.1.vsix

### For Development:

# Clone and run  
git clone [repository]  
cd atlassian-ai-assistant  
npm install  
npm run compile  
# Press F5 in VS Code to launch Extension Development Host

## 🎬 Demo Scenario

This extension demonstrates a complete **Jira → Confluence → AI → Code** workflow:

1. **Jira Story**: “Implement OAuth authentication” (SCRUM-1)
2. **Confluence Standards**: Team coding patterns and API guidelines
3. **AI Generation**: TypeScript authentication service following team standards
4. **Result**: Production-ready code with proper error handling and documentation

## 🏗️ Architecture

* **Atlassian API Integration**: Direct REST API calls to Jira and Confluence
* **AI Context Building**: Combines multiple data sources for intelligent generation
* **VS Code Extension Framework**: Professional extension development patterns
* **TypeScript Implementation**: Type-safe development with modern patterns

## 🔒 Security Notes

* API tokens are currently stored in source code for demo purposes
* Production implementation would use secure credential storage
* All API calls use HTTPS and proper authentication

## 🎯 Use Cases

* **Rapid Prototyping**: Generate code scaffolding from requirements
* **Standards Compliance**: Ensure new code follows team patterns
* **Knowledge Integration**: Combine scattered documentation into actionable code
* **Developer Productivity**: Reduce time from story to implementation

## 🔮 Future Enhancements

* OpenAI integration for production-grade AI generation
* Secure credential management
* Multiple programming language support
* Custom team template integration
* Automated code review suggestions

## 📈 Strategic Value

This extension demonstrates how **organized knowledge architecture** enables superior AI experiences:

* **Atlassian’s integrated ecosystem** provides context that scattered tools cannot
* **Confluence documentation** becomes immediately actionable through AI
* **Jira requirements** are automatically translated into compliant code
* **Team productivity** increases through intelligent automation

## 📝 Release Notes

### 0.0.1 - Demo Release

* Initial implementation with Jira and Confluence integration
* AI-powered TypeScript code generation
* Demo scenario with authentication service generation
* Professional VS Code extension architecture

**Built for demonstrating the power of integrated knowledge platforms in AI-driven development workflows.**

## Following extension guidelines

Ensure that you’ve read through the extensions guidelines and follow the best practices for creating your extension.

* [Extension Guidelines](https://code.visualstudio.com/api/references/extension-guidelines)

## Working with Markdown

You can author your README using Visual Studio Code. Here are some useful editor keyboard shortcuts:

* Split the editor (Cmd+\ on macOS or Ctrl+\ on Windows and Linux).
* Toggle preview (Shift+Cmd+V on macOS or Shift+Ctrl+V on Windows and Linux).
* Press Ctrl+Space (Windows, Linux, macOS) to see a list of Markdown snippets.

## For more information

* [Visual Studio Code’s Markdown Support](http://code.visualstudio.com/docs/languages/markdown)
* [Markdown Syntax Reference](https://help.github.com/articles/markdown-basics/)

**Enjoy!**