

Configuration Guide

Configuration File

Settings are stored in `~/.config/ai-orchestrator/settings.json`

Default Configuration

```
{
  "mcpServerURL": "http://localhost:3000",
  "connectionTimeout": 30.0,
  "maxRetries": 3,
  "preferredModel": "claude-3-5-sonnet",
  "analysisModel": "gpt-4o",
  "fixModel": "claude-3-5-sonnet",
  "autoApplyFixes": false,
  "verifyFixesAfterBuild": true,
  "showDiffBeforeApply": true,
  "insertExplanationsAsComments": true,
  "testFramework": "XCTest",
  "documentationStyle": "swift-doc"
}
```

Configuration Options

Server Settings

Option	Type	Default	Description
<code>mcpServerURL</code>	String	<code>http://localhost:3000</code>	URL of the MCP server
<code>connectionTimeout</code>	Double	<code>30.0</code>	Connection timeout in seconds
<code>maxRetries</code>	Int	<code>3</code>	Maximum retry attempts for failed requests

AI Model Settings

Option	Type	Default	Description
preferredModel	String	claude-3-5-sonnet	Default AI model for code tasks
analysisModel	String	gpt-4o	Model for code analysis
fixModel	String	claude-3-5-sonnet	Model for generating fixes

Available Models:

- claude-3-5-sonnet - Best for code generation
- gpt-4o - Best for analysis
- gemini-2.5-flash - Fast and efficient
- moonshot-v1 - Code review

Behavior Settings

Option	Type	Default	Description
autoApplyFixes	Bool	false	Apply fixes without confirmation
verifyFixesAfter-Build	Bool	true	Rebuild after applying fixes
showDiffBeforeApply	Bool	true	Show diff preview before changes
insertExplanation-sAsComments	Bool	true	Add explanations as code comments

Code Generation Settings

Option	Type	Default	Description
testFramework	String	XCTest	Test framework for generated tests
documentationStyle	String	swift-doc	Documentation comment style

Test Frameworks:

- XCTest - Apple's native testing framework
- Quick/Nimble - BDD-style testing
- SwiftTesting - New Swift testing framework

Documentation Styles:

- `swift-doc` - Swift-style (`///` comments)
- `headerdoc` - Apple HeaderDoc format
- `doxygen` - Doxygen format

Environment Variables

The extension respects these environment variables:

```
export MCP_SERVER_URL="http://localhost:3000"
export AI_ORCHESTRATOR_LOG_LEVEL="info"
export AI_ORCHESTRATOR_TIMEOUT="30"
```

API Keys

API keys are stored securely in the macOS Keychain.

Setting Keys via Command Line

```
# OpenAI
security add-generic-password -a "openai" -s "com.debuggerlab.ai-orchestrator-xcode" -w "sk-...."

# Anthropic
security add-generic-password -a "anthropic" -s "com.debuggerlab.ai-orchestrator-xcode" -w "sk-ant-...."

# Google
security add-generic-password -a "google" -s "com.debuggerlab.ai-orchestrator-xcode" -w "AIza...".
```

Setting Keys via MCP Server

Keys can also be configured in the MCP server's `.env` file:

```
OPENAI_API_KEY=sk-....
ANTHROPIC_API_KEY=sk-ant-....
GEMINI_API_KEY=AIza...
```

Project-Specific Configuration

Create a `.ai-orchestrator.json` file in your project root:

```
{  
    "testFramework": "Quick/Nimble",  
    "documentationStyle": "swift-doc",  
    "excludePaths": [  
        "Pods/",  
        "Carthage/"  
    ],  
    "customPrompts": {  
        "refactor": "Refactor following our team's Swift style guide"  
    }  
}
```

Accessing Settings UI

1. In Xcode: **Editor > AI Orchestrator > Settings**
2. Or edit the JSON file directly

Resetting Configuration

```
# Reset to defaults  
rm ~/.config/ai-orchestrator/settings.json  
  
# Reinstall will create new defaults  
./Scripts/install.sh
```