

# 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
mcpServerURL	String	http://localhost:3000	URL of the MCP server
connectionTimeout	Double	30.0	Connection timeout in seconds
maxRetries	Int	3	Maximum retry attempts for failed requests

## AI Model Settings

Option	Type	Default	Description
<code>preferredModel</code>	String	<code>claude-3-5-sonnet</code>	Default AI model for code tasks
<code>analysisModel</code>	String	<code>gpt-4o</code>	Model for code analysis
<code>fixModel</code>	String	<code>claude-3-5-sonnet</code>	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
<code>autoApplyFixes</code>	Bool	<code>false</code>	Apply fixes without confirmation
<code>verifyFixesAfterBuild</code>	Bool	<code>true</code>	Rebuild after applying fixes
<code>showDiffBeforeApply</code>	Bool	<code>true</code>	Show diff preview before changes
<code>insertExplanation-asComments</code>	Bool	<code>true</code>	Add explanations as code comments

## Code Generation Settings

Option	Type	Default	Description
<code>testFramework</code>	String	<code>XCTest</code>	Test framework for generated tests
<code>documentationStyle</code>	String	<code>swift-doc</code>	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
```