





SuperClaude – Development Framework for Claude Code







License  [_ \(https://opensource.org/licenses/MIT\)](https://opensource.org/licenses/MIT) version  2.0.1
(<https://github.com/NomenAK/SuperClaude>)  issues 14 open
(<https://github.com/NomenAK/SuperClaude/issues>)  PRs welcome
(<https://github.com/NomenAK/SuperClaude/blob/master/CONTRIBUTING.md>)

SuperClaude Features

🔗 SuperClaude enhances Claude Code with:

- **18 Specialized Commands** covering development lifecycle tasks
- **9 Cognitive Personas** for domain-specific approaches
- **Token Optimization** with compression options
- **Evidence-Based Methodology** encouraging documentation
- **MCP Integration** with Context7, Sequential, Magic, Puppeteer
- **Git Checkpoint Support** for safe experimentation
- **Introspection Mode** for framework improvement and troubleshooting

v2.0.1 Installer Features:

-  **Update Mode:** Preserves customizations while updating
-  **Dry Run:** Preview changes before applying
-  **Smart Backups:** Automatic backup with timestamping
-  **Clean Updates:** Removes obsolete files
-  **Platform Detection:** Works with Linux, macOS, WSL
-  **Progress Tracking:** Installation feedback

Core Capabilities

Cognitive Personas (Now as Flags!)

Switch between different approaches with persona flags:

```
/analyze --code --persona-architect      # Systems thinking approach
/build --react --persona-frontend        # UX-focused development
/scan --security --persona-security      # Security-first analysis
/troubleshoot --prod --persona-analyzer  # Root cause analysis approach
```

⚡ 19 Commands

Development lifecycle coverage:

Development Commands

```
/build --react --magic --tdd      # Development with AI components
/dev-setup --ci --monitor         # Environment setup
/test --coverage --e2e --pup      # Testing strategies
```

Analysis & Quality

```
/review --quality --evidence --persona-qa    # AI-powered code review
/analyze --architecture --seq               # System analysis
/troubleshoot --prod --five-whys            # Issue resolution
/improve --performance --iterate            # Optimization
/explain --depth expert --visual             # Documentation
```

Operations & Security

```
/deploy --env prod --plan           # Deployment planning
/scan --security --owasp --deps      # Security audits
/migrate --dry-run --rollback        # Database migrations
/cleanup --all --validate            # Maintenance tasks
```

Token Efficiency

SuperClaude's [@include](https://github.com/include) (<https://github.com/include>) template system helps manage token usage: - **UltraCompressed mode** option for token reduction - **Template references** for configuration management - **Caching mechanisms** to avoid redundancy - **Context-aware compression** options

Example Workflows

Enterprise Architecture Flow

```
/design --api --ddd --bounded-context --persona-architect      # Domain-  
driven design  
/estimate --detailed --worst-case --seq                       # Resource  
planning  
/scan --security --validate --persona-security               # Security  
review  
/build --api --tdd --coverage --persona-backend              #  
Implementation
```

Production Issue Resolution

```
/troubleshoot --investigate --prod --persona-analyzer         # Analysis  
/analyze --profile --perf --seq                               # Performance  
review  
/improve --performance --threshold 95% --persona-performance # Optimization  
/test --integration --e2e --pup                               # Validation
```

Framework Troubleshooting & Improvement

```
/troubleshoot --introspect                                     # Debug  
SuperClaude behavior  
/analyze --introspect --seq                                    # Analyze  
framework patterns  
/improve --introspect --uc                                     # Optimize  
token usage
```

Full-Stack Feature Development

```
/build --react --magic --watch --persona-frontend            # UI  
development  
/test --coverage --e2e --strict --persona-qa                 # Quality  
assurance  
/scan --validate --deps --persona-security                    # Security  
check
```

Available Personas

Persona	Focus Area	Tools	Use Cases
architect	System design	Sequential, Context7	Architecture planning
frontend	User experience	Magic, Puppeteer, Context7	UI development
backend	Server systems	Context7, Sequential	API development
security	Security analysis	Sequential, Context7	Security reviews
analyzer	Problem solving	All MCP tools	Debugging
qa	Quality assurance	Puppeteer, Context7	Testing
performance	Optimization	Puppeteer, Sequential	Performance tuning
refactorer	Code quality	Sequential, Context7	Code improvement
mentor	Knowledge sharing	Context7, Sequential	Documentation

Configuration Options

Thinking Depth Control

```
# Standard analysis
/analyze --think

# Deeper analysis
/design --think-hard

# Maximum depth
/troubleshoot --ultrathink
```

Introspection Mode

```
# Enable self-aware analysis for SuperClaude improvement
/analyze --introspect

# Debug SuperClaude behavior
/troubleshoot --introspect --seq

# Optimize framework performance
/improve --introspect --persona-performance
```

Token Management

```
# Standard mode
/build --react --magic

# With compression
/analyze --architecture --uc

# Native tools only
/scan --security --no-mcp
```

Development (3 Commands)

- `/build` - Project builder with stack templates
- `/dev-setup` - Development environment setup
- `/test` - Testing framework

Analysis & Improvement (5 Commands)

- `/review` - AI-powered code review with evidence-based recommendations
- `/analyze` - Code and system analysis
- `/troubleshoot` - Debugging and issue resolution
- `/improve` - Enhancement and optimization
- `/explain` - Documentation and explanations

Operations (6 Commands)

- `/deploy` - Application deployment
- `/migrate` - Database and code migrations
- `/scan` - Security and validation

- `/estimate` - Project estimation
- `/cleanup` - Project maintenance
- `/git` - Git workflow management

Design & Workflow (5 Commands)

- `/design` - System architecture
- `/spawn` - Parallel task execution
- `/document` - Documentation creation
- `/load` - Project context loading
- `/task` - Task management

SuperClaude v2.0.1 – Development framework for Claude Code

★ [Star on GitHub \(https://github.com/NomenAK/SuperClaude\)](https://github.com/NomenAK/SuperClaude) | [🗨 Discussions \(https://github.com/NomenAK/SuperClaude/discussions\)](https://github.com/NomenAK/SuperClaude/discussions) | [🐛 Report Issues \(https://github.com/NomenAK/SuperClaude/issues\)](https://github.com/NomenAK/SuperClaude/issues)