

JDT As A Service

How The Eclipse Java Development Tools Can Benefit All IDEs

Roland Grunberg
Software Developer
IDE Services Team

There Is Demand For Java Support

13983 Results Showing: All categories Sort By: Installs

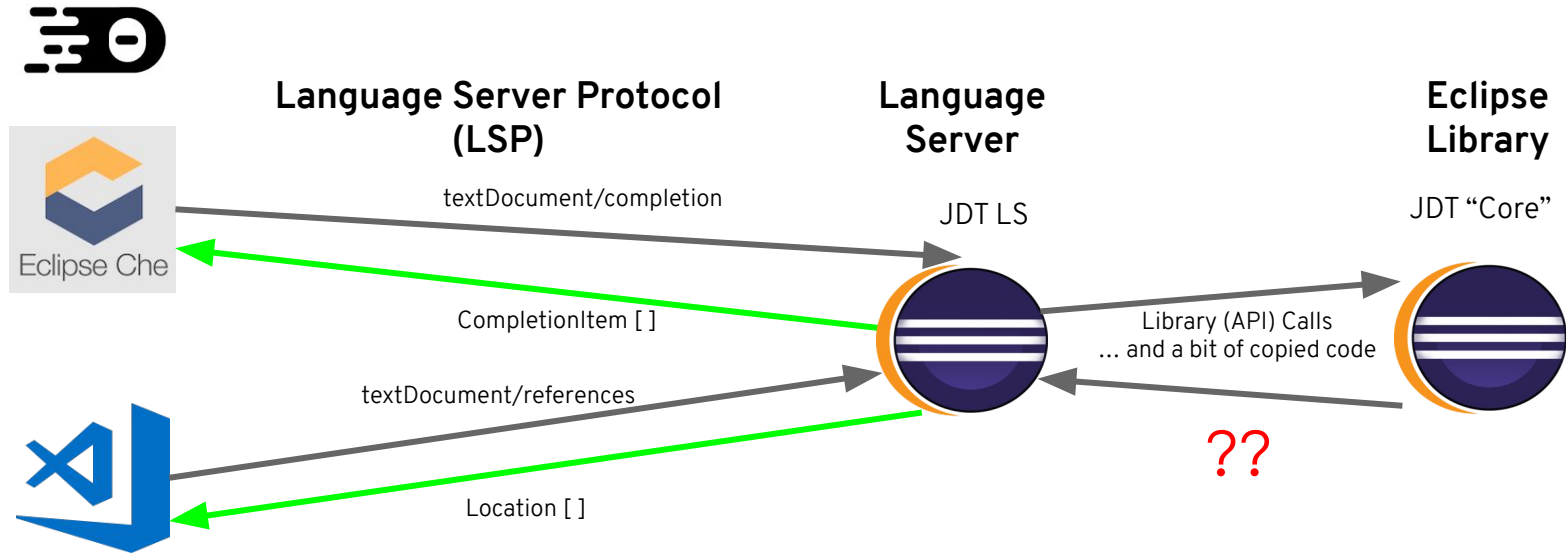
Total Installs: 538,312 Since Saturday, Sep 14, 2019

Rank	Title	Count
1	Darkest Dark Theme with DevStyle	43166
2	Spring Tools 4 - for Spring Boot (aka Spring Tool Suite 4)	31474
3	Subclipse	25878
4	SonarLint	17338
5	CodeMix 3	14351
6	Enhanced Class Decompiler	14067
7	PyDev - Python IDE for Eclipse	13380
8	Yaml Editor	13026
9	JBoss Tools	12513
10	Eclipse Java EE Developer Tools	12123
11	WindowBuilder	11398
12	TestNG for Eclipse	10222
13	e(f)xclipse	9089
14	Spring Tools 3 Add-On for Spring Tools 4	9082
15	Cucumber Eclipse Plugin	9044
16	Angular IDE	7608
17	Subversive - SVN Team Provider	7083
18	Eclipse Archive Utility	6473
19	Eclipse Web Developer Tools	6083
20	Maven Development Tools (requires m2e 1.6+)	6025
21	Groovy Development Tools	5762
22	Eclipse Tomcat Plugin	5557
23	Python: CodeMix	5536

Grid of extension cards (selected items are highlighted with red boxes):

- Python** (Microsoft, 12.3M, 5 stars, FREE)
- C/C++** (Microsoft, 7.5M, 4 stars, FREE)
- ESLint** (Dirk Baeumer, 6.3M, 5 stars, FREE)
- C#** (Microsoft, 5.4M, 4 stars, FREE)
- Debugger for Chrome** (Microsoft, 4.1M, 4 stars, FREE)
- Language Support for Java** (Red Hat, 3.9M, 4 stars, FREE)
- vscode-icons** (VSCode Icons Team, 3.8M, 5 stars, FREE)
- Prettier - Code formatter** (Esben Petersen, 3.7M, 4 stars, FREE)
- Beautify** (HookyQR, 3.6M, 4 stars, FREE)
- GitLens — Git supercharger** (Eric Amodio, 3.3M, 5 stars, FREE)
- Eno Snippets** (Eno Yao, 3.2M, 5 stars, FREE)
- Vetur** (Pine Wu, 3.1M, 5 stars, FREE)
- Docker** (Microsoft, 3M, 4 stars, FREE)
- Debugger for Java** (Microsoft, 3M, 5 stars, FREE)
- Maven for Java** (Microsoft, 2.9M, 5 stars, FREE)
- Live Server** (Ritwick Dey, 2.8M, 4 stars, FREE)
- Chinese (Simplified) Language Pack** (Microsoft, 2.8M, 5 stars, FREE)
- Java Test Runner** (Microsoft, 2.8M, 4 stars, FREE)

Quick Overview Of JDT-LS



How Does JDT-LS “use” JDT ?

- If JDT package has :
 - No dependence on `jdt.ui` → Use as a library (“internal” or API)
 - Dependence on `jdt.ui` → Copy code over, Refactor, Use

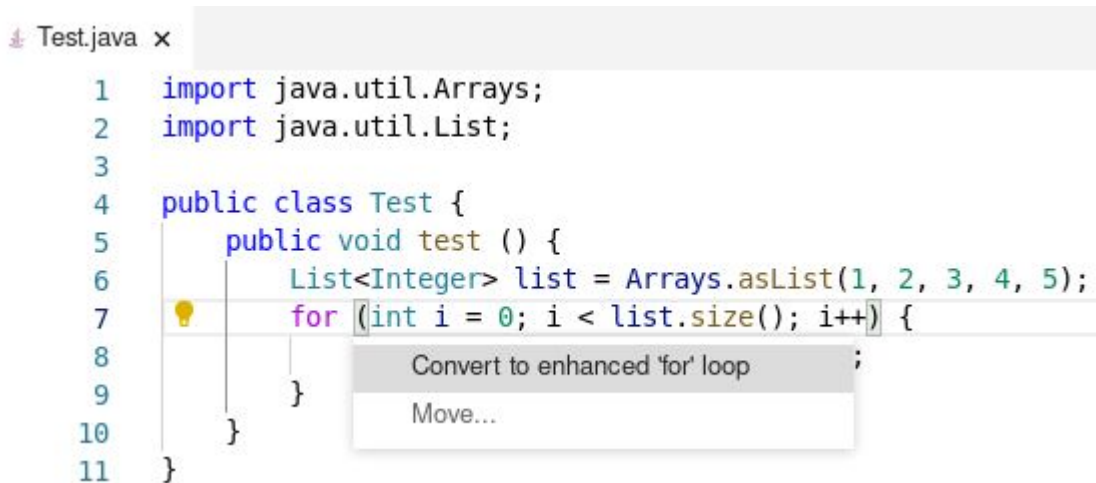
Why not just use API as-is and provide some kind of “dummy” display server ?

- How to package outside container environment across OS
- Increase in size and memory consumption

Quick Fixes, Assists, and Cleanup / Save Actions

- There are many of these that could be used
- Easier to migrate currently but still some more refactoring work sometimes required

```
Test.java x
1  import java.util.Arrays;
2  import java.util.List;
3
4  public class Test {
5      public void test () {
6          List<Integer> list = Arrays.asList(1, 2, 3, 4, 5);
7          for (int i = 0; i < list.size(); i++) {
8              ;
9          }
10     }
11 }
```



Why not just host features separately for users to install ?

- Users may not know about features
- Benefits to having features under one community

More Static Analysis (Spotbugs)

- There are well defined places to create quick/cleanup fixes but not for “arbitrary” problem markers
- Like SpotBugs (formerly FindBugs) but integrated into Editor and Problems View
 - It’s LGPL so at best can only reference as a 3rd party library
- Static detection would likely need to be re-written from scratch if using SpotBugs is no-go
 - <https://spotbugs.readthedocs.io/en/latest/bugDescriptions.html>
- Use `CompilationUnitEditor#reconciled(...)` as entry point

Chain Completion & Postfix Completion

```
InputStream in = URI.  
    }  
}
```

- create(String).getClass().getResourceAsStream(String) - 3 elements
- create(String).toURL().openStream() - 3 elements
- create(String).getAuthority().getClass().getResourceAsStream(String)

- Chain Completion (since 4.13)
 - Existing feature in Eclipse Code Recommenders project
 - Development stalled due to lack of contribution / maintenance

- Postfix Completion (new in 4.14 M1)
 - Initial patch in Bugzilla authored by Nicolaj Hoess
 - Contribution stalled due to limited resources

```
"File Not Found".var|  
    }  
}
```

- var - Creates a new variable

- Contribution mainly involved compliance with JDT API (Use proper ASTNode and IBinding API)

Asynchronous Completion (Authored By Mickael Istria)

- Moving completion proposal computations outside of the UI thread
- Patches already exist and are being prepared for contribution

Currently

- Long-running completion proposals will not freeze UI
- Visible improvement to responsiveness while typing

Planned

- Make it possible to enable even when some processors cannot run outside UI
- Allow Java proposals to populate completion popup in real-time

Issues To Consider

- Can uncover long-standing bugs with no easy solution (eg. Bug 550226)
- Time spent reviewing / improving contribution can add up
- More contributors needed
 - Improving our responsiveness on reviews (to those without commit rights)
 - Likely to get more contributions if we show we're willing to review
 - Pushing more manual workflow into CI instances for automation
 - `maven.compiler.failOnWarning`
 - “Rebase If Necessary” on Gerrit

Thank-You!

- Be sure to rate this talk !
- New for JDT 4.14 → <https://www.eclipse.org/eclipse/news/4.14/jdt.php>