

Code completion in ExtendJ using LSP

Daniel Tovesson

Why LSP?

- Programming language-specific features
- Just one language server needed
- Easier for vendors
- Growing support

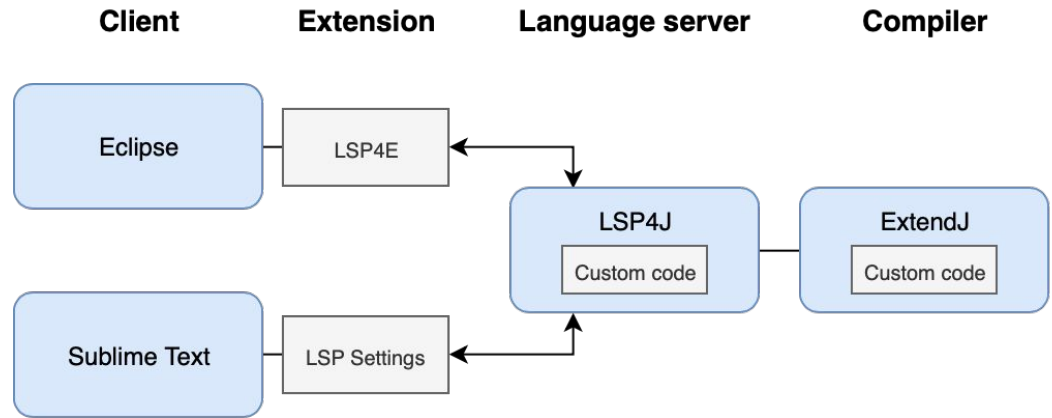


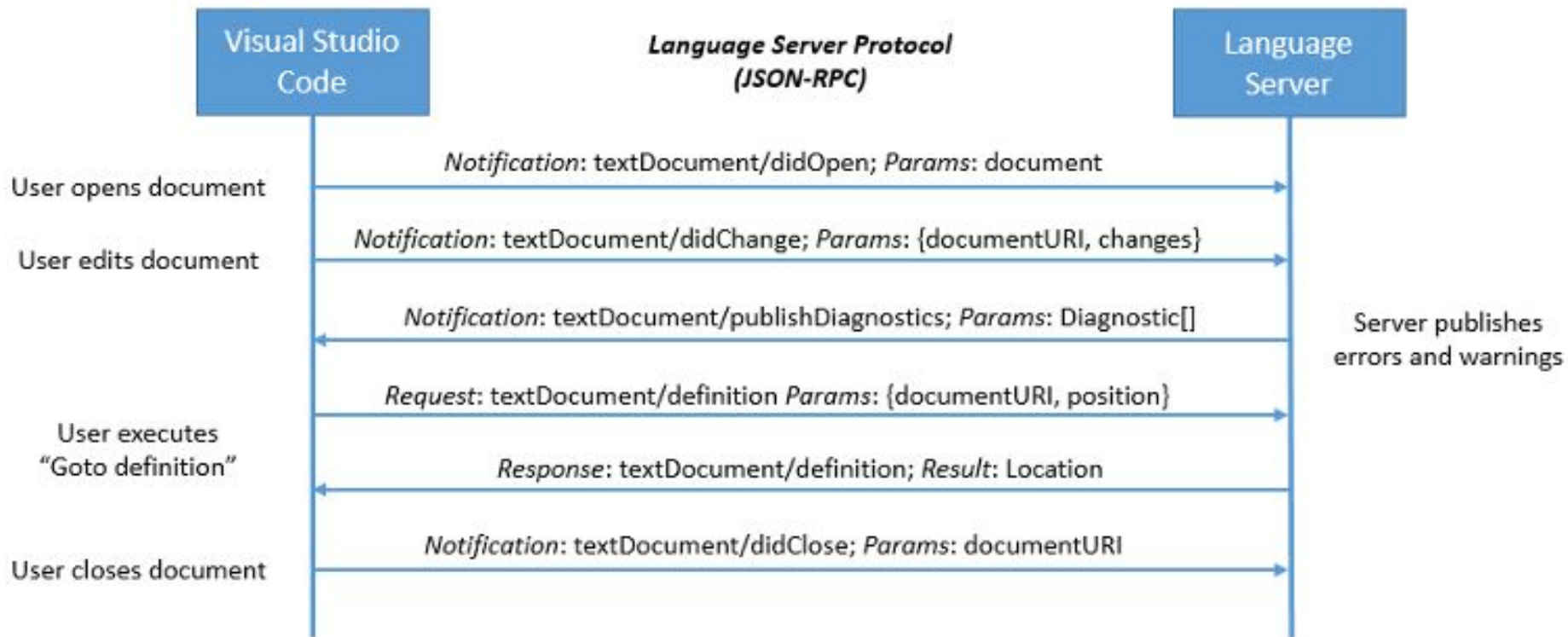
What is the Language Server Protocol?

- Messaging protocol between client and server
- Language server
- IDE plugin/extension
- JSON-based
- Flexible way to implement high-level support in an IDE
 - Code completion
 - Hover tooltips
 - Jump-to-definition
 - Diagnostics
 - and more...
- Developed by Microsoft

Adding a language server to your IDE

- Eclipse
 - About 20 lines of code
 - Java
- Sublime Text
 - About 20 lines of code
 - JSON
- Visual Studio Code
 - About 120 lines of code
 - TypeScript
- Vim
 - About 70 - 300 lines of code
 - Vim Script





```
{
  "jsonrpc": "2.0",
  "id" : 1,
  "method": "textDocument/definition",
  "params": {
    "textDocument": {
      "uri": "file:///path/file.java"
    },
    "position": {
      "line": 3,
      "character": 12
    }
  }
}
```

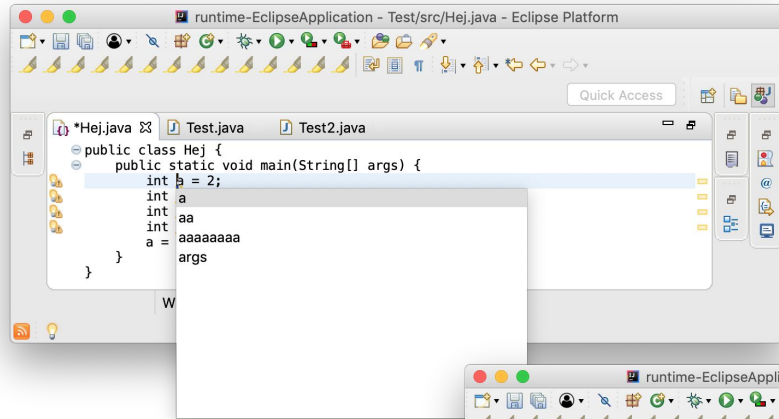
Background

- Based on EDAN70 (Language Server Protocol for ExtendJ)
- Error checking
- Eclipse and Sublime Text
- New feature: code completion

Implementation

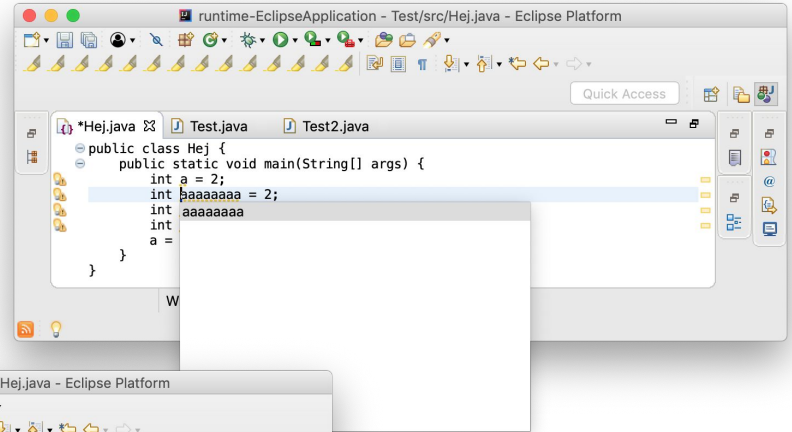
- Code completion in ExtendJ
 - Cursor position
 - Identifier
 - Scope
 - Variables
 - Filter
- Problems/limitations

Result



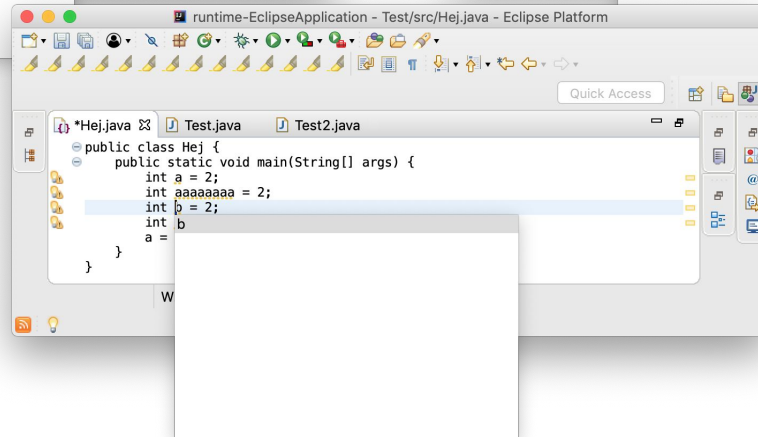
runtime-EclipseApplication - Test/src/Hej.java - Eclipse Platform

```
public class Hej {  
    public static void main(String[] args) {  
        int b = 2;  
        int a  
        int aa  
        int a = aaaaaaaa  
        args  
    }  
}
```



runtime-EclipseApplication - Test/src/Hej.java - Eclipse Platform

```
public class Hej {  
    public static void main(String[] args) {  
        int a = 2;  
        int aaaaaaaa = 2;  
        int aaaaaaaa  
        int a =  
    }  
}
```



runtime-EclipseApplication - Test/src/Hej.java - Eclipse Platform

```
public class Hej {  
    public static void main(String[] args) {  
        int a = 2;  
        int aaaaaaaa = 2;  
        int b = 2;  
        int b  
        a =  
    }  
}
```

Thank you for listening