How to set up antlr4 with intelliJ on windows 10:

      The version of antlr being used is: Antlr 4.7.1 (this version needs at least Java 7 to work)

      The version of intelliJ being used is: community version 2017.3.3

      The version of Java being used is: Java jdk 9.0.4

1. download intelliJ from the jetBrains website

2. Setup intelliJ, after setting up you should be prompted with this screen:

3. Before starting a new project you're going to want to download antlr4 and java, if your computer doesn't already have it.

4. To get antlr4, type "antlr4 plugin intelliJ" into your web browser, and the first link should be the plugins.jetbrains.com; if not look for that link then download it from there. You will need to unzip the file, and also remember the folder/directory you downloaded the zipped file to if it wasn't the downloads' folder/directory.

5. Now back to the intelliJ IDE, in the picture above you should notice at the bottom right, there is a configuration gear, click on it and select "plugins".

6. Click on the "Install Plugin from disk…", and type in the folder/directory that the zipped antlr4 file is sitting in, then select the zipped file:
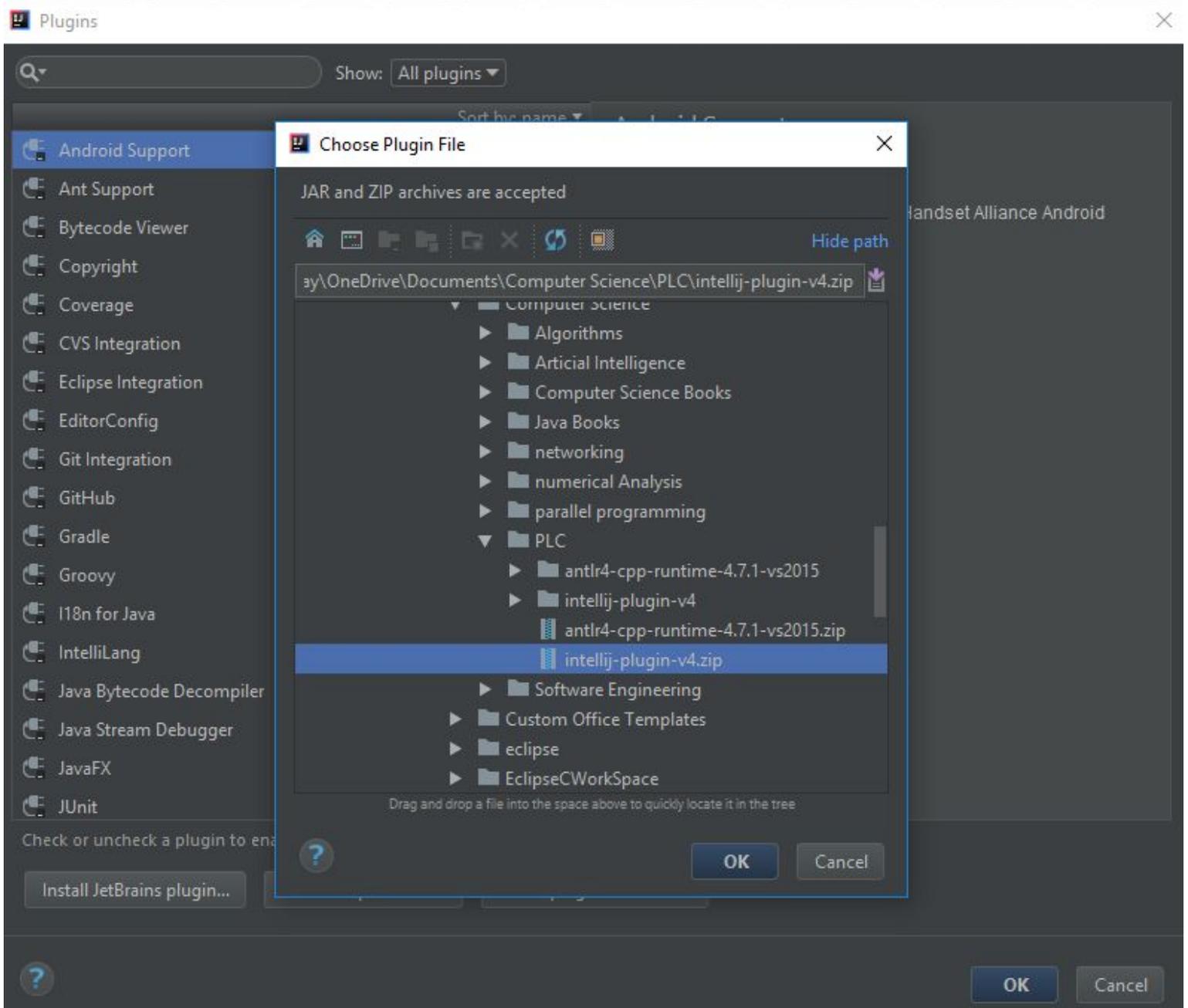
7. After clicking "OK", you will see that the plugin now appears in the plugins section and also with the antlr4 plugin selected you should see an option to restart intelliJ:

Plugins

Show: All plugins ▼

Sort by: name ▼

- Android Support ✓
- Ant Support ✓
- **ANTLR v4 grammar plugin** ✓
- Bytecode Viewer ✓
- Copyright ✓
- Coverage ✓
- CVS Integration ✓
- Eclipse Integration ✓
- EditorConfig ✓
- Git Integration ✓
- GitHub ✓
- Gradle ✓
- Groovy ✓
- I18n for Java ✓
- IntelliLang ✓
- Java Bytecode Decompiler ✓
- Java Stream Debugger ✓
- JavaFX ✓

ANTLR v4 grammar plugin

○ Restart IntelliJ IDEA

Version: 1.8.4

This plugin is for ANTLR v4 grammars and includes ANTLR 4.7. It works with Intellij 15, 2016.1-2017.3. It should work in other Jetbrains IDEs.

- syntax highlighting
- syntax error checking
- semantic error checking
- navigation window
- goto-declaration
- find usages
- rename tokens
- rename rules
- comment grammar rule lines with meta-/ (1.7)
- grammar/comment folding (1.7)
- generates parser code; shortcut (ctrl-shift-G / meta-shift-G) but it's in Tools menu and popups.
- code completion for tokens, rule names;
- finds tokenVocab option for code gen if there is a tokenVocab option, don't warn about implicit tokens.
- handles separate parsers and lectures like TParser.g4 and TLexer.g4 (1.7)
- Parse tree nodes show the alternative number the parser chose to match that node. (1.7)
- has live grammar interpreter for grammar preview. Right click on rule and say "Test ANTLR Rule".
- view parse trees in hierarchy (sideways tree) view. (1.8)
- can view parse trees for input matched in more than one

Check or uncheck a plugin to enable or disable it.

Install JetBrains plugin...   Browse repositories...   Install plugin from disk...
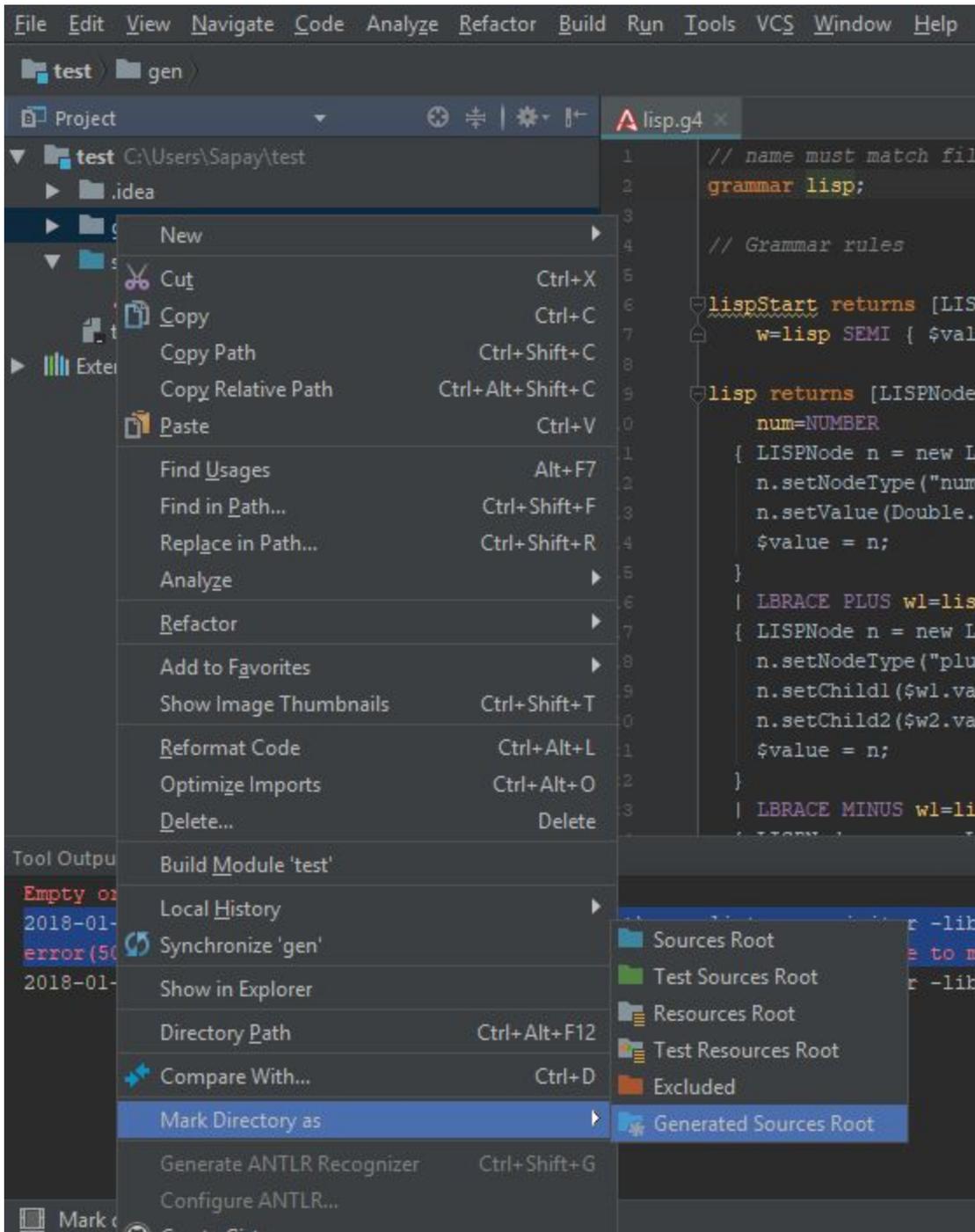
? OK Cancel

8. Restart intelliJ, now you are able to make a new project. Now on the new project screen, you should be able to select your java sdk (note this will most likely be in your C:\Program Files\Java folder/directory unless you changed the installation folder/directory), then click next until you get to the screen that asks you to name your project and save location. You should see something like this:
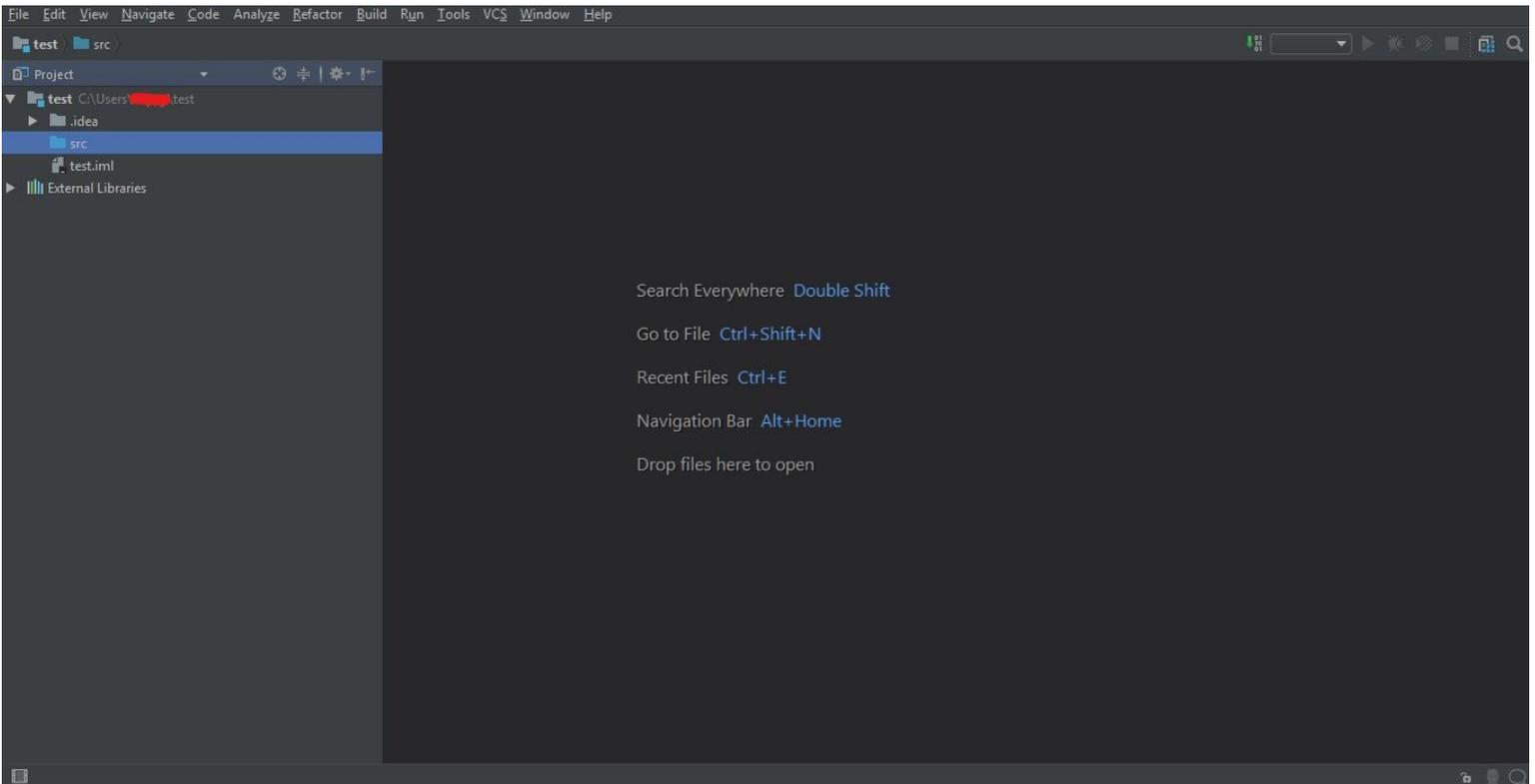
9. Right click src, and add a new a java class, then add a new file with the same name as the java class but add the extension ".g4". So for example test.g4, and make sure when you make the file you manually write the extension ".g4" as this will allow you to right click the file under the src folder/directory and select the option "Generate ANTLR Recognizer" (located at the bottom) AFTER writing the grammar. If you try to generate the antlr recognizer with a blank file IntelliJ will give you an error:

 Empty or bad grammar in file test.g4
2018-01-30 00:12:10: antlr4 -o C:/Users/Username/test\gen -listener -visitor -lib C:/Users/Username/test/src C:/Users/Username/test/src\test.g4
error(50): test.g4:1:0: syntax error: '<EOF>' came as a complete surprise to me

10.Once you have grammar that generates an antlr recognizer you should see a gen folder/directory appear in your project folder/directory, you will now want to right click on gen and at the bottom of the menu you should see an option "Mark Directory as" and then select the "Generated Sources root" option.

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

test > gen

Project

test C:\Users\Sapay\test
  .idea

New ▶
✂ Cut   Ctrl+X
📋 Copy   Ctrl+C
  Copy Path   Ctrl+Shift+C
  Copy Relative Path   Ctrl+Alt+Shift+C
📋 Paste   Ctrl+V

  Find Usages   Alt+F7
  Find in Path...   Ctrl+Shift+F
  Replace in Path...   Ctrl+Shift+R
  Analyze   ▶
  Refactor   ▶
  Add to Favorites   ▶
  Show Image Thumbnails   Ctrl+Shift+T
  Reformat Code   Ctrl+Alt+L
  Optimize Imports   Ctrl+Alt+O
  Delete...   Delete

Tool Output
  Build Module 'test'

Empty o
2018-01-   Local History   ▶
error(5   ⟳ Synchronize 'gen'
2018-01-   Show in Explorer

  Directory Path   Ctrl+Alt+F12

  Compare With...   Ctrl+D

  Mark Directory as   ▶   Sources Root
  Generate ANTLR Recognizer   Ctrl+Shift+G   Test Sources Root
  Configure ANTLR...   Resources Root
                                       Test Resources Root
                                       Excluded
                                       Generated Sources Root

```
// name must match fil
grammar lisp;

// Grammar rules

lispStart returns [LIS
    w=lisp SEMI { $val

lisp returns [LISPNode
    num=NUMBER
    { LISPNode n = new L
      n.setNodeType("num
      n.setValue(Double.
      $value = n;
    }
    | LBRACE PLUS wl=lis
    { LISPNode n = new L
      n.setNodeType("plu
      n.setChild1($wl.va
      n.setChild2($w2.va
      $value = n;
    }
    | LBRACE MINUS wl=li
```

11. Now click on the "File" option in the top menu, and select "project structure", or you could use the shortcut ctrl+atl+shift+s. Then under the "modules" section you want to go to the "dependencies" tab add a new module (the green plus button) and select the "JARs or directories…" option:
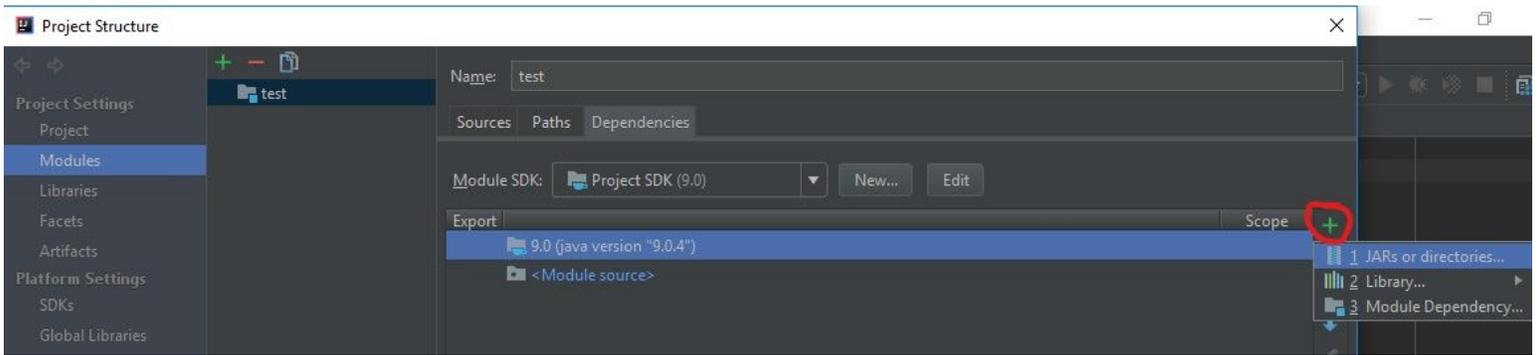
12. Last go back to the folder/directory you unzipped the antlr4 plugin, and go into the folder/directory until you find the JAR files and select the complete version; apply the module and leave the "scope" to compile.
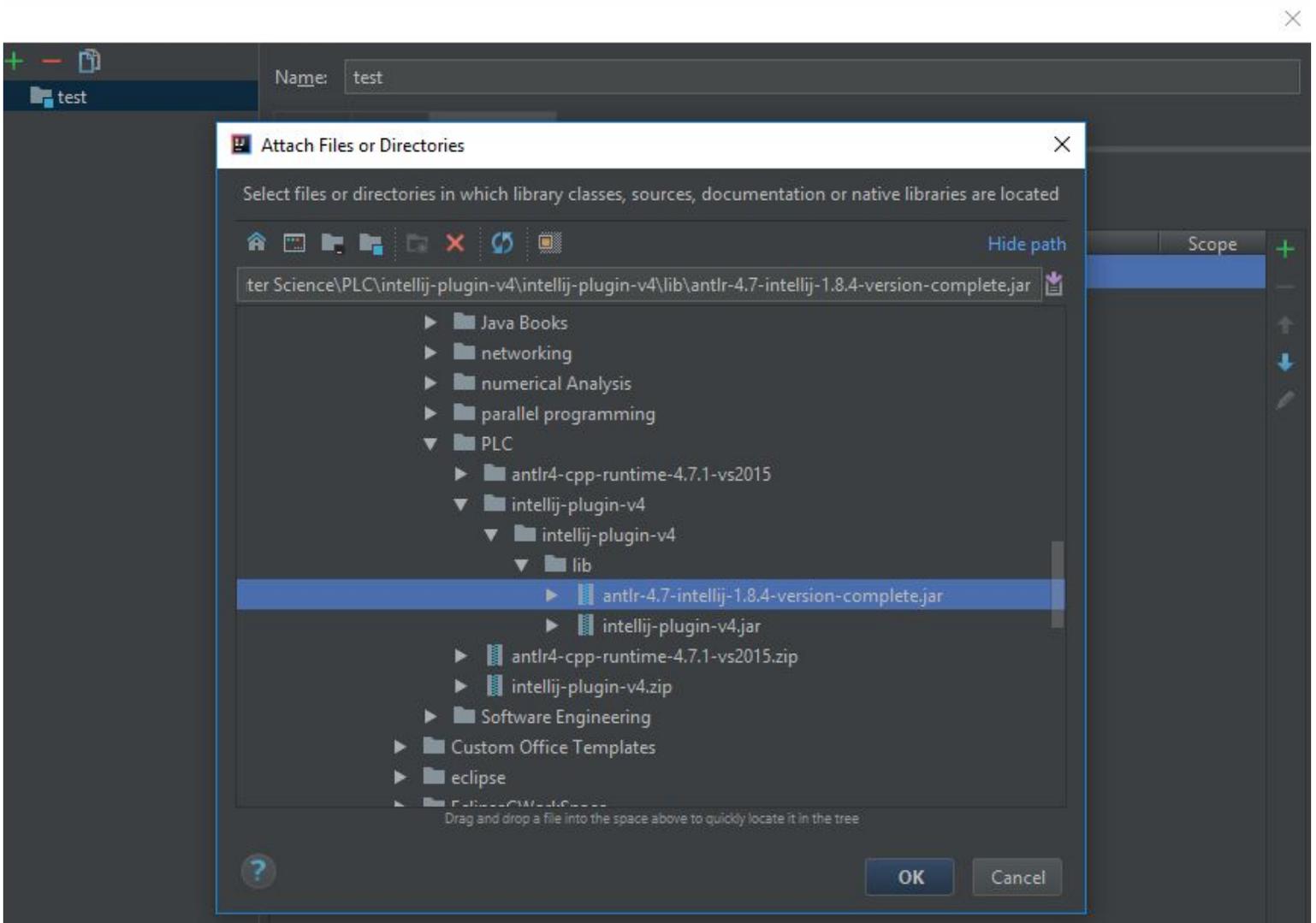
Now all you're java classes written using the antlr grammar will work.