**LEARNING ANTLR**

Contents

[**How to use ANTLR?** 3](#_Toc474860857)

[**Examples of ANTLR** 4](#_Toc474860858)

[**1.** **Sample Hello Example** 4](#_Toc474860859)

[**2.** **Grammar file for Dockerfile** 8](#_Toc474860860)

[**References** 11](#_Toc474860861)

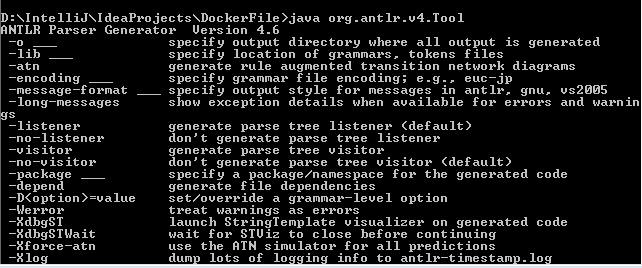
ANTLR is two things: a tool that translates your grammar to a parser/lexer in Java (or other target language) and the runtime needed by the generated parsers/lexers. Even if you are using the ANTLR Intellij plug-in or ANTLRWorks to run the ANTLR tool, the generated code will need the runtime library.

# **How to use ANTLR?**

1. Download antlr-4.5.3-complete.jar(or whatever version) from <http://www.antlr.org/download/> . Save in any local directory.
2. Add antlr-4.5-complete.jar to CLASSPATH, SET CLASSPATH=.;C:\JavaLib\antlr-4.5.3-complete.jar.
3. Create short convenient commands for the ANTLR Tool, and TestRig, using batch files.

* Create Batch files antlr4.bat and grun.bat
* java org.antlr.v4.Tool %\* -: antlr.bat
* java org.antlr.v4.gui.TestRig -: grun.bat

1. Testing for installation -: launch org.antlr.v4.Tool, java org.antlr.v4.Tool OR you can use java –jar option to launch the jar.



Check the link <https://github.com/antlr/antlr4/blob/master/doc/getting-started.md>, for getting started with antlr tutorial.

# **Examples of ANTLR**

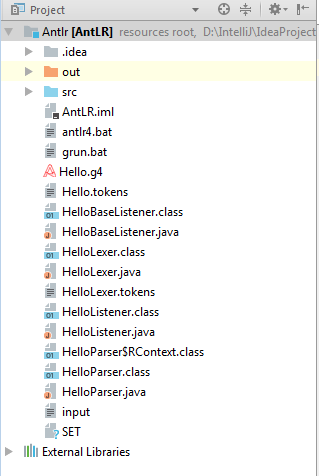
### **Sample Hello Example**

1. Create a grammar file Hello.g4,

*// Define a grammar called Hello***grammar** Hello;  
r : **'Hello'** ID NUMBER ; *// match keyword hello followed by an identifier*ID : [a-zA-Z]+ ;*// match lower-case or upper case identifiers*NUMBER: (**'0'**..**'9'**)+; //match the digits from 0 to 9.  
WS : [ \t\r\n]+ -> skip ; *// skip spaces, tabs, newlines*

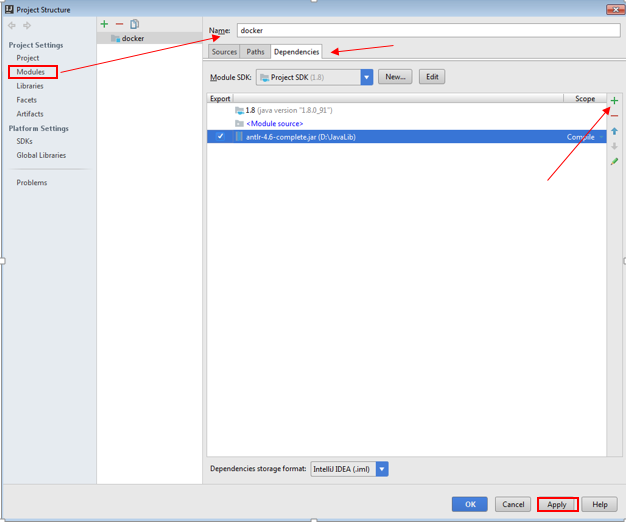
1. Run antlr tool on it (on cmd in same directory where you have your grammar file):

* Antlr4 Hello.g4 -: this will generate .java file corresponding to g4 file.
* Javac Hello\*.java-: this will compile .java files and generate class files.

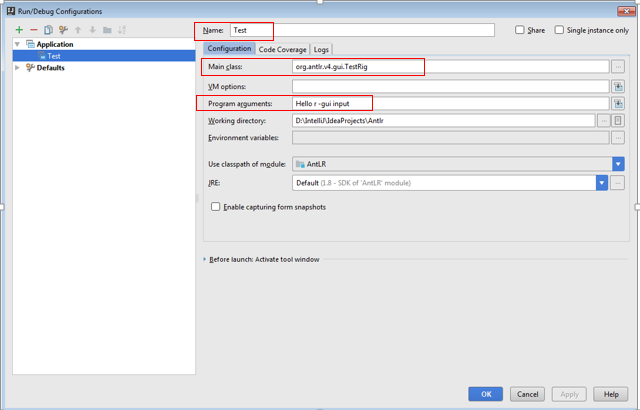


1. In IntelliJ IDE, you need to add the antlr-xxx-complete.jar in the module dependencies path as shown, Go to File 🡺 Project Structure.

* Click on modules, give a name.
* Got to dependencies, Add jar, and add path of antlr jar file.
* Apply and OK.

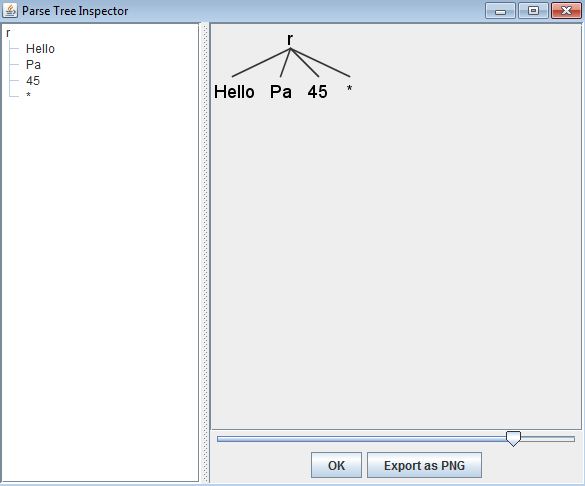


1. To test the grammar file using input, in IntelliJ Idea IDE, go to Run 🡺 Edit configurations

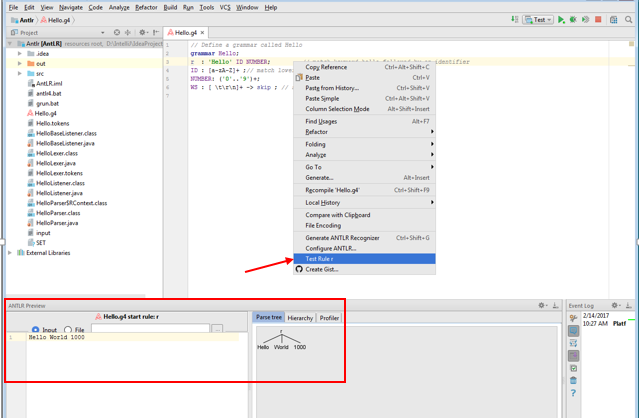


* Give name to application.
* In main class-: Give TestRig (it will load the class itself).
* In program arguments-: grammar\_file\_name rule\_name –rule\_name input\_file\_name
* Click Apply and Ok.

1. Go to Run🡺 Run Test(Application name as defined in above step), the ouput is as-:



1. To test any rule, right click on **rule** you have specified, as shown, give some inputs or you can browse a input file, and examine the parse tree.



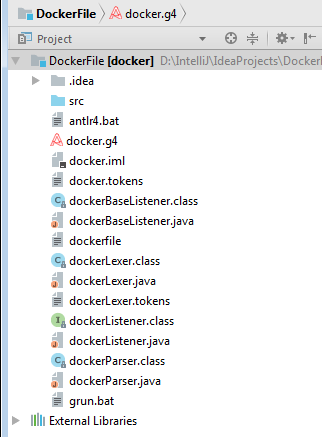
### **Grammar file for Dockerfile**

1. Create a grammar file docker.g4,

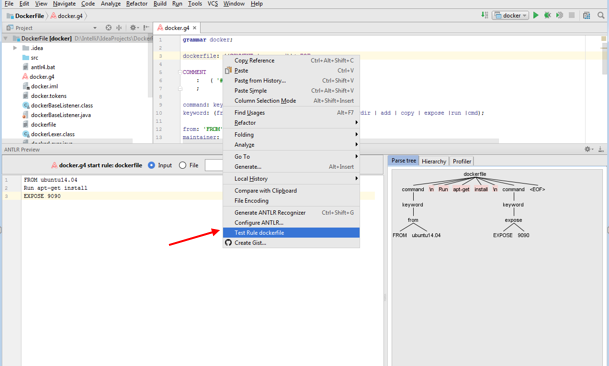
**grammar** docker;  
  
dockerfile: ((COMMENT | command))+ EOF; //rule   
  
COMMENT  
 : ( **'#'** ~[\r\n]\* **'\r'**? **'\n'**) -> skip //should skip this  
 ;  
  
command: keyword;  
keyword: (from | env | entrypoint | maintainer | workdir | add | copy | expose |run |cmd); //commands used in dockerfile  
  
from: **'FROM'** ANYKEYS;   
maintainer: **'MAINTAINER'** ANYKEYS ANYKEYS;  
env: **'ENV'** ANYKEYS ;  
entrypoint: **'ENTRYPOINT'** ANYKEYS;  
workdir: **'WORKDIR'** ANYKEYS;  
add: **'ADD'** src dest;  
copy: **'COPY'** src dest;  
src: ANYKEYS | **'.'**;  
dest: ANYKEYS | **'.'**;  
expose: **'EXPOSE'** NUMBER;  
run: **'RUN'** body NEWLINE;  
  
body: shellCmd (SHELLAND shellCmd)\* ;  
shellCmd: ANYKEYS+;  
SHELLAND: **'&&'** | (**'\\'** NEWLINE **'&&'**) | **'='** | **';'**;  
cmd: **'CMD'** body;  
  
  
NUMBER: [0-9]+; //any number from 0 to 9  
LETTER: [a-zA-Z]; // letters lowercase or uppercase  
SYMBOL: (**':'** | **'\_'** | **'-'** | **'/'** | **'|'** | **'"'** | **'='** | **'\*'** | **'\\'** | **'\''** | **'+'** | **']'** | **'['** | **'{'** | **'}'** | **';'** | **'!'** | **'~'** | **'.'** | **'–'** | **'$'** | **'<'** | **'>'** | **'@'** | **','**)+;  
ANYKEYS: (LETTER | NUMBER | SYMBOL)+;  
  
NEWLINE: (**'\n'** | **'\r'**)+;  
WS : ((**' '** | **'\t'**)+) -> skip;

1. Run antlr tool on it:

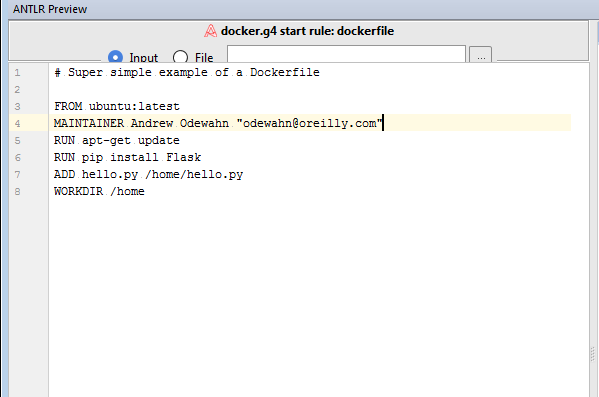
* Antlr4 docker.g4 -: this will generate .java file corresponding to g4 file.
* Javac docker\*.java-: this will compile .java files and generate class files.



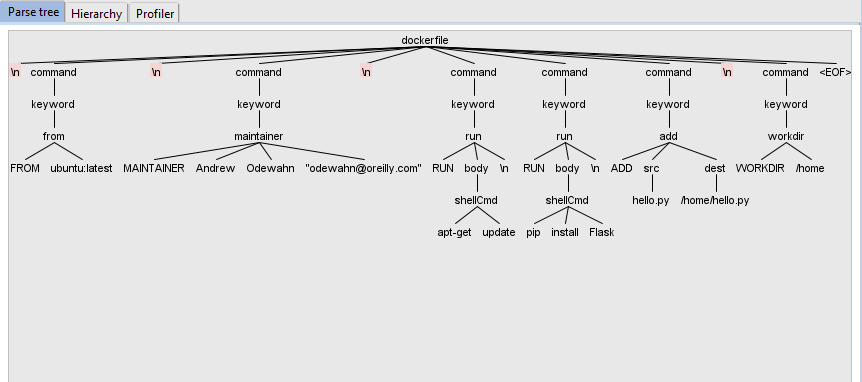
1. To test the rules from grammer file,



* If following Input is give-:



* The parse tree is as shown-:



# **References**

1. To check about more grammar file of XML, HTML, Java, C etc, you can check the link <https://github.com/antlr/grammars-v4>.