Github 地址：<https://github.com/antlr/antlr4/blob/master/doc/getting-started.md>

**ANTLR4**：把编程语言的语法规则转换成相对应的词法/语法分析器（lexer/parser）以及词法/语法分析器所需要的运行时库

## Windows下：

1. JDK1.6及以上
2. 从 <http://www.antlr.org/download/> 下载对应的版本（如antlr-4.4.-complete.jar，另eclipse里的antlr-v4 插件为4.4），保存到第三方库目录下（任意目录，例如：C:\java）
3. 添加 antlr-4.4.-complete.jar（你下载的版本）到CLASSPATH路径（也可以临时在命令行添加：SET CLASSPATH=.;C:\Javalib\antlr-4.7.1-complete.jar;%CLASSPATH%）
4. 为ANTLR4 Tool 创建简短的命令行：（可以使用doskey命令创建）

在命令行中输入：

doskey antlr4=java org.antlr.v4.Tool $\*

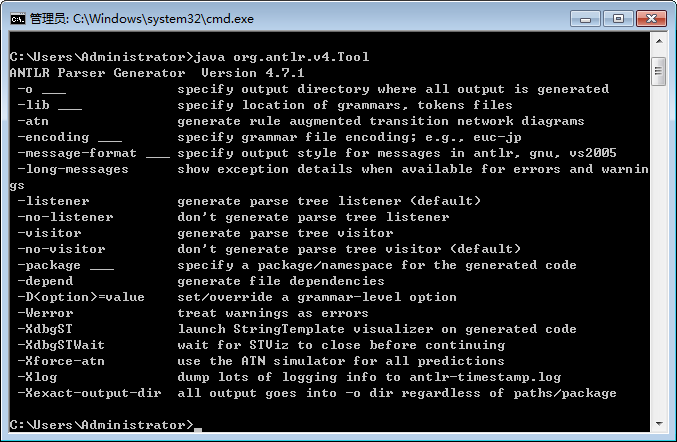
doskey grun=java org.antlr.v4.gui.TestRig $\*

### 测试

命令行输入：

Java org.antlr.v4.Tool

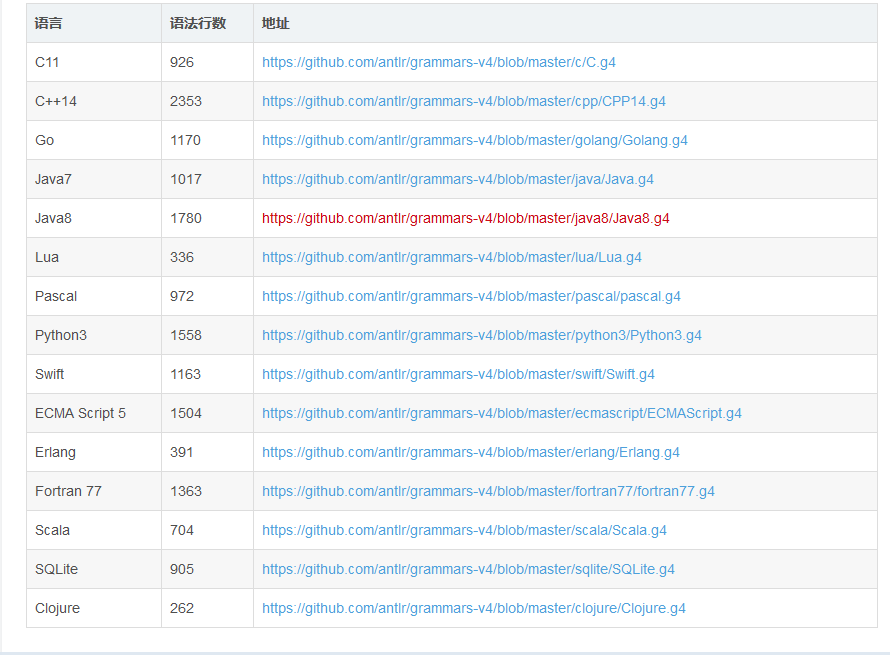
或者你设定的 antlr4



例子:（语法为java的语法规则，同时生成.java文件）：

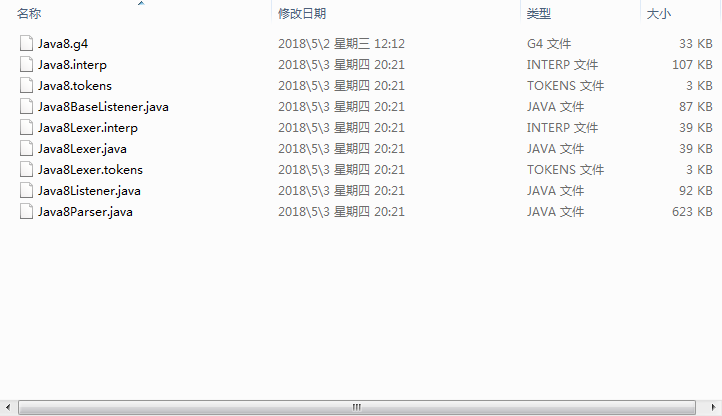
语法规则下载地址：<https://github.com/antlr/grammars-v4> 此地址中包含C\C++\Python\Java等语言的语法规则，

另，<https://blog.csdn.net/lusing/article/details/60869930> 此地址可查询不同语法规则对应的文件



此处以java8语法规则为例：

1. 到/grammars-v4/java8/目录下找到Java8.g4文件，此文件即包含对应的语法规则
2. 建立一个临时目录用来测试，如c:\java\javatest
3. 将java8.g4文件拷贝到c:\java\javatest目录下
4. 命令行进入目录下cd c:\java\javatest
5. 如果已经配置好antlr4命令，则直接在命令行输入antlr4 Java8.g4，即可看到此文件夹下生成了多个.java文件(注意大小写)



例子2：（语法为java的语法规则，同时生成.py文件）：

<https://github.com/antlr/antlr4/blob/master/doc/python-target.md> 可以适当查阅此网址

前几步操作相同，基本区别就在于最后的命令行的输入

1. 到/grammars-v4/java8/目录下找到Java8.g4文件，此文件即包含对应的语法规则
2. 建立一个临时目录用来测试，如c:\java\javatest
3. 将java8.g4文件拷贝到c:\java\javatest目录下
4. 命令行进入目录下cd c:\java\javatest
5. 如果已经配置好antlr4命令，则直接在命令行输入antlr4 -Dlanguage=Python3 Java.g4(注意大小写)，即可看到此文件夹下生成了多个.py 文件



另，为获取java代码的token流，写了一个对应的JavaTemplate.py文件，放于同级目录之下。

import antlr4

from Java8Lexer import Java8Lexer

import re

def parseJava(code):

code = code.replace('\\n', '\n')

parsedVersion = []

stream = antlr4.InputStream(code)

lexer = Java8Lexer(stream)

toks = antlr4.CommonTokenStream(lexer)

toks.fetch(500)

identifiers = {}

identCount = 0

for token in toks.tokens:

if token.type == 109:

parsedVersion += ["CODE\_INTEGER"]

elif token.type == 111:

parsedVersion += ["CODE\_REAL"]

elif token.type == 112:

parsedVersion += ["CODE\_CHAR"]

elif token.type == 113:

parsedVersion += ["CODE\_STRING"]

elif token.type == 9 or token.type == 7 or token.type == 6: # whitespace and comments and newline

pass

else:

parsedVersion += [str(token.text)]

return parsedVersion

if \_\_name\_\_ == '\_\_main\_\_':

print (parseJava("public class a {public static void main(String[] args) { //VariableDeclarationStatement \n String s = \"abc\"; s.toCharArray();//EmptyStatement \n}}"))

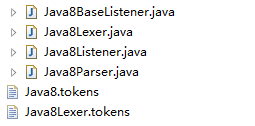
运行时会报错（具体原因暂时不清楚），将JavaLexer.py文件下从

def sempred(self, localctx:RuleContext, ruleIndex:int, predIndex:int):开始全部注释掉即可运行成功。

注：以上为在命令行执行对应的命令

## Antlr4在Eclipse下的使用：

1. Eclispe->Help->Eclipse Marketplace 搜索antlr4，然后安装
2. Eclipse->Help->Eclipse Marketplace 搜索 Xtext2.7.3，然后安装
3. 安装完成，重启Eclipse
4. 新建java project
5. 将对应语言的语法规则文件targetLanguage.g4放入项目根目录下，然后在eclipse中fresh项目
6. 找到相应的.g4文件，然后，右击，Run As->Gengrate ANTLR Organizer，会生成对应的target文件夹，此文件夹下包含生成的词法语法分析器文件等。



1. 右击项目->properties->java build path->source框下，选择 add folder，将target文件夹下的子文件夹antlr4选中，点击确定，然后项目会报错
2. 将下载的antlr-4.4-complete.jar包，放到项目根目录下的文件夹lib中（lib如果不存在，则新建lib），然后fresh项目
3. 在properties->java build path->libraries框下，选择add jars，将 antlr-4.4-complete.jar包添加一下，报错消失。
4. 新建一个package，如src， 以及一个用于测试的.java文件，如Test.java.
5. 到grammars-v4/java8 复制Test.java文件到之前建立的Test.java，进行修改替换
6. 修改Test.java 的main方法，如下：

**public** **static** **void** main(String[] args) {

String []files=**new** String[1];

// files[0]="C:\\Users\\Administrator\\Desktop\\ai\\AbstractAction.java";

files[0]="src/HelloWorld.java";

*doAll*(files);

}

1. 修改Test.java的成员变量 gui，将false改成true，即可实现最后语法树结果的可视化