무료 전자 책

배우기

ANTLR

Free unaffiliated eBook created from
Stack Overflow contributors.

#antlr

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: antlr

It is an unofficial and free ANTLR ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ANTLR.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: ANTLR

ANTLR ( ) , , , . , . , ANTLR .

- antlr ( )

Antlr
Antlr ( ) . antlr V1.V2.V3 .

- V1 : V1 .
- V2 : V2 (:)
- V3 : .

Antlr Java . antlr . ( ) .

-
- #
- (2 3)
-

2.0	1997-05-01
3.0	2011 1 19
4.0	2013-01-21
4.1	2013-07-01
4.2	2014-02-05
4.2.1	2014-03-25
4.2.2	2014-04-07
4.3	2014-06-19
4.4	2014-07-16
4.5	2015-01-23
4.5.1	2016-07-16
4.5.2	2016-01-30
4.5.3	2016-03-31

4.6	2016-12-15
4.7	2017-03-30

Examples

hello world .

```
// define a grammar called Hello
grammar Hello;
r   : 'hello' ID;
ID  : [a-z]+ ;
WS  : [ \t\r\n]+ -> skip ;
```

.g4 / .

```
Java -jar antlr-4.5.3-complete.jar Hello.g4

//OR if you have setup an alias or use the recommended batch file

antlr4 Hello.g4
```

Hello.g4 .

1. Hello.tokens
2. HelloBaseListener.java
3. HelloLexer.java
4. HelloLexer.tokens
5. HelloListener.java
6. HelloParser.java

ANTLR jar . Java .

```
javac *.java
```

ANTLR : https://riptutorial.com/ko/antlr/topic/4453/antlr-

2: ANTLR v3

Examples

Eclipse ANTLR

(Indigo and ANTLR IDE 2.1.2)

1. Eclipse .
2. ANTLR v2 ANTLR jar . temp . antlr-nn (: Eclipse) .
3. Eclipse ANTLR IDE .
   - Eclipse Help Install New Software .
   - ... .
   - http://antlrv3ide.sourceforge.net/updates ANTLR IDE .
   - ANTLR IDE vn.nn . Eclipse .
4. ANTLR IDE .
   - Eclipse .
   - ANTLR .
   - ... .
   - ANTLR ... antlr-nn .
   - ANTLR .
   - . . : antlr-java antlr-generated.
   - Building General -nfa -dfa . ANTLR Java .
   - .
5. Java ANTLR .
   - Eclipse File, New, Java Project . .
   - ANTLR ( ) , ANTLR .
   - ANTLR jar . , Java , JAR ... ANTLR jar . .
6. ANTLR .
   - ANTLR : src , , , ANTLR . .
   - ".g" . language = Java, @header, @lexer :: header @members ( ). (Ctrl +).
7. .
   - Java . Building General ANTLR Preferences -nfa -dfa (4g). [: CLASSPATH Eclipse (32 64) Java7 Windows Java7 SDK .]
   - " "Java Java , .

```
grammar test; //must match filename.g

options {
    language = Java;
}

@header { //parser
    package pkgName; //optional
    import java.<whatever you need>.*;
}
```

```
@members { //parser
    // java code here
}

@lexer::header { //lexer
    package pkgName; //optional
    import java.<whatever you need>.*;
}

@lexer::members {
    // java code here
}
/*------------------------------------------------------------------
 * PARSER RULES (convention is all lowercase)
 *------------------------------------------------------------------*/
parserule: LEXRULE;

/*------------------------------------------------------------------
 * LEXER RULES (convention is all uppercase)
 *------------------------------------------------------------------*/
LEXRULE: 'a'..'z';
```

ANTLR v3 : https://riptutorial.com/ko/antlr/topic/6629/antlr-v3-

3: ANTLR v4

ANTLR v4 / . ANTLR (, AST) . . ANTLR v4 Java, C #, JavaScript, Python2 Python3 . C ++ . GUI IDE Visual Studio, Intellij, NetBeans Eclipse .

ANTLR . ANTLR Terrence Parr (ANTLR) ANTLR 4 Reference .

- 4.5 : 01/22/15 - JavaScript C # . 4.5
- 4.4 : 07/16/14 - Python2 Python3 . 4.4
- 4.3 : 06/18/14 - . . 4.3
- 4.2 : 02/04/14 - / . 4.2
- 4.1 : 06/30/13 - . AST PNG . 4.1
- 4.0 : 01/21/13 - .

Examples

ANTLR Java Jar . . ANTLR jar Java . .

ANTLR JAR JAR ANTLR .

```
Java -jar antlr-4.5.3-complete.jar
```

antlr-4.5.3-complete.jar .

.

```
ANTLR Parser Generator  Version 4.5.3
 -o ___              specify output directory where all output is generated
 -lib ___            specify location of grammars, tokens files
 -atn                generate rule augmented transition network diagrams
 -encoding ___       specify grammar file encoding; e.g., euc-jp
 -message-format ___ specify output style for messages in antlr, gnu, vs2005
 -long-messages      show exception details when available for errors and warnings
 -listener           generate parse tree listener (default)
 -no-listener        don't generate parse tree listener
 -visitor            generate parse tree visitor
 -no-visitor         don't generate parse tree visitor (default)
 -package ___        specify a package/namespace for the generated code
 -depend             generate file dependencies
 -D<option>=value    set/override a grammar-level option
 -Werror             treat warnings as errors
 -XdbgST             launch StringTemplate visualizer on generated code
 -XdbgSTWait         wait for STViz to close before continuing
 -Xforce-atn         use the ATN simulator for all predictions
 -Xlog               dump lots of logging info to antlr-timestamp.log
```

.

```
     1. Add antlr4-complete.jar to CLASSPATH, either: Permanently:
Using System Properties dialog > Environment variables > Create or append to CLASSPATH
```

```
variable Temporarily, at command line: SET CLASSPATH=.;C:\Javalib\antlr4-
complete.jar;%CLASSPATH%
     3.Create batch commands for ANTLR Tool, TestRig in dir in PATH
         antlr4.bat: java org.antlr.v4.Tool %*
         grun.bat:   java org.antlr.v4.gui.TestRig %*
```

.g4 .

```
Java -jar antlr-4.5.3-complete.jar yourGrammar.g4
```

-Dlanguage . C # .

```
java -jar antlr-4.5.3-complete.jar yourGrammar.g4 -Dlanguage=CSharp
```

.

ANTLR .

Maven, Gradle ( ) `org.antlr:antlr4-runtime` .

- - Maven : `org.antlr:antlr4` .

Eclipse Hello World

(ANTLR 4.5.3, Eclipse Neon, ANTLR 4 IDE 0.3.5 Java 1.8)

1. ANTLR . ANTLR Java jar . (: Java) . .

2. Eclipse ANTLR IDE .

    - Eclipse Help Eclipse Marketplace .
    - : antlr .
    - ANTLR 4 IDE .
    - .
    - .
    - Eclipse .

3. " ..." .

    - Eclipse ANTLR 4 HOME . com.github.jknack.antlr-4ide.Antlr4 for com.github.jknack.antlr-4ide.Antlr4 . .
    - HOME . .
    - Xtext 2.7.3 antlr-nnn-complete.jar .
    - Eclipse Help Install New Software .
    - ... .
    - , xtext 2.7.3 Archive ... Xtext 2.7.3 OK .
    - > . .
    - Eclipse .

4. ANTLR Eclipse / Java .

    - Eclipse Window Preferences .
    - Java .
    - ... ... antlr-nnn-complete.jar . Classpath Variables OK .
    - .

5. ( ) ANTLR IDE .

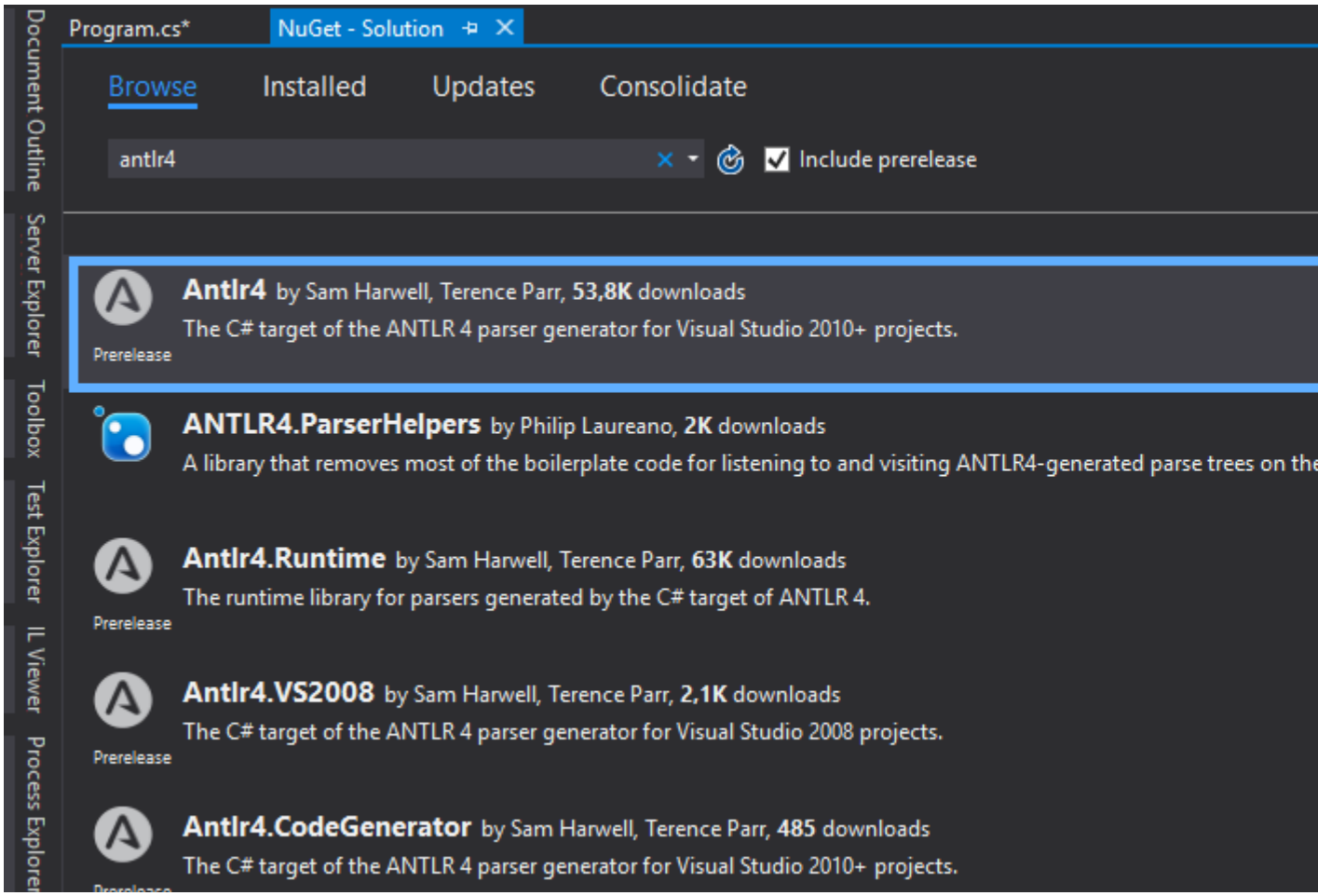
    - Eclipse .
    - ANTLR 4 5 .
    - . , java ./antlr-java .
    - .

6. ANTLR 4 .

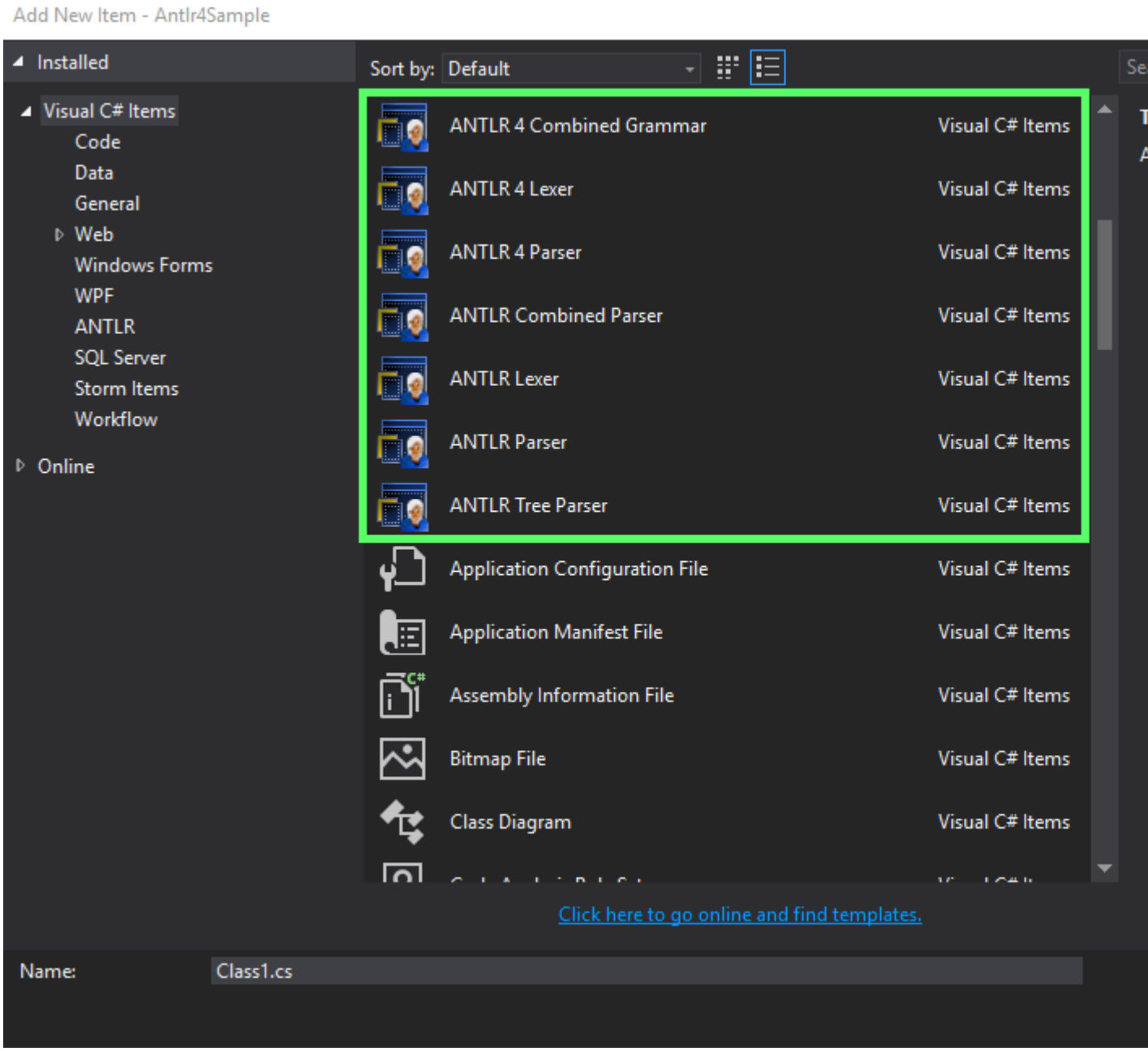
    - Eclipse File, New, Project .
    - ANTLR 4 .
    - .
    - Hello.g4 "Hello World" .
    - g4 target (5) .

Visual Studio 2015 ANTLR (Nuget)

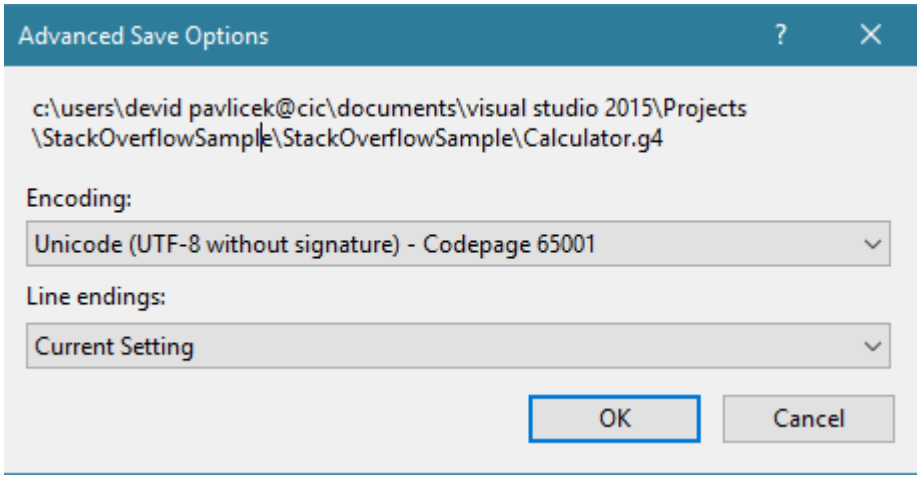
1. Visual Studio 2015 → → Antlr . ANTLR (Sam Harwell) Visual Studio .
2. . → Nuget Packages for Solution → Browse (Tab) Antlr4 .


Program.cs*
NuGet - Solution
Browse
Installed
Updates
Consolidate
antlr4
Include prerelease
Antlr4 by Sam Harwell, Terence Parr, 53,8K downloads
The C# target of the ANTLR 4 parser generator for Visual Studio 2010+ projects.
Prerelease
ANTLR4.ParserHelpers by Philip Laureano, 2K downloads
A library that removes most of the boilerplate code for listening to and visiting ANTLR4-generated parse trees on the
Antlr4.Runtime by Sam Harwell, Terence Parr, 63K downloads
The runtime library for parsers generated by the C# target of ANTLR 4.
Prerelease
Antlr4.VS2008 by Sam Harwell, Terence Parr, 2,1K downloads
The C# target of the ANTLR 4 parser generator for Visual Studio 2008 projects.
Prerelease
Antlr4.CodeGenerator by Sam Harwell, Terence Parr, 485 downloads
The C# target of the ANTLR 4 parser generator for Visual Studio 2010+ projects.
Document Outline
Server Explorer
Toolbox
Test Explorer
IL Viewer
Process Explorer

3. . ANTLR4 .

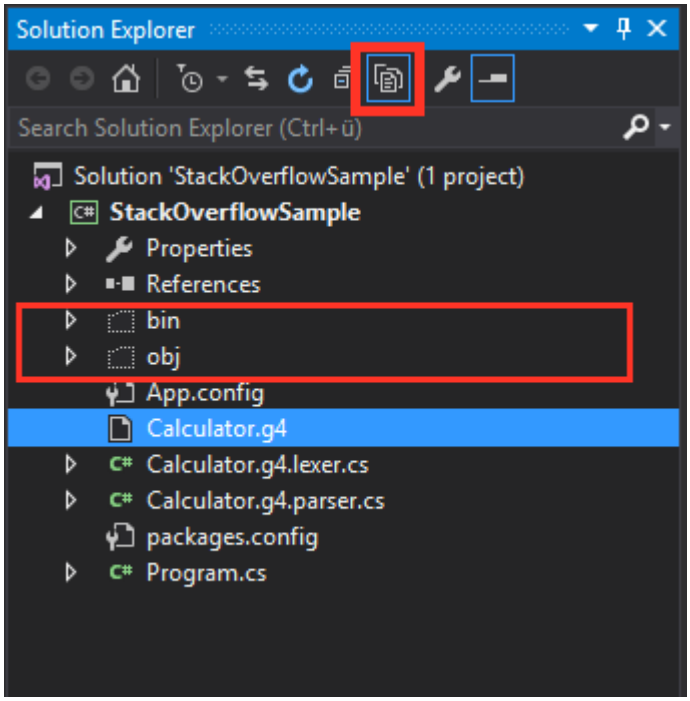


4. ANTLR (.g4) → (UTF-8) - 65001 . .

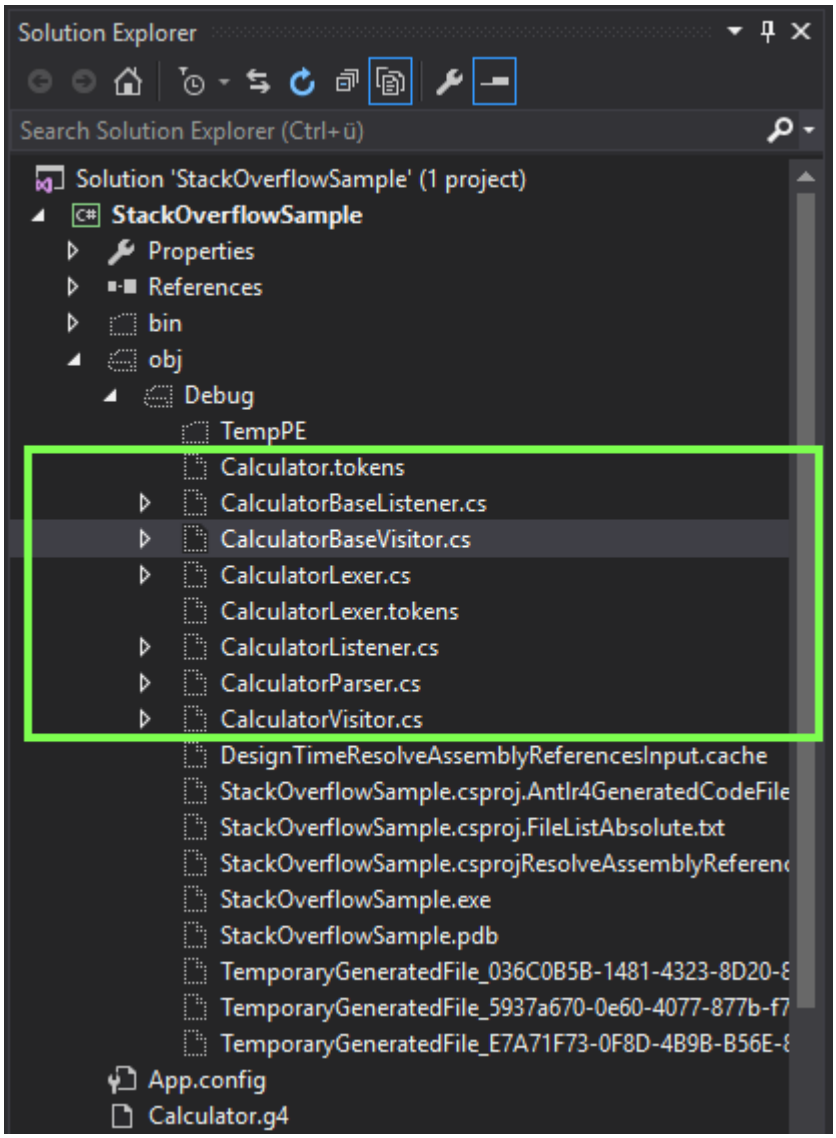


- ANTLR 4 Calculator.g4 .
- Github : Tom Everett

- .
- → .



- ()
- obj CS . . Visual Studio 2015 ANTLR .



ANTLR v4 : https://riptutorial.com/ko/antlr/topic/2856/antlr-v4-

4: ANTLR /

Examples

ANTLR .

1. C #
2. Python
3.
4.

ANTLR Java :

```
Java -jar antlr-4.5.3-complete.jar yourGrammar.g4 //Will output a
    java parser
```

OS / .

```
    antlr4 -Dlanguage=Python3 yourGrammar.g4
//with alias
    java -jar antlr-4.5.3-complete.jar -Dlanguage=Python3 yourGrammar.g4
//without alias
```

/ '-Dlanguage' .g4 .

```
options {
    language  = "CSharp";
}
//or
options {
    language="Python";
}
```

ANTLR .

1. CSharp
2. 2
3. 3

ANTLR

.g4 ANTLR.jar .

```
1.yourGrammarNameListener.py
2.yourGrammarNameParser.py
3.yourGrammarName.tokens
...
```

. ANTLR . IDE .

```
#main.py
import yourGrammarNameParser
import sys

#main method and entry point of application

def main(argv):
    """Main method calling a single debugger for an input script"""
    parser = yourGrammarNameParser
    parser.parse(argv)

if __name__ == '__main__':
    main(sys.argv)
```

.

```
#yourGrammarNameParser.py
from yourGrammarNameLexer import yourGrammarNameLexer
from yourGrammarNameListener import yourGrammarNameListener
from yourGrammarNameParser import yourGrammarNameParser
from antlr4 import *
import sys

class yourGrammarNameParser(object):
    """
    Debugger class - accepts a single input script and processes
    all subsequent requirements
    """
def __init__(self): # this method creates the class object.
    pass


#function used to parse an input file
def parse(argv):
    if len(sys.argv) > 1:
        input = FileStream(argv[1]) #read the first argument as a filestream
        lexer = yourGrammarNameLexer(input) #call your lexer
        stream = CommonTokenStream(lexer)
        parser = yourGrammarNameParser(stream)
        tree = parser.program() #start from the parser rule, however should be changed to your
entry rule for your specific grammar.
        printer = yourGrammarNameListener(tree,input)
        walker = ParseTreeWalker()
        walker.walk(printer, tree)
    else:
        print('Error : Expected a valid file')
```

ANTLR .

.

ANTLR / : https://riptutorial.com/ko/antlr/topic/3414/antlr-----

5: Lexer v4

Examples

. .

```
INTEGER: [0-9]+;
IDENTIFIER: [a-zA-Z_] [a-zA-Z_0-9]*;

OPEN_PAREN: '(';
CLOSE_PAREN: ')';
```

:

A	A
AB	A B
(A|B)	A B
'text'	""
A?	0 1 A
A*	0 A
A+	A A
[A-Z0-9]	(AZ 0-9) .
'a'..'z'	
~[AZ]	-
.	

.

```
INTEGER: DIGIT+
       | '0' [Xx] HEX_DIGIT+
       ;

fragment DIGIT: [0-9];
fragment HEX_DIGIT: [0-9A-Fa-f];
```

'{' .

, :

```
OPEN_BRACE: '{';
```

.

```
parserRule: '{';
parserRule: OPEN_BRACE;
```

OPEN_BRACE . . .

. .

- ,
- (: '{')
-

:

```
grammar LexerPriorityRulesExample;

// Parser rules

randomParserRule: 'foo'; // Implicitly declared token type

// Lexer rules

BAR: 'bar';
IDENTIFIER: [A-Za-z]+;
BAZ: 'baz';

WS: [ \t\r\n]+ -> skip;
```

:

```
aaa foo bar baz barz
```

:

```
IDENTIFIER 'foo' BAR IDENTIFIER IDENTIFIER
```

- aaa IDENTIFIER .

  IDENTIFIER .

- foo 'foo' .

  randomParserRule 'foo' , IDENTIFIER .

- bar BAR .

  IDENTIFIER BAR .

- baz IDENTIFIER .

  BAZ IDENTIFIER . BAR .

  , BAZ , IDENTIFIER BAZ .

- barz IDENTIFIER .

  BAR (bar) 3 IDENTIFIER 4 . IDENTIFIER BAR BAR .

, . .

'foo' .

Lexer

:

```
WHITESPACE: [ \r\n] -> skip;
```

-> .

- skip : skip .
- channel(n) : .
- type(n) : .
- mode(n) , pushMode(n) , popMode , more : .

{ ... } , .

```
IDENTIFIER: [A-Z]+ { log("matched rule"); };
```

{ ... }? }? . false .

```
IDENTIFIER: [A-Z]+ { identifierIsValid() }?;
```

.

Lexer v4 : https://riptutorial.com/ko/antlr/topic/3271/lexer--v4

6: TestRig / grun

Examples

TestRig

ANTLR . .

ANTLR jar ANTLR .

```
export CLASSPATH=".:/usr/local/lib/antlr-4.5.3-complete.jar:$CLASSPATH"
```

: java Dot .

Alises Linux / MAC / Unix :

```
alias antlr4='java -jar /usr/local/lib/antlr-4.5.3-complete.jar'
//or any directory where your jar is located
```

Windows . .

TestRig

TestRig . .

```
alias grun='java org.antlr.v4.runtime.misc.TestRig'
```

Windows ANTLR jar TestRig .

```
java -cp .;antlr.4.5.3-complete.jar org.antlr.v4.runtime.misc.TestRig
//or
java -cp .;antlr.4.5.3-complete.jar org.antlr.v4.gui.TestRig
```

TestRig .

```
grun yourGrammar yourRule -tree //using the setup alias
java -cp .;antlr.4.5.3-complete.jar org.antlr.v4.gui.TestRig yourGrammar YourRule -tree //on
windows with no alias
java -cp .;antlr.4.5.3-complete.jar org.antlr.v4.gui.TestRig yourGrammar Hello r -tree
//Windows with the grammar Hello.g4 starting from the rule 'r'.
```

ANTLR `-gui` . :

:

JSON.g4

```
/** Taken from "The Definitive ANTLR 4 Reference" by Terence Parr */

// Derived from http://json.org
grammar JSON;

json
   : value
   ;

object
   : '{' pair (',' pair)* '}'
   | '{' '}'
   ;

pair
   : STRING ':' value
   ;

array
   : '[' value (',' value)* ']'
   | '[' ']'
   ;

value
   : STRING
   | NUMBER
   | object
   | array
   | 'true'
   | 'false'
   | 'null'
   ;


STRING
   : '"' (ESC | ~ ["\\])* '"'
   ;
fragment ESC
   : '\\' (["\\/bfnrt] | UNICODE)
   ;
fragment UNICODE
   : 'u' HEX HEX HEX HEX
   ;
fragment HEX
   : [0-9a-fA-F]
   ;
NUMBER
   : '-'? INT '.' [0-9] + EXP? | '-'? INT EXP | '-'? INT
   ;
fragment INT
   : '0' | [1-9] [0-9]*
   ;
// no leading zeros
fragment EXP
   : [Ee] [+\-]? INT
   ;
// \- since - means "range" inside [...]
WS
   : [ \t\n\r] + -> skip
   ;
```

JSON :

example.json

```
{
  "name": "John Doe",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

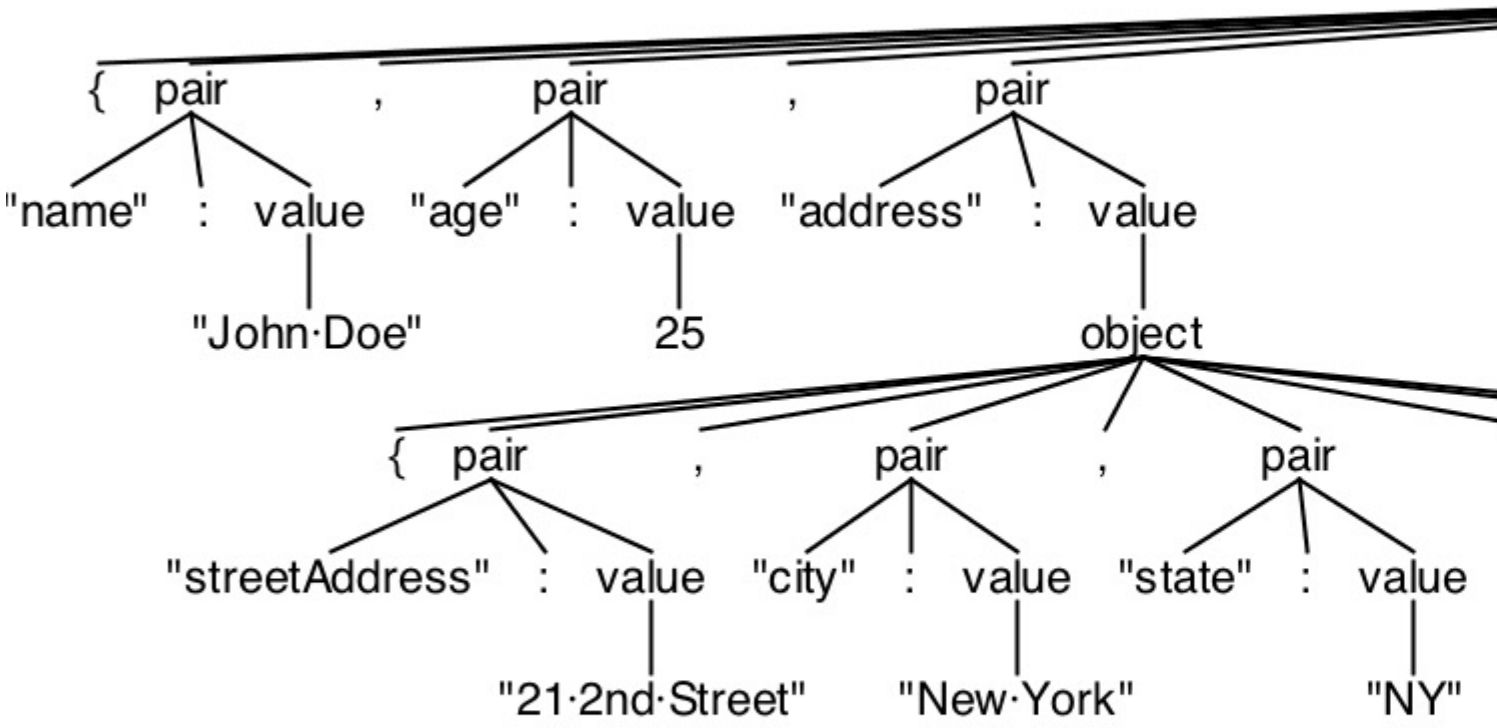
.

```
export CLASSPATH=".:/usr/local/lib/antlr-4.0-complete.jar:$CLASSPATH"

alias antlr4='java -jar /usr/local/lib/antlr-4.0-complete.jar'

alias grun='java org.antlr.v4.runtime.misc.TestRig'

antlr4 -o . -lib . -no-listener -no-visitor  JSON.g4; javac *.java; grun JSON json -gui
example.json
```

.java .tokens .class .

```
JSON.g4                         JSONLexer.class                JSONListener.java
JSONParser$PairContext.class    JSON.tokens                    JSONLexer.java
JSONParser$ArrayContext.class   JSONParser$ValueContext.class  JSONBaseListener.class
JSONLexer.tokens                JSONParser$JsonContext.class   JSONParser.class
JSONBaseListener.java           JSONListener.class
JSONParser$ObjectContext.class  JSONParser.java
```

:



TestRig / grun : https://riptutorial.com/ko/antlr/topic/3270/testrig---grun

7:

? ANTLR . visit () . . .

Examples

(Expr.g4)

```
grammar Expr;
prog:    (expr NEWLINE)* ;
expr:    expr ('*'|'/') expr

    |    expr ('+'|'-') expr
    |    INT
    |    '(' expr ')'
    ;
NEWLINE : [\r\n]+ ;
INT     : [0-9]+ ;
```

.

```
 -visitor           generate parse tree visitor
 -no-visitor        don't generate parse tree visitor (default)
```

/ .

```
java - jar antlr-4.5.3-complete.jar Expr.g4 -visitor
java - jar antlr-4.5.3-complete.jar Expr.g4 -no-visitor
```

/ .

ExprBaseVisitor.java ExprVisitor.java . Java . ExprBaseVisitor .

```
// Generated from Expr.g4 by ANTLR 4.5.3
import org.antlr.v4.runtime.tree.AbstractParseTreeVisitor;

/**
 * This class provides an empty implementation of {@link ExprVisitor},
 * which can be extended to create a visitor which only needs to handle a subset
 * of the available methods.
 *
 * @param <T> The return type of the visit operation. Use {@link Void} for
 * operations with no return type.
 */
public class ExprBaseVisitor<T> extends AbstractParseTreeVisitor<T> implements ExprVisitor<T>
{
    /**
     * {@inheritDoc}
     *
     * <p>The default implementation returns the result of calling
     * {@link #visitChildren} on {@code ctx}.</p>
     */
```

```
    @Override public T visitProg(ExprParser.ProgContext ctx) { return visitChildren(ctx); }
    /**
     * {@inheritDoc}
     *
     * <p>The default implementation returns the result of calling
     * {@link #visitChildren} on {@code ctx}.</p>
     */
    @Override public T visitExpr(ExprParser.ExprContext ctx) { return visitChildren(ctx); }
}
```

: https://riptutorial.com/ko/antlr/topic/8211/

8:

Examples

ANTLR .

.

```
// Rule
type : int     #typeInt
     | short   #typeShort
     | long    #typeLong
     | string  #typeString
     ;

// Tokens
int : 'int' ;
short : 'short' ;
long : 'long' ;
string : 'string' ;
```

ParseTreeListener .

```
public void enterTypeInt(TypeShortContext ctx);
public void enterTypeShort(TypeIntContext ctx);
public void enterTypeLong(TypeLongContext ctx);
public void enterTypeString(TypeStringContext ctx);
```

: https://riptutorial.com/ko/antlr/topic/6717/

S. No		Contributors
1	ANTLR	Athafoud, cb4, Community, D3181, Gábor Bakos, KvanTTT
2	ANTLR v3	Athafoud, cb4
3	ANTLR v4	Athafoud, cb4, Community, D3181, Devid, Gábor Bakos, GRosenberg, Lucas Trzesniewski
4	ANTLR /	D3181
5	Lexer v4	Athafoud, bn., Loxley, Lucas Trzesniewski
6	TestRig / grun	bn., D3181, Lucas Trzesniewski, Pascal Le Merrer
7		D3181
8		bn., Lucas Trzesniewski