19CSE401 - Compiler Design

Programming Language : Racket

Name	Roll Number
Raghul K B	CB.EN.U4CSE19346
O Mahanth	CB.EN.U4CSE19340
V Devakumar	CB.EN.U4CSE19358
V Nithin Krishna	CB.EN.U4CSE19360

Tokens to RE Conversion :

Token	Regular Expression
COMMENT	(';' (LETTER|SYMBOL|SPECIALINITIAL|CONSTANT|' ')+)
KEYWORD	('if'|'define'|'else'|'and'|'or'|'case'|'display'|'loop'|'do'|'quasiquote'|'begin'|'quote'|'lambda'|'set!'|'cond'|'let '|'unquote'|'delay')
WHITESPACE	[\r\n\t]+ -> skip
PAIR	'('(CONSTANT)' . '(CONSTANT)')'
SYMBOL	('+'| '-' |'/'|'*'|'^'|'('|')'|'.'|'<'|'='|'>'|'\"'|',')
IDENTIFIER	(INITIAL (SUBSEQUENT)*)
CONSTANT	(LETTER | STRING | BOOLEAN | DIGIT | NUMBER)
INITIAL	(LETTER | SPECIALINITIAL)
LETTER	(('a'..'z') | ('A'..'Z'))
BOOLEAN	('#t' | '#f')
SPECIALINITIAL	('!'| '$'| '%' |'*'|'/'|':'|'?'|'_')
SUBSEQUENT	(INITIAL | DIGIT)
NUMBER	(('1'..'9') (DIGIT)*)

DIGIT	('0'..'9')
STRING	(LETTER | SPECIALINITIAL | SYMBOL)+
REQUIRE	‘require’
PROVIDE	‘provide’
MAKEHASH	‘make-hash’
CASE	‘case’
DISPLAY	‘display’
LOOP	‘loop’
COMBINATIONS	‘combinations’
REGEXP	‘regexp’
HASHSET	‘hash-set!’
DEFINE	‘define’
IF	‘if’
COMMENT	(';' (*LETTER*|*SYMBOL*|*SPECIALINITIAL*|*CONSTANT*|' ')+)
STRINGAPPEND	‘string-append’
STRINGTRIM	‘string-trim’
ELSE	‘else’
AND	‘and’
CAR	‘car’
CDR	‘cdr’
OR	‘or’
DO	‘do’
MAP	‘map’
LAMBDASYM	'λ'

BEGIN	‘begin’’
QUOTE	'quote'
QUASIQUOTE	‘quasiquote’
SET	‘set’
LAMBDA	‘lambda’
SETNOT	‘set!’
REVERSE	‘reverse’
COND	‘cond’
LET	‘let’
UNQOUTE	‘unquote’
DELAY	‘delay’
WS	‘[\r\n\t]+ **->** skip’
PAIR	'('(*CONSTANT*)' . '(*CONSTANT*)')'
PLUS	‘+’
MINUS	‘-’
DOUBLEQ	‘“”’
BACKSLASH	‘\\’
LEFTB	‘(’
RIGHTB	‘)’
FULLSTOP	‘.’
LESSTHAN	‘<’
ASSIGNMENT	‘=’
SINGLEQ	‘’
GREATERTHAN	‘>’

COMMA	‘,’
EXCLAMATIONMARK	‘!’
DOLLAR	‘$’
PERCENTAGE	‘%’
MUL	‘*’
DIVISION	‘/’
COLON	‘:’
QUESTIONMARK	‘?’
UNDERSCORE	‘_’
ISNULL	‘null?’
IMPORT	‘('#lang '('racket'|'lisp'|'info')+)’

Lexer for Racket ：

```
import java.io.IOException;
import org.antlr.v4.runtime.ANTLRFileStream;
import org.antlr.v4.runtime.ANTLRInputStream;
import org.antlr.v4.runtime.Token;

public class Main {

	public static void main(String[] args) throws IOException {
		ANTLRFileStream input = new ANTLRFileStream(args[0]);
	  ANTLRInputStream ip = new ANTLRInputStream(input.toString());
	  hello lex = new hello(ip);
	  Token token;
	  while ((token = lex.nextToken()).getType()!= -1)
	  {
		System.out.println("line " + token.getLine()+ ":" +
token.getStartIndex() + " token <"
				+ token.getType() + ", "+token.getText() +
">");
	  }
}
}
```

Programs :

Basic :

```
#lang racket
( begin
  ( display "Hello, World!" )
   newline  )
```

Output :

```
line 1:0 token <57, #lang racket>
line 2:13 token <29, (>
line 2:15 token <14, begin>
line 3:23 token <29, (>
line 3:25 token <10, display>
line 3:33 token <50, "Hello,>
line 3:41 token <50, World!">
line 3:49 token <30, )>
line 4:53 token <29, (>
line 4:55 token <49, newline>
line 4:63 token <30, )>
line 4:65 token <30, )>
```

Stack :

```
#lang racket
( define stack empty )
( define ( push x )
  ( set! stack ( append stack ( list x ) ) ) )
( define pop ( reverse stack ) )
  ( define result ( car pop ) )
    ( set! stack ( reverse ( cdr pop ) ) )
     result

( push "abc" )
( push "efg" )
( push "ijk" )
pop
display stack
```

Output :

```
line 1:0 token <39, #lang racket>
line 2:13 token <47, (>
line 2:15 token <3, define>
line 2:22 token <67, stack>
line 2:28 token <67, empty>
line 2:34 token <48, )>
line 3:36 token <47, (>
line 3:38 token <3, define>
line 3:45 token <47, (>
line 3:47 token <67, push>
line 3:52 token <67, x>
line 3:54 token <48, )>
line 4:58 token <47, (>
line 4:60 token <33, set!>
line 4:65 token <67, stack>
line 4:71 token <47, (>
line 4:73 token <67, append>
line 4:80 token <67, stack>
line 4:86 token <47, (>
line 4:88 token <67, list>
line 4:93 token <67, x>
line 4:95 token <48, )>
line 4:97 token <48, )>
line 4:99 token <48, )>
line 4:101 token <48, )>
line 5:103 token <47, (>
line 5:105 token <3, define>
line 5:112 token <67, pop>
line 5:116 token <47, (>
line 5:118 token <34, reverse>
line 5:126 token <67, stack>
line 5:132 token <48, )>
line 5:134 token <48, )>
line 6:138 token <47, (>
line 6:140 token <3, define>
line 6:147 token <67, result>
line 6:154 token <47, (>
line 6:156 token <18, car>
line 6:160 token <67, pop>
line 6:164 token <48, )>
line 6:166 token <48, )>
line 7:172 token <47, (>
line 7:174 token <33, set!>
line 7:179 token <67, stack>
line 7:185 token <47, (>
line 7:187 token <34, reverse>
```

line 7:195 token <47, (>
line 7:197 token <19, cdr>
line 7:201 token <67, pop>
line 7:205 token <48,)>
line 7:207 token <48,)>
line 7:209 token <48,)>
line 8:217 token <67, result>
line 10:231 token <47, (>
line 10:233 token <67, push>
line 10:238 token <68, "abc">
line 10:244 token <48,)>
line 11:246 token <47, (>
line 11:248 token <67, push>
line 11:253 token <68, "efg">
line 11:259 token <48,)>
line 12:261 token <47, (>
line 12:263 token <67, push>
line 12:268 token <68, "ijk">
line 12:274 token <48,)>
line 13:276 token <67, pop>
line 14:280 token <23, display>
line 14:288 token <67, stack>

List :

```
#lang racket

( define ( sum lst )
  ( if ( null? lst ) 0
      ( + ( car lst ) ( sum ( cdr lst ) ) ) ) )
( sum  ( 1 2 3 ) )

( define ( sublsum lst )
  ( if ( null? lst )
      '()
      ( sublsum ( combinations lst ) ) ) )

( sublsum  ( 1 2 3 4 -5 ) )
( sublsum  ( 1 2 3 4 5 ) )
```

Output :

```
line 1:0 token <57, #lang racket>
line 2:13 token <1, ;helper function>
line 3:30 token <29, (>
line 3:32 token <3, define>
line 3:39 token <29, (>
line 3:41 token <49, sum>
line 3:45 token <49, lst>
line 3:49 token <30, )>
line 4:53 token <29, (>
line 4:55 token <2, if>
line 4:58 token <29, (>
line 4:60 token <49, null?>
line 4:66 token <49, lst>
line 4:70 token <30, )>
line 4:72 token <46, 0>
line 5:76 token <29, (>
line 5:78 token <24, +>
line 5:80 token <29, (>
line 5:82 token <6, car>
line 5:86 token <49, lst>
line 5:90 token <30, )>
line 5:92 token <29, (>
line 5:94 token <49, sum>
line 5:98 token <29, (>
line 5:100 token <7, cdr>
line 5:104 token <49, lst>
line 5:108 token <30, )>
line 5:110 token <30, )>
line 5:112 token <30, )>
line 5:114 token <30, )>
line 5:116 token <30, )>
line 6:118 token <29, (>
line 6:120 token <49, sum>
line 6:124 token <35, '>
line 6:126 token <29, (>
line 6:128 token <45, 1>
line 6:130 token <45, 2>
line 6:132 token <45, 3>
line 6:134 token <30, )>
line 6:136 token <30, )>
line 6:138 token <1, ; 6>
line 7:142 token <1, ;car, cdr, combinations, >
line 9:169 token <29, (>
line 9:171 token <3, define>
line 9:178 token <29, (>
line 9:180 token <50, sublsum-aux>
```

line 9:192 token <49, lst>
line 9:196 token <30,)>
line 10:200 token <29, (>
line 10:202 token <2, if>
line 10:205 token <29, (>
line 10:207 token <49, null?>
line 10:213 token <49, lst>
line 10:217 token <30,)>
line 11:220 token <50, '()>
line 12:225 token <29, (>
line 12:227 token <2, if>
line 12:230 token <29, (>
line 12:232 token <5, and>
line 12:236 token <29, (>
line 12:238 token <33, =>
line 12:240 token <29, (>
line 12:242 token <49, sum>
line 12:246 token <29, (>
line 12:248 token <6, car>
line 12:252 token <49, lst>
line 12:256 token <30,)>
line 12:258 token <30,)>
line 12:260 token <46, 0>
line 12:262 token <30,)>
line 12:264 token <29, (>
line 12:266 token <49, not>
line 12:270 token <29, (>
line 12:272 token <49, null?>
line 12:278 token <29, (>
line 12:280 token <6, car>
line 12:284 token <49, lst>
line 12:288 token <30,)>
line 12:290 token <30,)>
line 12:292 token <30,)>
line 12:294 token <30,)>
line 12:296 token <1, ; car returns the head of the list>
line 13:334 token <29, (>
line 13:336 token <49, cons>
line 13:341 token <29, (>
line 13:343 token <6, car>
line 13:347 token <49, lst>
line 13:351 token <30,)>
line 13:353 token <29, (>
line 13:355 token <50, sublsum-aux>
line 13:367 token <29, (>
line 13:369 token <7, cdr>
line 13:373 token <49, lst>
line 13:377 token <30,)>
line 13:379 token <30,)>

line 13:381 token <30,)>
line 14:389 token <29, (>
line 14:391 token <50, sublsum-aux>
line 14:403 token <29, (>
line 14:405 token <7, cdr>
line 14:409 token <49, lst>
line 14:413 token <30,)>
line 14:415 token <30,)>
line 14:417 token <30,)>
line 14:419 token <30,)>
line 14:421 token <30,)>
line 16:424 token <29, (>
line 16:426 token <3, define>
line 16:433 token <29, (>
line 16:435 token <49, sublsum>
line 16:443 token <49, lst>
line 16:447 token <30,)>
line 17:451 token <29, (>
line 17:453 token <2, if>
line 17:456 token <29, (>
line 17:458 token <49, null?>
line 17:464 token <49, lst>
line 17:468 token <30,)>
line 18:471 token <50, '()>
line 19:476 token <29, (>
line 19:478 token <50, sublsum-aux>
line 19:490 token <29, (>
line 19:492 token <49, combinations>
line 19:505 token <49, lst>
line 19:509 token <30,)>
line 19:511 token <30,)>
line 19:513 token <30,)>
line 19:515 token <30,)>
line 19:517 token <1, ; combinations>
line 20:532 token <29, (>
line 20:534 token <49, sublsum>
line 20:542 token <35, '>
line 20:544 token <29, (>
line 20:545 token <45, 1>
line 20:547 token <45, 2>
line 20:549 token <45, 3>
line 20:551 token <45, 4>
line 20:553 token <25, ->
line 20:554 token <45, 5>
line 20:556 token <30,)>
line 20:558 token <30,)>
line 21:560 token <29, (>
line 21:562 token <49, sublsum>
line 21:570 token <35, '>

line 21:572 token <29, (>
line 21:574 token <45, 1>
line 21:576 token <45, 2>
line 21:578 token <45, 3>
line 21:580 token <45, 4>
line 21:582 token <45, 5>
line 21:584 token <30,)>
line 21:586 token <30,)>

Tree :

```
#lang racket

( define ( element-of-set? x set )
  ( cond ( ( null? set ) false )
         ( ( equal? x ( car set ) ) true)
         ( else (element-of-set? x ( cdr set ) ) )
   )
 )

; 2) adjoin set
; cons ( set element ) if element not in set

( define ( adjoin-set x set )
  ( if ( element-of-set? x set )
      set
  ( cons x set ) )
)

; 3) intersection-set T(n) = revisit

( define ( intersection-set set1 set2 )
  ( cond ( ( or ( null? set1 ) ( null ? set2 ) ) )
         ( ( element-of-set? ( car set1 ) set2 )
           (cons (car set1)
( intersection-set ( cdr set1 ) set2 ) ) )
         ( else ( intersection-set ( cdr set1 ) set2 ) ) ) )

; intersection-set takes 2 sets
; returns a set that contains only the common elements
; sets as ordered lists - this speeds up traversal

( define ( element-of-set? x set )
```

```
  ( cond ( ( null? set ) false )
         ( ( = x ( car set ) ) true )
         ( ( < x ( car set ) ) false )
         ( else ( element-of-set? x ( cdr set ) ) ) ) )

; define a tree based on procedure:
; each node will be a list of 3 items
; 1 - the entry at the node
; 2 - the left subtree
; 3 - the right subtree

( define ( entry tree ) ( car tree ) )
( define ( left-branch tree ) ( cadr tree ) )
( define ( right-branch tree) ( caddr tree ) )
( define ( make-tree entry left right )
  ( list entry left right ) )

; this needs a new element-of-set?
; T(n) = revisit

( define ( element-of-set? x set )
  ( cond ( ( null? set ) false)
         ( ( = x ( entry set ) ) true)
         ( (< x ( entry set ) )
          ( element-of-set? x ( left-branch set ) ) )
         ( (> x ( entry set ) )
          ( element-of-set? x ( left-branch set ) ) ) ) )

; now adjoin-set

( define ( adjoin-set x set )
  ( cond ( ( null ? set ) ( make-tree x ' ( ) '( ) ) )
         ( ( = x ( entry set ) ) set )
         ( ( < x ( entry set ) )
          ( make-tree ( entry set )
                      ( adjoin-set x ( left-branch set ) )
                      ( right-branch set ) ) )
         ( ( > x ( entry set ) )
          ( make-tree ( entry set )
                      ( left-branch set )
                      ( adjoin-set x ( right-branch set ) ) ) ) ) )
```

Output :

```
line 1:0 token <57, #lang racket>
line 3:14 token <29, (>
line 3:16 token <3, define>
line 3:23 token <29, (>
line 3:25 token <50, element-of-set?>
line 3:41 token <49, x>
line 3:43 token <17, set>
line 3:47 token <30, )>
line 4:51 token <29, (>
line 4:53 token <18, cond>
line 4:58 token <29, (>
line 4:60 token <29, (>
line 4:62 token <49, null?>
line 4:68 token <17, set>
line 4:72 token <30, )>
line 4:74 token <49, false>
line 4:80 token <30, )>
line 5:90 token <29, (>
line 5:92 token <29, (>
line 5:94 token <49, equal?>
line 5:101 token <49, x>
line 5:103 token <29, (>
line 5:105 token <6, car>
line 5:109 token <17, set>
line 5:113 token <30, )>
line 5:115 token <30, )>
line 5:117 token <50, true)>
line 6:131 token <29, (>
line 6:133 token <4, else>
line 6:138 token <50, (element-of-set?>
line 6:155 token <49, x>
line 6:157 token <29, (>
line 6:159 token <7, cdr>
line 6:163 token <17, set>
line 6:167 token <30, )>
line 6:169 token <30, )>
line 6:171 token <30, )>
line 7:176 token <30, )>
line 8:179 token <30, )>
line 10:182 token <1, ; 2) adjoin set>
line 11:198 token <1, ; cons ( set element ) if element not in set>
line 13:244 token <29, (>
line 13:246 token <3, define>
line 13:253 token <29, (>
line 13:255 token <50, adjoin-set>
```

line 13:266 token <49, x>
line 13:268 token <17, set>
line 13:272 token <30,)>
line 14:276 token <29, (>
line 14:278 token <2, if>
line 14:281 token <29, (>
line 14:283 token <50, element-of-set?>
line 14:299 token <49, x>
line 14:301 token <17, set>
line 14:305 token <30,)>
line 15:313 token <17, set>
line 16:319 token <29, (>
line 16:321 token <49, cons>
line 16:326 token <49, x>
line 16:328 token <17, set>
line 16:332 token <30,)>
line 16:334 token <30,)>
line 17:336 token <30,)>
line 19:339 token <1, ; 3) intersection-set T(n) = revisit>
line 21:377 token <29, (>
line 21:379 token <3, define>
line 21:386 token <29, (>
line 21:388 token <50, intersection-set>
line 21:405 token <49, set1>
line 21:410 token <49, set2>
line 21:415 token <30,)>
line 22:420 token <29, (>
line 22:422 token <18, cond>
line 22:427 token <29, (>
line 22:429 token <29, (>
line 22:431 token <8, or>
line 22:434 token <29, (>
line 22:436 token <49, null?>
line 22:442 token <49, set1>
line 22:447 token <30,)>
line 22:449 token <29, (>
line 22:451 token <49, null>
line 22:456 token <43, ?>
line 22:458 token <49, set2>
line 22:463 token <30,)>
line 22:465 token <30,)>
line 22:467 token <30,)>
line 23:477 token <29, (>
line 23:479 token <29, (>
line 23:481 token <50, element-of-set?>
line 23:497 token <29, (>
line 23:499 token <6, car>
line 23:503 token <49, set1>
line 23:508 token <30,)>

line 23:510 token <49, set2>
line 23:515 token <30,)>
line 24:527 token <50, (cons>
line 24:533 token <50, (car>
line 24:538 token <49, set1>
line 24:542 token <30,)>
line 25:544 token <29, (>
line 25:546 token <50, intersection-set>
line 25:563 token <29, (>
line 25:565 token <7, cdr>
line 25:569 token <49, set1>
line 25:574 token <30,)>
line 25:576 token <49, set2>
line 25:581 token <30,)>
line 25:583 token <30,)>
line 25:585 token <30,)>
line 26:595 token <29, (>
line 26:597 token <4, else>
line 26:602 token <29, (>
line 26:604 token <50, intersection-set>
line 26:621 token <29, (>
line 26:623 token <7, cdr>
line 26:627 token <49, set1>
line 26:632 token <30,)>
line 26:634 token <49, set2>
line 26:639 token <30,)>
line 26:641 token <30,)>
line 26:643 token <30,)>
line 26:645 token <30,)>
line 28:650 token <1, ; intersection-set takes 2 sets>
line 29:682 token <1, ; returns a set that contains only the common elements>
line 30:737 token <1, ; sets as ordered lists - this speeds up traversal>
line 32:789 token <29, (>
line 32:791 token <3, define>
line 32:798 token <29, (>
line 32:800 token <50, element-of-set?>
line 32:816 token <49, x>
line 32:818 token <17, set>
line 32:822 token <30,)>
line 33:826 token <29, (>
line 33:828 token <18, cond>
line 33:833 token <29, (>
line 33:835 token <29, (>
line 33:837 token <49, null?>
line 33:843 token <17, set>
line 33:847 token <30,)>
line 33:849 token <49, false>
line 33:855 token <30,)>
line 34:865 token <29, (>

line 34:867 token <29, (>
line 34:869 token <33, =>
line 34:871 token <49, x>
line 34:873 token <29, (>
line 34:875 token <6, car>
line 34:879 token <17, set>
line 34:883 token <30,)>
line 34:885 token <30,)>
line 34:887 token <49, true>
line 34:892 token <30,)>
line 35:902 token <29, (>
line 35:904 token <29, (>
line 35:906 token <32, <>
line 35:908 token <49, x>
line 35:910 token <29, (>
line 35:912 token <6, car>
line 35:916 token <17, set>
line 35:920 token <30,)>
line 35:922 token <30,)>
line 35:924 token <49, false>
line 35:930 token <30,)>
line 36:940 token <29, (>
line 36:942 token <4, else>
line 36:947 token <29, (>
line 36:949 token <50, element-of-set?>
line 36:965 token <49, x>
line 36:967 token <29, (>
line 36:969 token <7, cdr>
line 36:973 token <17, set>
line 36:977 token <30,)>
line 36:979 token <30,)>
line 36:981 token <30,)>
line 36:983 token <30,)>
line 36:985 token <30,)>
line 38:988 token <1, ; define a tree based on procedure>
line 38:1022 token <42, :>
line 39:1024 token <1, ; each node will be a list of 3 items>
line 40:1062 token <1, ; 1 - the entry at the node>
line 41:1090 token <1, ; 2 - the left subtree>
line 42:1113 token <1, ; 3 - the right subtree>
line 44:1138 token <29, (>
line 44:1140 token <3, define>
line 44:1147 token <29, (>
line 44:1149 token <49, entry>
line 44:1155 token <49, tree>
line 44:1160 token <30,)>
line 44:1162 token <29, (>
line 44:1164 token <6, car>
line 44:1168 token <49, tree>

line 44:1173 token <30,)>
line 44:1175 token <30,)>
line 45:1177 token <29, (>
line 45:1179 token <3, define>
line 45:1186 token <29, (>
line 45:1188 token <50, left-branch>
line 45:1200 token <49, tree>
line 45:1205 token <30,)>
line 45:1207 token <29, (>
line 45:1209 token <49, cadr>
line 45:1214 token <49, tree>
line 45:1219 token <30,)>
line 45:1221 token <30,)>
line 46:1223 token <29, (>
line 46:1225 token <3, define>
line 46:1232 token <29, (>
line 46:1234 token <50, right-branch>
line 46:1247 token <50, tree)>
line 46:1253 token <29, (>
line 46:1255 token <49, caddr>
line 46:1261 token <49, tree>
line 46:1266 token <30,)>
line 46:1268 token <30,)>
line 47:1270 token <29, (>
line 47:1272 token <3, define>
line 47:1279 token <29, (>
line 47:1281 token <50, make-tree>
line 47:1291 token <49, entry>
line 47:1297 token <49, left>
line 47:1302 token <49, right>
line 47:1308 token <30,)>
line 48:1312 token <29, (>
line 48:1314 token <49, list>
line 48:1319 token <49, entry>
line 48:1325 token <49, left>
line 48:1330 token <49, right>
line 48:1336 token <30,)>
line 48:1338 token <30,)>
line 50:1341 token <1, ; this needs a new element-of-set?>
line 51:1376 token <1, ; T(n) = revisit>
line 53:1394 token <29, (>
line 53:1396 token <3, define>
line 53:1403 token <29, (>
line 53:1405 token <50, element-of-set?>
line 53:1421 token <49, x>
line 53:1423 token <17, set>
line 53:1427 token <30,)>
line 54:1431 token <29, (>
line 54:1433 token <18, cond>

line 54:1438 token <29, (>
line 54:1440 token <29, (>
line 54:1442 token <49, null?>
line 54:1448 token <17, set>
line 54:1452 token <30,)>
line 54:1454 token <50, false)>
line 55:1469 token <29, (>
line 55:1471 token <29, (>
line 55:1473 token <33, =>
line 55:1475 token <49, x>
line 55:1477 token <29, (>
line 55:1479 token <49, entry>
line 55:1485 token <17, set>
line 55:1489 token <30,)>
line 55:1491 token <30,)>
line 55:1493 token <50, true)>
line 56:1507 token <29, (>
line 56:1509 token <50, (<>
line 56:1512 token <49, x>
line 56:1514 token <29, (>
line 56:1516 token <49, entry>
line 56:1522 token <17, set>
line 56:1526 token <30,)>
line 56:1528 token <30,)>
line 57:1539 token <29, (>
line 57:1541 token <50, element-of-set?>
line 57:1557 token <49, x>
line 57:1559 token <29, (>
line 57:1561 token <50, left-branch>
line 57:1573 token <17, set>
line 57:1577 token <30,)>
line 57:1579 token <30,)>
line 57:1581 token <30,)>
line 58:1591 token <29, (>
line 58:1593 token <50, (>>
line 58:1596 token <49, x>
line 58:1598 token <29, (>
line 58:1600 token <49, entry>
line 58:1606 token <17, set>
line 58:1610 token <30,)>
line 58:1612 token <30,)>
line 59:1623 token <29, (>
line 59:1625 token <50, element-of-set?>
line 59:1641 token <49, x>
line 59:1643 token <29, (>
line 59:1645 token <50, left-branch>
line 59:1657 token <17, set>
line 59:1661 token <30,)>
line 59:1663 token <30,)>

line 59:1665 token <30,)>
line 59:1667 token <30,)>
line 59:1669 token <30,)>
line 61:1673 token <1, ; now adjoin-set>
line 63:1691 token <29, (>
line 63:1693 token <3, define>
line 63:1700 token <29, (>
line 63:1702 token <50, adjoin-set>
line 63:1713 token <49, x>
line 63:1715 token <17, set>
line 63:1719 token <30,)>
line 64:1723 token <29, (>
line 64:1725 token <18, cond>
line 64:1730 token <29, (>
line 64:1732 token <29, (>
line 64:1734 token <49, null>
line 64:1739 token <43, ?>
line 64:1741 token <17, set>
line 64:1745 token <30,)>
line 64:1747 token <29, (>
line 64:1749 token <50, make-tree>
line 64:1759 token <49, x>
line 64:1761 token <35, '>
line 64:1763 token <29, (>
line 64:1765 token <30,)>
line 64:1767 token <50, '(>
line 64:1770 token <30,)>
line 64:1772 token <30,)>
line 64:1774 token <30,)>
line 65:1784 token <29, (>
line 65:1786 token <29, (>
line 65:1788 token <33, =>
line 65:1790 token <49, x>
line 65:1792 token <29, (>
line 65:1794 token <49, entry>
line 65:1800 token <17, set>
line 65:1804 token <30,)>
line 65:1806 token <30,)>
line 65:1808 token <17, set>
line 65:1812 token <30,)>
line 66:1822 token <29, (>
line 66:1824 token <29, (>
line 66:1826 token <32, <>
line 66:1828 token <49, x>
line 66:1830 token <29, (>
line 66:1832 token <49, entry>
line 66:1838 token <17, set>
line 66:1842 token <30,)>
line 66:1844 token <30,)>

line 67:1855 token <29, (>
line 67:1857 token <50, make-tree>
line 67:1867 token <29, (>
line 67:1869 token <49, entry>
line 67:1875 token <17, set>
line 67:1879 token <30,)>
line 68:1901 token <29, (>
line 68:1903 token <50, adjoin-set>
line 68:1914 token <49, x>
line 68:1916 token <29, (>
line 68:1918 token <50, left-branch>
line 68:1930 token <17, set>
line 68:1934 token <30,)>
line 68:1936 token <30,)>
line 69:1958 token <29, (>
line 69:1960 token <50, right-branch>
line 69:1973 token <17, set>
line 69:1977 token <30,)>
line 69:1979 token <30,)>
line 69:1981 token <30,)>
line 70:1991 token <29, (>
line 70:1993 token <29, (>
line 70:1995 token <34, >>
line 70:1997 token <49, x>
line 70:1999 token <29, (>
line 70:2001 token <49, entry>
line 70:2007 token <17, set>
line 70:2011 token <30,)>
line 70:2013 token <30,)>
line 71:2024 token <29, (>
line 71:2026 token <50, make-tree>
line 71:2036 token <29, (>
line 71:2038 token <49, entry>
line 71:2044 token <17, set>
line 71:2048 token <30,)>
line 72:2070 token <29, (>
line 72:2072 token <50, left-branch>
line 72:2084 token <17, set>
line 72:2088 token <30,)>
line 73:2110 token <29, (>
line 73:2112 token <50, adjoin-set>
line 73:2123 token <49, x>
line 73:2125 token <29, (>
line 73:2127 token <50, right-branch>
line 73:2140 token <17, set>
line 73:2144 token <30,)>
line 73:2146 token <30,)>
line 73:2148 token <30,)>
line 73:2150 token <30,)>

line 73:2152 token <30,)>
line 73:2154 token <30,)>

HTTP Server :

```
#lang racket
( require web-server/servlet
           web-server/servlet-env )

( provide http/get
         http/post
         http/delete
         http/put
         server/set-port
         request/param
         request/cookie
         response/add-header
         response/add-cookie
         response/make
         response/404
         server/run )

; Request handlers
( define handlers ( make-hash ) )
; Listening port
( define env/port 8000 )
; HTTP-params hash(for GET/POST etc. vars)
( define params ( make-hash ) )
; Cookies hash
( define cookies ( make-hash ) )
; Headers list
( define headers ( list ( make-header #"Cache-Control" #"no-cache" ) )
)

; Creating empty hash-maps for request handlers
( map
 (λ ( element )
   ( hash-set! handlers element ( make-hash ) ) )
 '( get post put delete ) )

; Adds header to http response
```

```
( define ( response/add-header header )
  ( set! headers ( append headers ( list header ) ) ) )

; Creates cookie
( define ( response/add-cookie name value )
  ( response/add-header
        ( cookie->header
         ( make-cookie name value ) ) ) )

; Remember request handler
; Params are - request method(get/post/put/delete), path("/",
"/hello/:id/") and callback
( define ( http/handler method path proc )
  ( hash-set! ( hash-ref handlers method ) path proc ) )

; GET request handler
( define ( http/get path proc )
  ( http/handler 'get path proc ) )

; POST request handler
( define ( http/post path proc )
  ( http/handler 'post path proc ) )

; PUT request handler
( define (http/put path proc )
  ( http/handler 'put path proc ) )

; DELETE request handler
( define ( http/delete path proc )
  ( http/handler 'delete path proc ) )

; Http path template to regexp translation
( define ( path->regexp path )
  ( regexp
   ( string-append
    "^/"
    ( string-trim
     ( regexp-replace* #rx":[^\\/]+" path "([^/?]+)" ) "/" )
    "(?:\\?|$)" ) ) )

; Set listen port
( define ( server/set-port [ port 8000 ] )
```

```
  ( set! env/port port ) )

; Get request parameter
( define ( request/param name )
  ( hash-ref params name #f ) )

; Parse cookies to hash-map
( define ( server/parse-cookies req )
  ( map ( λ ( cookie )
         ( hash-set! cookies
                     ( client-cookie-name cookie )
                     ( client-cookie-value cookie ) ) )
       ( request-cookies req ) ) )

; Get cookie by name
( define ( request/cookie name )
  ( hash-ref cookies name #f ) )

; Make response
( define ( response/make #:code [ code 200 ]
                         #:message [ message #"OK" ]
                         #:seconds [ seconds ( current-seconds ) ]
                         #:mime-type [ mime-type TEXT/HTML-MIME-TYPE ]
                         #:headers [ headers ( cond
                                               [ ( empty? headers ) (
list ( make-header #"Cache-Control" #"no-cache" ) ) ]
                                               [ else headers ] ) ]
                         content )
  ( response/full code
                  message
                  seconds
                  mime-type
                  headers
                  ( list ( string->bytes/utf content) ) ) )

; 404 Response
( define ( response/404 )
  ( response/make #:code 404 "Page not found" ) )

; Find handler
( define ( server/find-handler req )
```

```
  ( let* ( ( path ( regexp-replace* #rx"\\?.*" ( url->string (
request-uri req ) ) "" ) )
         ( method
          ( case ( request-method req)
            [ ( #"GET" ) 'get ]
            [ ( #"POST" ) 'post ]
            [ ( #"PUT" ) 'put ]
            [ ( #"DELETE" ) 'delete ] ) )
         ( handler-key
          ( findf ( λ ( key )  ( regexp-match ( path->regexp key )
path ) )
                  ( hash-keys ( hash-ref handlers method ) ) ) ) )
    ( list path handler-key method ) ) )

; Makes response
( define ( server/response path-handler-method req )
  ( let* ( ( path ( car path-handler-method ) )
         ( handler-key ( cadr path-handler-method ) )
         ( method ( caddr path-handler-method ) ) )
    ( case handler-key
      [ ( #f ) ( response/404 ) ]
      [ else
       ( let* ( ( keys ( map
                       ( λ ( match ) ( string->symbol ( substring match
2 ) ) )
                       ( regexp-match* #rx"/:([^\\/]+)" handler-key ) )
)
              ( pairs
               ( for/list ( [ key keys ]
                            [ val ( cdr ( regexp-match ( path->regexp
handler-key ) path ) ) ] )
                  ( cons key val ) ) ) )
         ( set! params (make-hash ( append pairs ( url-query (
request-uri req ) ) ) ) )
         ( server/parse-cookies req )
         ( let* ( ( handler ( hash-ref handlers method ) )
                ( response ( ( hash-ref handler handler-key #f ) req )
) )
           ( if ( response? response )
               response
               ( response/make ( ( hash-ref handler handler-key #f )
req ) ) ) ) ) ] ) ) )
```

```
; Runs server
( define ( server/run )
  ( serve/servlet
   ( λ ( req )
     ( set! headers ( list (make-header #"Cache-Control" #"no-cache" )
) )
     ( set! cookies ( make-hash ) )
     ( server/response
      ( server/find-handler req )
      req ) ) ) )
```

Output :

line 1:0 token <68, #lang racket>
line 2:13 token <41, (>
line 2:15 token <4, require>
line 2:23 token <62, web-server/servlet>
line 3:53 token <62, web-server/servlet-env>
line 3:76 token <42,)>
line 5:81 token <41, (>
line 5:83 token <5, provide>
line 5:91 token <61, http/get>
line 6:109 token <61, http/post>
line 7:128 token <61, http/delete>
line 8:149 token <61, http/put>
line 9:167 token <62, server/set-port>
line 10:192 token <61, request/param>
line 11:215 token <61, request/cookie>
line 12:239 token <62, response/add-header>
line 13:268 token <62, response/add-cookie>
line 14:297 token <61, response/make>
line 15:320 token <61, response/404>
line 16:342 token <61, server/run>
line 16:353 token <42,)>
line 18:356 token <1, ; Request handlers>
line 19:375 token <41, (>
line 19:377 token <3, define>
line 19:384 token <61, handlers>
line 19:393 token <41, (>
line 19:395 token <6, make-hash>
line 19:405 token <42,)>
line 19:407 token <42,)>
line 20:409 token <1, ; Listening port>
line 21:426 token <41, (>
line 21:428 token <3, define>
line 21:435 token <61, env/port>
line 21:444 token <57, 8000>

line 21:449 token <42,)>
line 22:451 token <1, ; HTTP-params hash(for GET/POST etc. vars)>
line 23:494 token <41, (>
line 23:496 token <3, define>
line 23:503 token <61, params>
line 23:510 token <41, (>
line 23:512 token <6, make-hash>
line 23:522 token <42,)>
line 23:524 token <42,)>
line 24:526 token <1, ; Cookies hash>
line 25:541 token <41, (>
line 25:543 token <3, define>
line 25:550 token <61, cookies>
line 25:558 token <41, (>
line 25:560 token <6, make-hash>
line 25:570 token <42,)>
line 25:572 token <42,)>
line 26:574 token <1, ; Headers list>
line 27:589 token <41, (>
line 27:591 token <3, define>
line 27:598 token <61, headers>
line 27:606 token <41, (>
line 27:608 token <61, list>
line 27:613 token <41, (>
line 27:615 token <62, make-header>
line 27:627 token <62, #"Cache-Control">
line 27:644 token <62, #"no-cache">
line 27:656 token <42,)>
line 27:658 token <42,)>
line 27:660 token <42,)>
line 29:663 token <1, ; Creating empty hash-maps for request handlers>
line 30:711 token <41, (>
line 30:713 token <21, map>
line 31:718 token <41, (>
line 31:719 token <22, λ>
line 31:721 token <41, (>
line 31:723 token <61, element>
line 31:731 token <42,)>
line 32:736 token <41, (>
line 32:738 token <7, hash-set!>
line 32:748 token <61, handlers>
line 32:757 token <61, element>
line 32:765 token <41, (>
line 32:767 token <6, make-hash>
line 32:777 token <42,)>
line 32:779 token <42,)>
line 32:781 token <42,)>
line 33:784 token <62, '(>
line 33:787 token <61, get>

line 33:791 token <61, post>
line 33:796 token <61, put>
line 33:800 token <61, delete>
line 33:807 token <42,)>
line 33:809 token <42,)>
line 35:812 token <1, ; Adds header to http response>
line 36:843 token <41, (>
line 36:845 token <3, define>
line 36:852 token <41, (>
line 36:854 token <62, response/add-header>
line 36:874 token <61, header>
line 36:881 token <42,)>
line 37:885 token <41, (>
line 37:887 token <28, set!>
line 37:892 token <61, headers>
line 37:900 token <41, (>
line 37:902 token <61, append>
line 37:909 token <61, headers>
line 37:917 token <41, (>
line 37:919 token <61, list>
line 37:924 token <61, header>
line 37:931 token <42,)>
line 37:933 token <42,)>
line 37:935 token <42,)>
line 37:937 token <42,)>
line 39:940 token <1, ; Creates cookie>
line 40:957 token <41, (>
line 40:959 token <3, define>
line 40:966 token <41, (>
line 40:968 token <62, response/add-cookie>
line 40:988 token <61, name>
line 40:993 token <61, value>
line 40:999 token <42,)>
line 41:1003 token <41, (>
line 41:1005 token <62, response/add-header>
line 42:1034 token <41, (>
line 42:1036 token <62, cookie->header>
line 43:1061 token <41, (>
line 43:1063 token <62, make-cookie>
line 43:1075 token <61, name>
line 43:1080 token <61, value>
line 43:1086 token <42,)>
line 43:1088 token <42,)>
line 43:1090 token <42,)>
line 43:1092 token <42,)>
line 45:1095 token <1, ; Remember request handler>
line 46:1122 token <1, ; Params are - request method(get/post/put/delete), path("/", "/hello/>
line 46:1192 token <54, :>
line 46:1193 token <62, id/")>

line 46:1199 token <13, and>
line 46:1203 token <61, callback>
line 47:1212 token <41, (>
line 47:1214 token <3, define>
line 47:1221 token <41, (>
line 47:1223 token <61, http/handler>
line 47:1236 token <61, method>
line 47:1243 token <61, path>
line 47:1248 token <61, proc>
line 47:1253 token <42,)>
line 48:1257 token <41, (>
line 48:1259 token <7, hash-set!>
line 48:1269 token <41, (>
line 48:1271 token <62, hash-ref>
line 48:1280 token <61, handlers>
line 48:1289 token <61, method>
line 48:1296 token <42,)>
line 48:1298 token <61, path>
line 48:1303 token <61, proc>
line 48:1308 token <42,)>
line 48:1310 token <42,)>
line 50:1313 token <1, ; GET request handler>
line 51:1335 token <41, (>
line 51:1337 token <3, define>
line 51:1344 token <41, (>
line 51:1346 token <61, http/get>
line 51:1355 token <61, path>
line 51:1360 token <61, proc>
line 51:1365 token <42,)>
line 52:1369 token <41, (>
line 52:1371 token <61, http/handler>
line 52:1384 token <62, 'get>
line 52:1389 token <61, path>
line 52:1394 token <61, proc>
line 52:1399 token <42,)>
line 52:1401 token <42,)>
line 54:1404 token <1, ; POST request handler>
line 55:1427 token <41, (>
line 55:1429 token <3, define>
line 55:1436 token <41, (>
line 55:1438 token <61, http/post>
line 55:1448 token <61, path>
line 55:1453 token <61, proc>
line 55:1458 token <42,)>
line 56:1462 token <41, (>
line 56:1464 token <61, http/handler>
line 56:1477 token <62, 'post>
line 56:1483 token <61, path>
line 56:1488 token <61, proc>

line 56:1493 token <42,)>
line 56:1495 token <42,)>
line 58:1498 token <1, ; PUT request handler>
line 59:1520 token <41, (>
line 59:1522 token <3, define>
line 59:1529 token <62, (http/put>
line 59:1539 token <61, path>
line 59:1544 token <61, proc>
line 59:1549 token <42,)>
line 60:1553 token <41, (>
line 60:1555 token <61, http/handler>
line 60:1568 token <62, 'put>
line 60:1573 token <61, path>
line 60:1578 token <61, proc>
line 60:1583 token <42,)>
line 60:1585 token <42,)>
line 62:1588 token <1, ; DELETE request handler>
line 63:1613 token <41, (>
line 63:1615 token <3, define>
line 63:1622 token <41, (>
line 63:1624 token <61, http/delete>
line 63:1636 token <61, path>
line 63:1641 token <61, proc>
line 63:1646 token <42,)>
line 64:1650 token <41, (>
line 64:1652 token <61, http/handler>
line 64:1665 token <62, 'delete>
line 64:1673 token <61, path>
line 64:1678 token <61, proc>
line 64:1683 token <42,)>
line 64:1685 token <42,)>
line 66:1688 token <1, ; Http path template to regexp translation>
line 67:1731 token <41, (>
line 67:1733 token <3, define>
line 67:1740 token <41, (>
line 67:1742 token <62, path->regexp>
line 67:1755 token <61, path>
line 67:1760 token <42,)>
line 68:1764 token <41, (>
line 68:1766 token <9, regexp>
line 69:1776 token <41, (>
line 69:1778 token <10, string-append>
line 70:1796 token <62, "^/">
line 71:1805 token <41, (>
line 71:1807 token <11, string-trim>
line 72:1824 token <41, (>
line 72:1826 token <62, regexp-replace*>
line 72:1842 token <62, #rx">
line 72:1846 token <54, :>

line 72:1847 token <62, [^>
line 72:1849 token <38, \>
line 72:1850 token <38, \>
line 72:1851 token <62, /]+">
line 72:1856 token <61, path>
line 72:1861 token <62, "([^/?]+)">
line 72:1872 token <42,)>
line 72:1874 token <62, "/">
line 72:1878 token <42,)>
line 73:1884 token <62, "(?>
line 73:1887 token <54, :>
line 73:1888 token <38, \>
line 73:1889 token <38, \>
line 73:1890 token <62, ?|$)">
line 73:1896 token <42,)>
line 73:1898 token <42,)>
line 73:1900 token <42,)>
line 75:1903 token <1, ; Set listen port>
line 76:1921 token <41, (>
line 76:1923 token <3, define>
line 76:1930 token <41, (>
line 76:1932 token <62, server/set-port>
line 76:1948 token <60, [>
line 76:1950 token <61, port>
line 76:1955 token <57, 8000>
line 76:1960 token <60,]>
line 76:1962 token <42,)>
line 77:1966 token <41, (>
line 77:1968 token <28, set!>
line 77:1973 token <61, env/port>
line 77:1982 token <61, port>
line 77:1987 token <42,)>
line 77:1989 token <42,)>
line 79:1992 token <1, ; Get request parameter>
line 80:2016 token <41, (>
line 80:2018 token <3, define>
line 80:2025 token <41, (>
line 80:2027 token <61, request/param>
line 80:2041 token <61, name>
line 80:2046 token <42,)>
line 81:2050 token <41, (>
line 81:2052 token <62, hash-ref>
line 81:2061 token <61, params>
line 81:2068 token <61, name>
line 81:2073 token <62, #f>
line 81:2076 token <42,)>
line 81:2078 token <42,)>
line 83:2081 token <1, ; Parse cookies to hash-map>
line 84:2109 token <41, (>

line 84:2111 token <3, define>
line 84:2118 token <41, (>
line 84:2120 token <62, server/parse-cookies>
line 84:2141 token <61, req>
line 84:2145 token <42,)>
line 85:2149 token <41, (>
line 85:2151 token <21, map>
line 85:2155 token <41, (>
line 85:2157 token <22, λ>
line 85:2159 token <41, (>
line 85:2161 token <61, cookie>
line 85:2168 token <42,)>
line 86:2179 token <41, (>
line 86:2181 token <7, hash-set!>
line 86:2191 token <61, cookies>
line 87:2219 token <41, (>
line 87:2221 token <62, client-cookie-name>
line 87:2240 token <61, cookie>
line 87:2247 token <42,)>
line 88:2269 token <41, (>
line 88:2271 token <62, client-cookie-value>
line 88:2291 token <61, cookie>
line 88:2298 token <42,)>
line 88:2300 token <42,)>
line 88:2302 token <42,)>
line 89:2311 token <41, (>
line 89:2313 token <62, request-cookies>
line 89:2329 token <61, req>
line 89:2333 token <42,)>
line 89:2335 token <42,)>
line 89:2337 token <42,)>
line 91:2340 token <1, ; Get cookie by name>
line 92:2361 token <41, (>
line 92:2363 token <3, define>
line 92:2370 token <41, (>
line 92:2372 token <61, request/cookie>
line 92:2387 token <61, name>
line 92:2392 token <42,)>
line 93:2396 token <41, (>
line 93:2398 token <62, hash-ref>
line 93:2407 token <61, cookies>
line 93:2415 token <61, name>
line 93:2420 token <62, #f>
line 93:2423 token <42,)>
line 93:2425 token <42,)>
line 95:2428 token <1, ; Make response>
line 96:2444 token <41, (>
line 96:2446 token <3, define>
line 96:2453 token <41, (>

line 96:2455 token <61, response/make>
line 96:2469 token <60, #>
line 96:2470 token <54, :>
line 96:2471 token <61, code>
line 96:2476 token <60, [>
line 96:2478 token <61, code>
line 96:2483 token <57, 200>
line 96:2487 token <60,]>
line 97:2512 token <60, #>
line 97:2513 token <54, :>
line 97:2514 token <61, message>
line 97:2522 token <60, [>
line 97:2524 token <61, message>
line 97:2532 token <62, #"OK">
line 97:2538 token <60,]>
line 98:2563 token <60, #>
line 98:2564 token <54, :>
line 98:2565 token <61, seconds>
line 98:2573 token <60, [>
line 98:2575 token <61, seconds>
line 98:2583 token <41, (>
line 98:2585 token <62, current-seconds>
line 98:2601 token <42,)>
line 98:2603 token <60,]>
line 99:2628 token <60, #>
line 99:2629 token <54, :>
line 99:2630 token <62, mime-type>
line 99:2640 token <60, [>
line 99:2642 token <62, mime-type>
line 99:2652 token <62, TEXT/HTML-MIME-TYPE>
line 99:2672 token <60,]>
line 100:2697 token <60, #>
line 100:2698 token <54, :>
line 100:2699 token <61, headers>
line 100:2707 token <60, [>
line 100:2709 token <61, headers>
line 100:2717 token <41, (>
line 100:2719 token <30, cond>
line 101:2769 token <60, [>
line 101:2771 token <41, (>
line 101:2773 token <61, empty?>
line 101:2780 token <61, headers>
line 101:2788 token <42,)>
line 101:2790 token <41, (>
line 101:2792 token <61, list>
line 101:2797 token <41, (>
line 101:2799 token <62, make-header>
line 101:2811 token <62, #"Cache-Control">
line 101:2828 token <62, #"no-cache">

line 101:2840 token <42,)>
line 101:2842 token <42,)>
line 101:2844 token <60,]>
line 102:2890 token <60, [>
line 102:2892 token <12, else>
line 102:2897 token <61, headers>
line 102:2905 token <60,]>
line 102:2907 token <42,)>
line 102:2909 token <60,]>
line 103:2934 token <61, content>
line 103:2942 token <42,)>
line 104:2946 token <41, (>
line 104:2948 token <61, response/full>
line 104:2962 token <61, code>
line 105:2984 token <61, message>
line 106:3009 token <61, seconds>
line 107:3034 token <62, mime-type>
line 108:3061 token <61, headers>
line 109:3086 token <41, (>
line 109:3088 token <61, list>
line 109:3093 token <41, (>
line 109:3095 token <62, string->bytes/utf>
line 109:3113 token <62, content)>
line 109:3122 token <42,)>
line 109:3124 token <42,)>
line 109:3126 token <42,)>
line 111:3129 token <1, ; 404 Response>
line 112:3144 token <41, (>
line 112:3146 token <3, define>
line 112:3153 token <41, (>
line 112:3155 token <61, response/404>
line 112:3168 token <42,)>
line 113:3172 token <41, (>
line 113:3174 token <61, response/make>
line 113:3188 token <60, #>
line 113:3189 token <54, :>
line 113:3190 token <61, code>
line 113:3195 token <57, 404>
line 113:3199 token <62, "Page>
line 113:3205 token <61, not>
line 113:3209 token <62, found">
line 113:3216 token <42,)>
line 113:3218 token <42,)>
line 115:3221 token <1, ; Find handler >
line 116:3237 token <41, (>
line 116:3239 token <3, define>
line 116:3246 token <41, (>
line 116:3248 token <62, server/find-handler>
line 116:3268 token <61, req>

line 116:3272 token <42,)>
line 117:3276 token <41, (>
line 117:3278 token <61, let*>
line 117:3283 token <41, (>
line 117:3285 token <41, (>
line 117:3287 token <61, path>
line 117:3292 token <41, (>
line 117:3294 token <62, regexp-replace*>
line 117:3310 token <62, #rx">
line 117:3314 token <38, \>
line 117:3315 token <38, \>
line 117:3316 token <62, ?.*">
line 117:3321 token <41, (>
line 117:3323 token <62, url->string>
line 117:3335 token <41, (>
line 117:3337 token <62, request-uri>
line 117:3349 token <61, req>
line 117:3353 token <42,)>
line 117:3355 token <42,)>
line 117:3357 token <62, "">
line 117:3360 token <42,)>
line 117:3362 token <42,)>
line 118:3373 token <41, (>
line 118:3375 token <61, method>
line 119:3392 token <41, (>
line 119:3394 token <17, case>
line 119:3399 token <41, (>
line 119:3401 token <62, request-method>
line 119:3416 token <62, req)>
line 120:3433 token <60, [>
line 120:3435 token <41, (>
line 120:3437 token <62, #"GET">
line 120:3444 token <42,)>
line 120:3446 token <62, 'get>
line 120:3451 token <60,]>
line 121:3465 token <60, [>
line 121:3467 token <41, (>
line 121:3469 token <62, #"POST">
line 121:3477 token <42,)>
line 121:3479 token <62, 'post>
line 121:3485 token <60,]>
line 122:3499 token <60, [>
line 122:3501 token <41, (>
line 122:3503 token <62, #"PUT">
line 122:3510 token <42,)>
line 122:3512 token <62, 'put>
line 122:3517 token <60,]>
line 123:3531 token <60, [>
line 123:3533 token <41, (>

line 123:3535 token <62, #"DELETE">
line 123:3545 token <42,)>
line 123:3547 token <62, 'delete>
line 123:3555 token <60,]>
line 123:3557 token <42,)>
line 123:3559 token <42,)>
line 124:3570 token <41, (>
line 124:3572 token <62, handler-key>
line 125:3594 token <41, (>
line 125:3596 token <61, findf>
line 125:3602 token <41, (>
line 125:3604 token <22, λ>
line 125:3606 token <41, (>
line 125:3608 token <61, key>
line 125:3612 token <42,)>
line 125:3615 token <41, (>
line 125:3617 token <62, regexp-match>
line 125:3630 token <41, (>
line 125:3632 token <62, path->regexp>
line 125:3645 token <61, key>
line 125:3649 token <42,)>
line 125:3651 token <61, path>
line 125:3656 token <42,)>
line 125:3658 token <42,)>
line 126:3677 token <41, (>
line 126:3679 token <62, hash-keys>
line 126:3689 token <41, (>
line 126:3691 token <62, hash-ref>
line 126:3700 token <61, handlers>
line 126:3709 token <61, method>
line 126:3716 token <42,)>
line 126:3718 token <42,)>
line 126:3720 token <42,)>
line 126:3722 token <42,)>
line 126:3724 token <42,)>
line 127:3730 token <41, (>
line 127:3732 token <61, list>
line 127:3737 token <61, path>
line 127:3742 token <62, handler-key>
line 127:3754 token <61, method>
line 127:3761 token <42,)>
line 127:3763 token <42,)>
line 127:3765 token <42,)>
line 129:3768 token <1, ; Makes response>
line 130:3785 token <41, (>
line 130:3787 token <3, define>
line 130:3794 token <41, (>
line 130:3796 token <61, server/response>
line 130:3812 token <62, path-handler-method>

line 130:3832 token <61, req>
line 130:3836 token <42,)>
line 131:3840 token <41, (>
line 131:3842 token <61, let*>
line 131:3847 token <41, (>
line 131:3849 token <41, (>
line 131:3851 token <61, path>
line 131:3856 token <41, (>
line 131:3858 token <14, car>
line 131:3862 token <62, path-handler-method>
line 131:3882 token <42,)>
line 131:3884 token <42,)>
line 132:3895 token <41, (>
line 132:3897 token <62, handler-key>
line 132:3909 token <41, (>
line 132:3911 token <61, cadr>
line 132:3916 token <62, path-handler-method>
line 132:3936 token <42,)>
line 132:3938 token <42,)>
line 133:3949 token <41, (>
line 133:3951 token <61, method>
line 133:3958 token <41, (>
line 133:3960 token <61, caddr>
line 133:3966 token <62, path-handler-method>
line 133:3986 token <42,)>
line 133:3988 token <42,)>
line 133:3990 token <42,)>
line 134:3996 token <41, (>
line 134:3998 token <17, case>
line 134:4003 token <62, handler-key>
line 135:4021 token <60, [>
line 135:4023 token <41, (>
line 135:4025 token <62, #f>
line 135:4028 token <42,)>
line 135:4030 token <41, (>
line 135:4032 token <61, response/404>
line 135:4045 token <42,)>
line 135:4047 token <60,]>
line 136:4055 token <60, [>
line 136:4057 token <12, else>
line 137:4069 token <41, (>
line 137:4071 token <61, let*>
line 137:4076 token <41, (>
line 137:4078 token <41, (>
line 137:4080 token <61, keys>
line 137:4085 token <41, (>
line 137:4087 token <21, map>
line 138:4112 token <41, (>
line 138:4114 token <22, λ>

line 138:4116 token <41, (>
line 138:4118 token <61, match>
line 138:4124 token <42,)>
line 138:4126 token <41, (>
line 138:4128 token <62, string->symbol>
line 138:4143 token <41, (>
line 138:4145 token <61, substring>
line 138:4155 token <61, match>
line 138:4161 token <57, 2>
line 138:4163 token <42,)>
line 138:4165 token <42,)>
line 138:4167 token <42,)>
line 139:4190 token <41, (>
line 139:4192 token <62, regexp-match*>
line 139:4206 token <62, #rx"/>
line 139:4211 token <54, :>
line 139:4212 token <62, ([^>
line 139:4215 token <38, \>
line 139:4216 token <38, \>
line 139:4217 token <62, /]+)">
line 139:4223 token <62, handler-key>
line 139:4235 token <42,)>
line 139:4237 token <42,)>
line 139:4239 token <42,)>
line 140:4255 token <41, (>
line 140:4257 token <61, pairs>
line 141:4278 token <41, (>
line 141:4280 token <61, for/list>
line 141:4289 token <41, (>
line 141:4291 token <60, [>
line 141:4293 token <61, key>
line 141:4297 token <61, keys>
line 141:4302 token <60,]>
line 142:4330 token <60, [>
line 142:4332 token <61, val>
line 142:4336 token <41, (>
line 142:4338 token <15, cdr>
line 142:4342 token <41, (>
line 142:4344 token <62, regexp-match>
line 142:4357 token <41, (>
line 142:4359 token <62, path->regexp>
line 142:4372 token <62, handler-key>
line 142:4384 token <42,)>
line 142:4386 token <61, path>
line 142:4391 token <42,)>
line 142:4393 token <42,)>
line 142:4395 token <60,]>
line 142:4397 token <42,)>
line 143:4416 token <41, (>

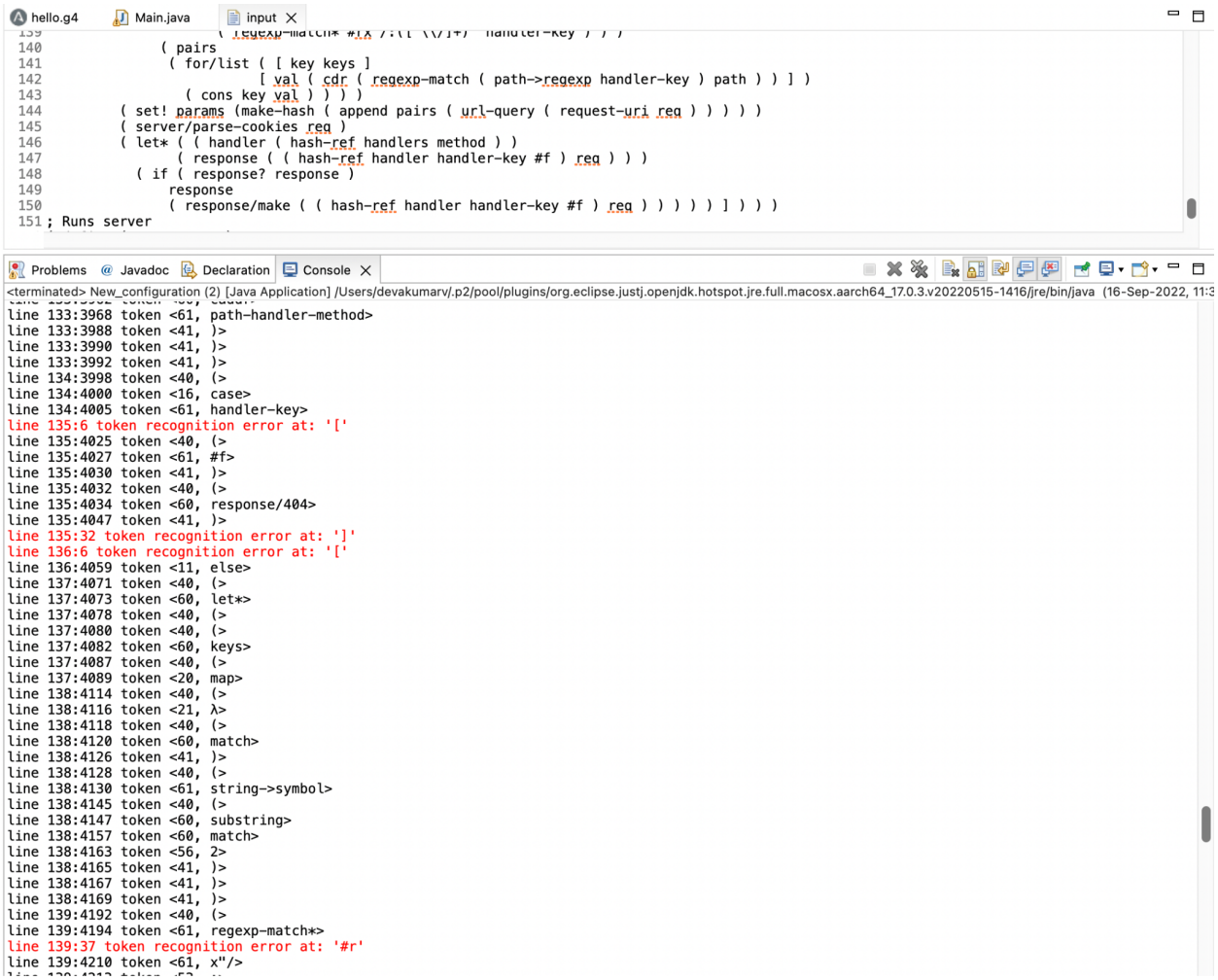
line 143:4418 token <61, cons>
line 143:4423 token <61, key>
line 143:4427 token <61, val>
line 143:4431 token <42,)>
line 143:4433 token <42,)>
line 143:4435 token <42,)>
line 143:4437 token <42,)>
line 144:4448 token <41, (>
line 144:4450 token <28, set!>
line 144:4455 token <61, params>
line 144:4462 token <62, (make-hash>
line 144:4473 token <41, (>
line 144:4475 token <61, append>
line 144:4482 token <61, pairs>
line 144:4488 token <41, (>
line 144:4490 token <62, url-query>
line 144:4500 token <41, (>
line 144:4502 token <62, request-uri>
line 144:4514 token <61, req>
line 144:4518 token <42,)>
line 144:4520 token <42,)>
line 144:4522 token <42,)>
line 144:4524 token <42,)>
line 144:4526 token <42,)>
line 145:4537 token <41, (>
line 145:4539 token <62, server/parse-cookies>
line 145:4560 token <61, req>
line 145:4564 token <42,)>
line 146:4575 token <41, (>
line 146:4577 token <61, let*>
line 146:4582 token <41, (>
line 146:4584 token <41, (>
line 146:4586 token <61, handler>
line 146:4594 token <41, (>
line 146:4596 token <62, hash-ref>
line 146:4605 token <61, handlers>
line 146:4614 token <61, method>
line 146:4621 token <42,)>
line 146:4623 token <42,)>
line 147:4641 token <41, (>
line 147:4643 token <61, response>
line 147:4652 token <41, (>
line 147:4654 token <41, (>
line 147:4656 token <62, hash-ref>
line 147:4665 token <61, handler>
line 147:4673 token <62, handler-key>
line 147:4685 token <62, #f>
line 147:4688 token <42,)>
line 147:4690 token <61, req>

line 147:4694 token <42,)>
line 147:4696 token <42,)>
line 147:4698 token <42,)>
line 148:4711 token <41, (>
line 148:4713 token <2, if>
line 148:4716 token <41, (>
line 148:4718 token <61, response?>
line 148:4728 token <61, response>
line 148:4737 token <42,)>
line 149:4754 token <61, response>
line 150:4778 token <41, (>
line 150:4780 token <61, response/make>
line 150:4794 token <41, (>
line 150:4796 token <41, (>
line 150:4798 token <62, hash-ref>
line 150:4807 token <61, handler>
line 150:4815 token <62, handler-key>
line 150:4827 token <62, #f>
line 150:4830 token <42,)>
line 150:4832 token <61, req>
line 150:4836 token <42,)>
line 150:4838 token <42,)>
line 150:4840 token <42,)>
line 150:4842 token <42,)>
line 150:4844 token <42,)>
line 150:4846 token <60,]>
line 150:4848 token <42,)>
line 150:4850 token <42,)>
line 150:4852 token <42,)>
line 151:4854 token <1, ; Runs server>
line 152:4868 token <41, (>
line 152:4870 token <3, define>
line 152:4877 token <41, (>
line 152:4879 token <61, server/run>
line 152:4890 token <42,)>
line 153:4894 token <41, (>
line 153:4896 token <61, serve/servlet>
line 154:4913 token <41, (>
line 154:4915 token <22, λ>
line 154:4917 token <41, (>
line 154:4919 token <61, req>
line 154:4923 token <42,)>
line 155:4930 token <41, (>
line 155:4932 token <28, set!>
line 155:4937 token <61, headers>
line 155:4945 token <41, (>
line 155:4947 token <61, list>
line 155:4952 token <62, (make-header>
line 155:4965 token <62, #"Cache-Control">

line 155:4982 token <62, #"no-cache">
line 155:4994 token <42,)>
line 155:4996 token <42,)>
line 155:4998 token <42,)>
line 156:5005 token <41, (>
line 156:5007 token <28, set!>
line 156:5012 token <61, cookies>
line 156:5020 token <41, (>
line 156:5022 token <6, make-hash>
line 156:5032 token <42,)>
line 156:5034 token <42,)>
line 157:5041 token <41, (>
line 157:5043 token <61, server/response>
line 158:5065 token <41, (>
line 158:5067 token <62, server/find-handler>
line 158:5087 token <61, req>
line 158:5091 token <42,)>
line 159:5099 token <61, req>
line 159:5103 token <42,)>
line 159:5105 token <42,)>
line 159:5107 token <42,)>
line 159:5109 token <42,)>

Sample Output of Lexical Errors :

Even after lexical errors , compiler can proceed to the eof with tokenizer


hello.g4
Main.java
input
(pairs
(for/list ([key keys]
[val (cdr (regexp-match (path->regexp handler-key) path))])
(cons key val))))
(set! params (make-hash (append pairs (url-query (request-uri req)))))
(server/parse-cookies req)
(let* ((handler (hash-ref handlers method))
(response ((hash-ref handler handler-key #f) req)))
(if (response? response)
response
(response/make ((hash-ref handler handler-key #f) req))))])))
; Runs server
Problems
Javadoc
Declaration
Console
<terminated> New_configuration (2) [Java Application] /Users/devakumarv/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.3.v20220515-1416/jre/bin/java (16-Sep-2022, 11:3
line 133:3968 token <61, path-handler-method>
line 133:3988 token <41,)>
line 133:3990 token <41,)>
line 133:3992 token <41,)>
line 134:3998 token <40, (>
line 134:4000 token <16, case>
line 134:4005 token <61, handler-key>
line 135:6 token recognition error at: '['
line 135:4025 token <40, (>
line 135:4027 token <61, #f>
line 135:4030 token <41,)>
line 135:4032 token <40, (>
line 135:4034 token <60, response/404>
line 135:4047 token <41,)>
line 135:32 token recognition error at: ']'
line 136:6 token recognition error at: '['
line 136:4059 token <11, else>
line 137:4071 token <40, (>
line 137:4073 token <60, let*>
line 137:4078 token <40, (>
line 137:4080 token <40, (>
line 137:4082 token <60, keys>
line 137:4087 token <40, (>
line 137:4089 token <20, map>
line 138:4114 token <40, (>
line 138:4116 token <21, λ>
line 138:4118 token <40, (>
line 138:4120 token <60, match>
line 138:4126 token <41,)>
line 138:4128 token <40, (>
line 138:4130 token <61, string->symbol>
line 138:4145 token <40, (>
line 138:4147 token <60, substring>
line 138:4157 token <60, match>
line 138:4163 token <56, 2>
line 138:4165 token <41,)>
line 138:4167 token <41,)>
line 138:4169 token <41,)>
line 139:4192 token <40, (>
line 139:4194 token <61, regexp-match*>
line 139:37 token recognition error at: '#r'
line 139:4210 token <61, x"/>