

Grammar and semantic actions for the zimbra2gmail compiler

by Antonio Augusto de Cintra Batista <antonio.a.c.batista@gmail.com>

January of 2018

This is a regular grammar is for a top-down predictive parser.

*The semantic actions are annotated between <>: **<semantic-action>***

zimbra2gmail ::= CERCA NAME *<emailAddr = lexeme>* ID require { *<gamUser = "gam user %s filter ", emailAddr>* filters }

require ::= ZIMBRAMAILSIEVEScript DOISPONTOS REQUIRE ABRECOLCHETE requires-rep { VIRGULA requires-rep } FECHACOLCHETE PONTOEVIRGULA

requires-rep ::= ASPAS (FILEINTO | REJECT | TAG | FLAG) ASPAS

filters ::= CERCA *<thisFilterName = lexeme ; msg = "# " + lexeme ; boo(msg)>* IEXTENDED conditions action *<if len(gamAction>0) then { gamCommand = gamUser + gamCondition + gamAction ; boo(gamCommand) }> <if hasStop == "y" then gamCommand = gamUser + gamCondition + "label zLabelToStopActions" ; boo(gamCommand)> <boo("")>*

conditions ::= *<gamCondition = "haswords \"-label:zLabelToStopActions "> (<filterEnabled = "y"> IF | <filterEnabled = "n"> DISABLED_IF) (<conditionType = "anyof"> ANYOF | <conditionType = "allof"> ALLOF)* ABREPARENTESIS expr FECHAPARENTESIS

expr ::= *<gamCondition += "(">* expr-rep { *<gamCondition += "OR ">* VIRGULA expr-rep } *<gamCondition += ")"\" ">*

expr-rep ::= [*<gamCondition += "-">* NOT] (expr-header | expr-address | expr-body | expr-date | expr-bulk | expr-distribution-list | expr-to-me)

expr-header ::= HEADER DOISPONTOS (IS | CONTAINS | MATCHES) ABRECOLCHETE header { VIRGULA header } FECHACOLCHETE ASPAS *<replace(gamCondition, "xYz", lexeme)>* IEXTENDED ASPAS

expr-address ::= ADDRESS DOISPONTOS (ALL | DOMAIN) DOISPONTOS (IS | CONTAINS | MATCHES) DOISPONTOS COMPARATOR ASPAS ID PONTOEVIRGULA ASCII_CASEMAP ASPAS ABRECOLCHETE *<gamCondition += "(">* header { *<gamCondition += "OR ">* VIRGULA header } *<gamCondition += ")" ">* FECHACOLCHETE ASPAS *<replace(gamCondition, "xYz", lexeme)>* IEXTENDED ASPAS

expr-body ::= BODY *<gamCondition += "">* DOISPONTOS CONTAINS ASPAS *<gamCondition += lexeme ; gamCommand += "" ">* IEXTENDED ASPAS

expr-date ::= DATE DOISPONTOS (AFTER | BEFORE) ASPAS IDEXTENDED ASPAS

expr-to-me ::= ME DOISPONTOS IN ASPAS to-me ASPAS

expr-distribution-list ::= LIST

expr-bulk ::= BULK

to-me ::= <gamCondition += "("> (<gamCondition += "to:" + emailAddr + "" "> TO [<gamCondition += "OR
cc:" + emailAddr + "" "> VIRGULA CC] | <gamCondition += "cc:" + emailAddr + "" "> CC) <gamCondition
+= ")" ">

header ::= ASPAS (<gamCondition += "to:"> TO | <gamCondition += "cc:"> CC | <gamCondition += "from:">
FROM | <gamCondition += "subject:"> SUBJECT | <gamCondition += lexeme + ":"> HEADER | ID) ASPAS
<gamCondition += "xYz' ">

action ::= <gamAction = ""> ABRECHAVES { move | tag | flag | discard | keep | redirect }1 <hasStop = "n"> [
<hasStop = "y"> STOP PONTOEVIRGULA] FECHACHAVES

move ::= FILEINTO ASPAS <gamLabel = "label "" ; cleanLabel(lexeme) ; gamLabel += lexeme ; gamLabel += ""
archive "" ; gamCommand = gamUser + gamCondition + gamLabel ; boo(gamCommand)> IDEXTENDED ASPAS
PONTOEVIRGULA

tag ::= TAG ASPAS <gamLabel = "label myTaggedMessages/" ; cleanLabel(lexeme) ; gamLabel += lexeme ;
gamLabel += "" "" ; gamCommand = gamUser + gamCondition + gamLabel ; boo(gamCommand)> IDEXTENDED
ASPAS PONTOEVIRGULA

flag ::= FLAG ASPAS (<gamAction += "markread "> READ) ASPAS PONTOEVIRGULA

discard ::= <gamAction += "trash "> DISCARD PONTOEVIRGULA

keep ::= <gamLabel = "label inbox "" ; gamCommand = gamUser + gamCondition + gamLabel ;
boo(gamCommand)> KEEP PONTOEVIRGULA

redirect ::= REDIRECT ASPAS <gamAction += "forward "" ; gamAction += lexeme ; gamAction += "" ">
IDEXTENDED ASPAS PONTOEVIRGULA { REDIRECT ASPAS <gamAction += "forward "" ; gamAction +=
lexeme ; gamAction += "" "> IDEXTENDED ASPAS PONTOEVIRGULA } [keep <if false gamAction += "trash
>]