

Table 1: Tokens in Go subset accepted by grammar

| Token | Token Name | Token Class | Informal Description | Formal Description (regex) | Example(s) |
|-----------------------------|---------------|------------------------------------|---------------------------------------------------------------------------------|---------------------------------------|----------------------|
| Identifier | tok_id | identifier | at least one char, starts with letter, alphanumeric with _ | $[a-zA-Z]([a-zA-Z0-9_])^*$ | i, myId, _abc, _var1 |
| Package Identifier | tok_pid | identifier | at least one char, starts with letter, alphanumeric, cannot be blank identifier | $[a-zA-Z]([a-zA-Z0-9_])^+$ | main, _abc |
| (| tok_lparen | punctuation | - | $\{$ | - |
|) | tok_rparen | punctuation | - | $\}$ | - |
| { | tok_lcurly | punctuation | - | $\{$ | - |
| } | tok_rcurly | punctuation | - | $\}$ | - |
| ; | tok_semicolon | punctuation | - | $;$ | - |
| Main | tok_main | identifier (with semantic meaning) | - | "main" | - |
| Package Declaration | tok_package | reserved keyword | - | "package" | - |
| Function Declaration | tok_func | reserved keyword | - | "func" | - |
| EOF | tok_eof | reserved | - | "\Z" | - |
| Import | tok_imp | reserved keyword | - | "import" | - |
| Alphanumeric string literal | tok_string | string | numbers or letters delimited by " | $\backslash"[a-zA-Z0-9_+]\backslash"$ | "abc", "01234" |