

```

1 grammar CSVAG;
2
3 @header {
4 package ag;
5 import java.util.*;
6 }
7
8 file
9 locals [int i = 0]
10     : hdr ( rows += row[$hdr.text.split(",")] { $i++; })+ {
11         System.out.println("Totally " + $i + " rows");
12         for (RowContext r : $rows) {
13             System.out.println("Row token interval : " + r.
getSourceInterval());
14         }
15     }
16 ;
17
18 hdr : row[null] {
19     System.out.println("header: " + $text.trim());
20 };
21
22 row[String[] columns] returns [Map<String, String> values]
23 locals [int col = 0]
24 @init {
25     $values = new LinkedHashMap<>();
26 }
27 @after {
28     if ($values.size() > 0) {
29         System.out.println("values = " + $values);
30     }
31 }
32 : field {
33     if ($columns != null) {
34         $values.put($columns[$col++].trim(), $field.text.trim());
35     }
36 } (',' field
37 {
38     if ($columns != null) {
39         $values.put($columns[$col++].trim(), $field.text.trim
40         ());
41     }
42 })* '\r'? '\n'
43 ;
44
45 field : ID | NUMBER ;
46
47 ID : [a-z]+ ;
48 NUMBER : [0-9]+ ;
49 WS : [ \t]+ -> skip;

```