```
grammar CSVAG;

@header {
package ag;
import java.util.*;
}

file
locals [int i = 0]
    : hdr ( rows += row[$hdr.text.split(",")] { $i++; })+ {
            System.out.println("Totally " + $i + " rows");
            for (RowContext r : $rows) {
                System.out.println("Row token interval : " + r.getSourceInterval());
            }
        }
    ;

hdr : row[null] {
        System.out.println("header: " + $text.trim());
    };

row[String[] columns] returns [Map<String, String> values]
locals [int col = 0]
@init {
    $values = new LinkedHashMap<>();
}
@after {
    if ($values.size() > 0) {
        System.out.println("values = " + $values);
    }
}
    : field {
        if ($columns != null) {
            $values.put($columns[$col++].trim(), $field.text.trim());
        }
    } (',' field
        {
            if ($columns != null) {
                $values.put($columns[$col++].trim(), $field.text.trim());
            }
        }
    )* '\r'? '\n'
    ;

field : ID | NUMBER ;

ID : [a-z]+ ;
NUMBER : [0-9]+ ;
WS : [ \t]+ -> skip;
```