

# Regular Expressions

```
([^\-]*)\-|s([^\s]*)s([^\"]*)\"([^\s]*)s([^\"]*)\"s([0-9]*)s([0-9]*).*
```

Apply on access.log file:

- `([^\-]*)` // characters up to dash become first field
- `\-|s` // dash space
- `([^\s]*)` // characters up to next space become second field
- `\s` // space
- `([^\"]*)` // characters up to quote become third field
- `\"` // quote
- `([^\s]*)` // characters up to space become fourth field
- `\s` // space
- `([^\"]*)` // characters up to quote become fifth field
- `\\"s` // quote and space
- `([0-9]*)` // digits become sixth field
- `\s` // space
- `([0-9]*)` // digits become seventh field
- `.*` // anything else on the line ignored

To filter so you only see certain codes, eg. 403 in the sixth field:

- `([^\-]*)\-|s([^\s]*)s([^\"]*)\"([^\s]*)s([^\"]*)\"s(403)s([0-9]*).*`

if you don't want to see the 403 as a field remove the grouping parenthesis:

- `([^\-]*)\-|s([^\s]*)s([^\"]*)\"([^\s]*)s([^\"]*)\"s403s([0-9]*).*`

To use variable substitution, insert a `${code}`

- `([^\-]*)\-|s([^\s]*)s([^\"]*)\"([^\s]*)s([^\"]*)\"s(${code})s([0-9]*).*`

I've reinserted the grouping parenthesis in this example, so the actual code value becomes a field.