* JQL Basics
  + field

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Environment | Resolved | Updated |
| Attachments | Epic Link | Original Estimate | Voter |
| Category | Filter | Parent | Remaining Estimate |
| Comment | Fix Version | Priority | Text |
| Component | Issue Key | Project | Type |
| Created | LastViewed | Votes | Time Spent |
| Due | Level | Watcher | Resolution |
| Affected Version | Work Log Comment | Watchers | Status |
| Assignee | Work Log Date | Work Log Author | Summary |
| Sprint | Work Ratio | Creator | Reporter |

* + operator
    - matches: EQUALS =, NOT EQUALS !=, GREATER THAN >, GREAT THAN EQUALS >=, LESS THAN <, LESS THAN EQUALS <=
    - content: IN, NOT IN, CONTAINS -, DOES NOT CONTAIN !-
    - current: IS, IS NOT
    - historic: WAS, WAS IN, WAS NOT IN, WAS NOT, CHANGED
  + value
* "Filters" 🡪 "Advanced Issue Search" 🡪 "Switch to JQL"
* Keywords (very similar to SQL)
  + **and, or, not**
  + **empty**: can use this keyword together with a supported operator in order to filter for issues that have no value set for the given field
  + **null**: field has never been set
  + **order**
* Using Time in JQL
  + updated > "2019-01-01": updated on Jan. 1st, 2019
  + updated > -2w: updated two weeks ago
  + updated > -"4w 3d": updated four weeks and three days ago
  + timeSpent > 10d: spent ten days
* Functions
  + startOfDay()
  + startOfWeek()
  + startOfMonth()
  + startOfYear()
  + endOfDay()
  + endOfWeek()
  + endOfMonth()
  + endOfYear()
  + now()
  + currentLogin()
  + lastLogin()
* Text Search
  + description ~ "web"
    - any issue that contains the word *web* regardless of where in the field it was written
  + description ~ "web\*"
    - issues that contains words that begin with the word *web* (e.g. *webs*, *website*, etc.)
  + description ~ "w??"
    - issues that have three-letter words that begin with the letter *w* (e.g. *win*, *web*)
* Filters
  + Saved JQL queries to reuse
  + filters can also be shared between users
* More Examples
  + project=NTP and issuetype=bug
    - returns *bugs* in my project *NTP*
  + project=NTP and issuetype=bug and status="to do"
    - returns *bugs* in my project *NTP* with a status of *To Do*
* Reports
* Webhooks
  + user defined HTTP callback
  + e.x. If you want to alert your remote application when an issue has been closed or when a sprint has been completed
  + webhooks send data as JSON objects
  + requestbin