Craeg Strong

CTO, Ariel Partners
AKT, AKC, FLG, CSM, CSP, CSPO,
ITILv3, PMI-ACP, PMP, CLP, SPC,
ICP-ACC, ICP-ATF, PSM-II, PSK
CEH, ACP-JSW, AC-JPA, ACP-MJCP

www.arielpartners.com
cstrong@arielpartners.com
@ckstrong1

- Software Development since 1988

- Large Commercial & Government Projects

- Enterprise Architect / DevOps Engineer

- Portfolio Kanban / Flight Levels Coach

- Agile PMO / Agile Governance at Scale

- Certified Ethical Hacker

- New York & Washington DC Area

ARIEL PARTNERS
MISSION CRITICAL AGILE™

Why Learn Jira Query Language?
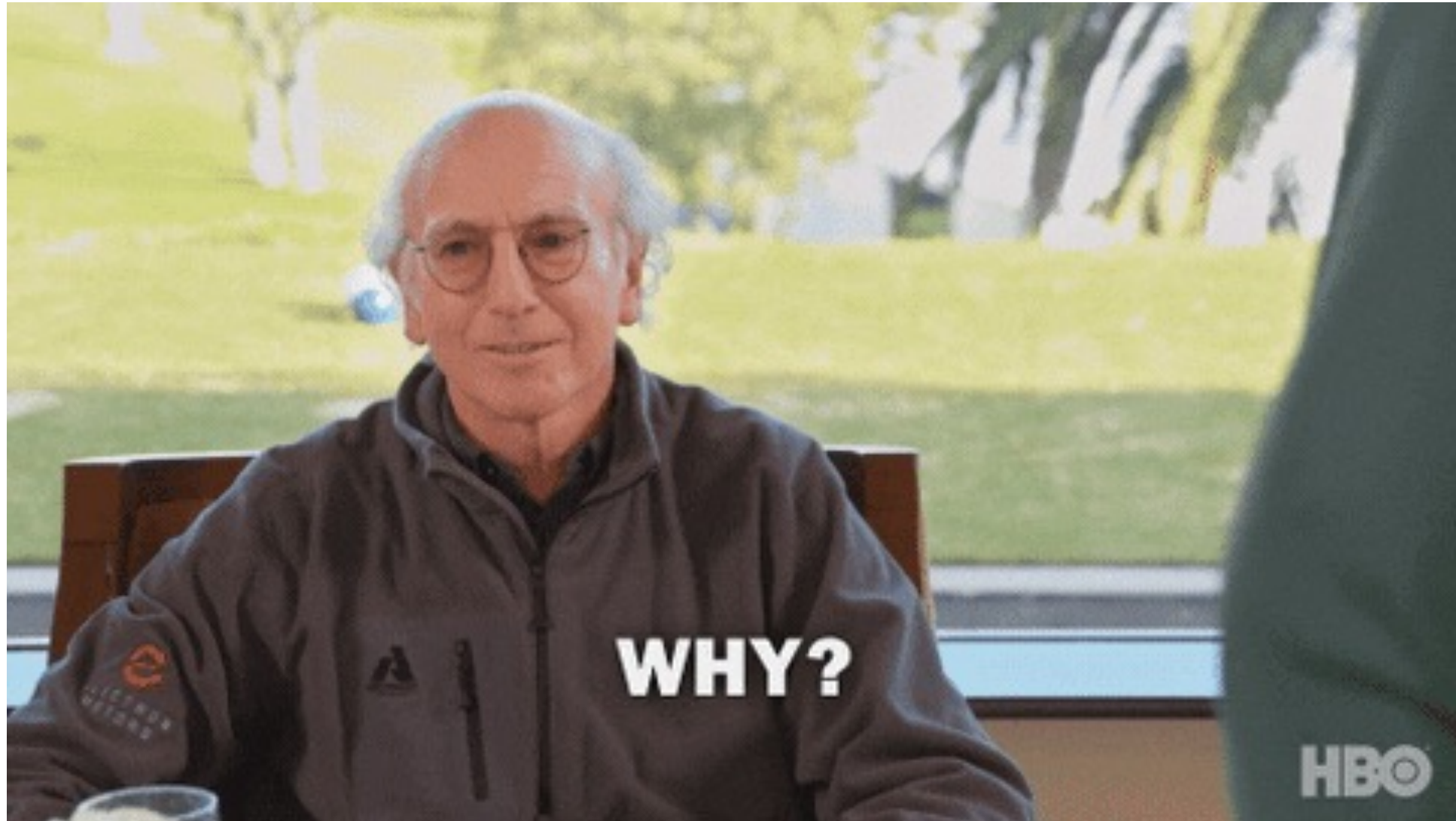
JQL: The Foundation of All Things Jira

Reports

Dashboards

Team Boards

Identify Problems

Find Duplicates

Analyze Trends

Agenda

1. Basic Search Review

2. Getting Started with JQL

3. Searching Field Values

4. Visualizing Results in Dashboards

5. Searching History, Dates, and SLAs

6. Putting it All Together

7. Wrapping Up

Want to Follow Along? Login here!

Browser	Google Chrome (Edge, Safari, or Firefox are fine too!)
URL	http://44.215.126.136:8080
Login	jira
Password	Ariel@2022!

Basic Search Review

Issue Navigator Screen

- Saved searches (aka "filters") show up on the left
- Column selector is on the right
- ☰ ⌄ Menu switches between list view and master-detail
- Export to Excel via "Export" button is on top right

Caution: *Once you go to the Issue Navigator, you are no longer in a Project. You will have to reselect your project from the "Projects" menu to return to your project.*

Basic Search

- Simple UI that allows you to find issues
  - ## Select a field and then enter the search criteria
- Can search all text fields or specific text fields
- Date searches can be for fixed or relative date ranges

Add more fields

Switch to JQL

Full-text search

Select one or more Status values

Controlling the Output

- You can control which fields are displayed for your search results

- You can pick any number of fields to be displayed from the "Columns" dropdown

- You can change the order of columns by dragging and dropping them

- If all the columns won't fit on the screen, there will be a scroll bar on the bottom

- You can sort on any field just by clicking on it.

- Clicking again reverses the sort
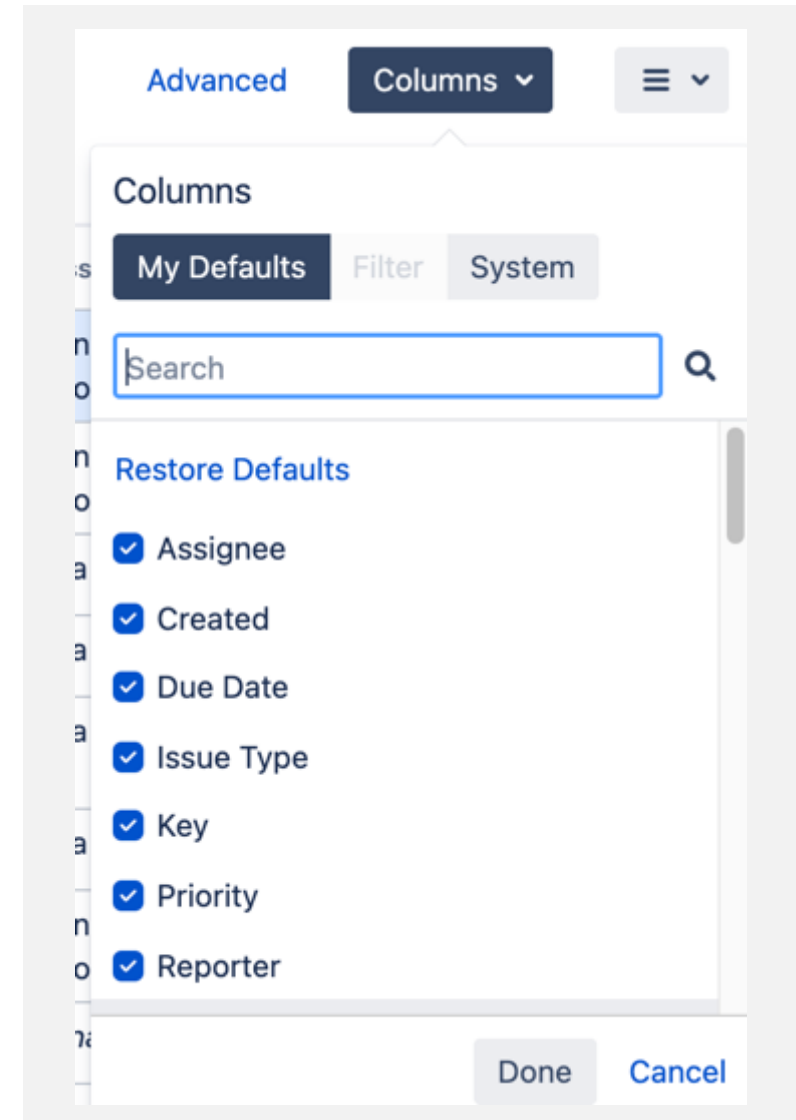
Exercise: Specifying and Rearranging Columns

- First, clear all dropdowns except for Project: Headquarters Help Desk

- Using the column menu, add and remove columns

- Then use drag and drop until you the columns are in *exactly* this order
  - Assignee
  - Created (when this issue was entered into Jira)
  - Due Date (when this issue is due)
  - Issue Type (Incident, Service Request, etc.)
  - Key (the issue's unique key)
  - Resolved (when this issue was "resolved" or completed)
  - Status (current status of issue)
  - Summary
  - Updated (when this issue was last updated)

- Sort in ascending order by Due (oldest at the top)

- *How many Issues with Due Dates are yet to be resolved?*
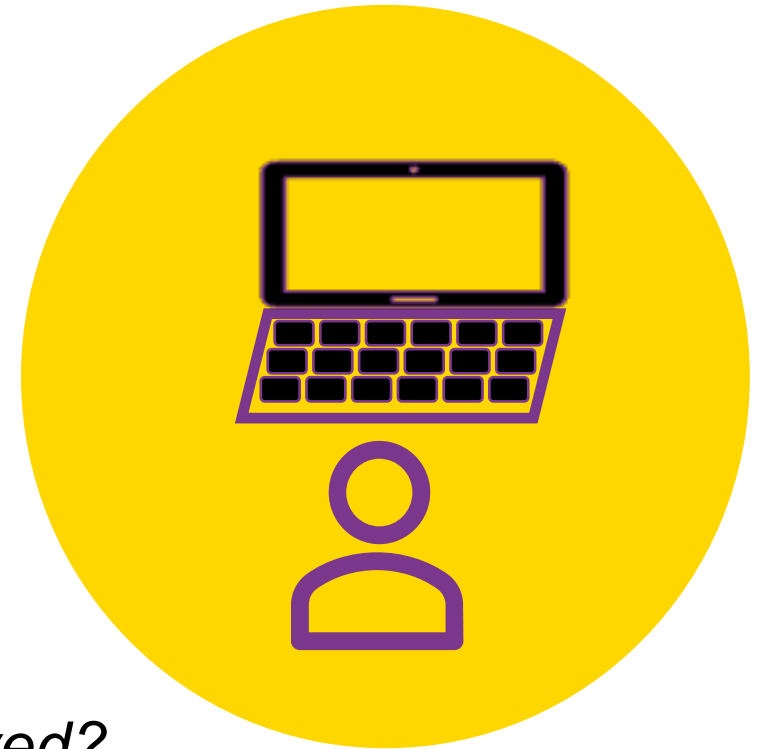
Homework Exercise: Basic Search

Find the most recently resolved issue that involves an access request that was **denied**. Who worked on it? When exactly was it completed? What is the issue key?

Hints:

- Select Resolved Status
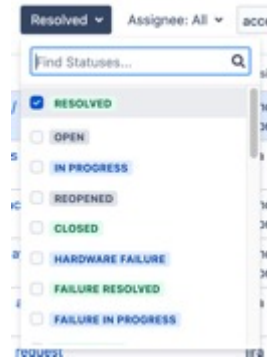
- Full-text search for access
- Select "Resolution" from "More" dropdown and select "Won't Do"
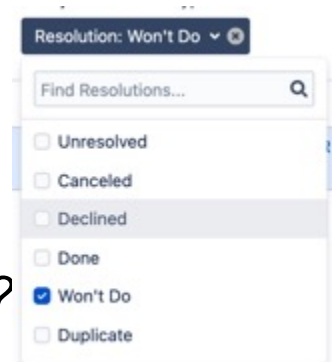
Extra credit: What does the JQL query look like?

Becoming a JQL Power User

Mastering JQL can take time.   Here is the easy way:

1. *Whenever you create a basic query, click the "Advanced" button to see the JQL version. Then click back, change something, and check the JQL again (for example, sort on a column).*

2. *Maintain a cheat sheet of favorite JQL queries. Whenever you see a great query, copy it into your cheat sheet.*

3. *As you build a JQL query, make use of Jira's auto-completion capability- it almost writes itself!*

4. *Pay attention to the green valid checkbox; if it turns red you can backtrack until its green again*

Before you know it, JQL will start to feel comfortable and familiar!

Getting Started with JQL

JQL Structure

JQL (Jira Query Language) is the search language available in the Advanced search mode in the Issue navigator

- Search terms are always field-operator-value/function

- Jira is not case-sensitive

- Combine terms with AND or OR
  - `project = hqhd AND (assignee = vwong OR reporter = vwong)`

- Optional sort statement at the end
  - `Project = HQHD AND (assignee = vwong OR reporter = vwong) ORDER BY created DESC`

project = GPHD and status = Open and assignee = currentUser()

field   operator value   field   operator   function
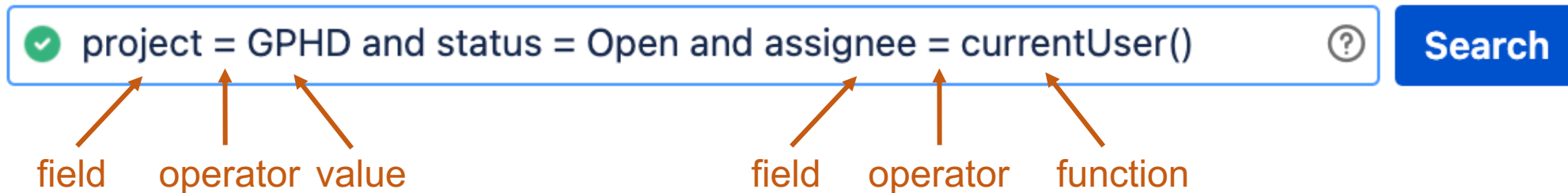
JQL Operators

operator	example	example would find
=	project = GPHD	All issues in project GPHD
!=	project != GPHD	All issues in any project except GPHD
in	project in (GPHD, HQHD)	All issues in project GPHD and all issues in project HQHD
not in	project not in (GPHD, HQND)	All issues in any project except those in GPHD or HQHD
is*	assignee is EMPTY	All unassigned issues
is not*	assignee is not EMPTY	All issues with an assignee value
>	due > endOfWeek(-110)	All issues due after the end of the week **110 weeks ago**
>=	dueDate >= endOfWeek()	All issues due at the end of this week or later
<	due < endOfWeek()	All issues due before the end of the current week
<=	dueDate <= endOfMonth()	All issues due at or before the end of the current month
~	summary ~ laptop	All issues with a word in the summary similar to "laptop"
!~	summary !~ laptop	All issues without a word in the summary similar to "laptop"

*The **is** and **is not** operators are always used with the key phrase EMPTY.

Exercise: Create a JQL Query

- Find all the Incidents with high or highest priority in the HQHD project. How many are there?

    - *Hint: try priority > Medium*

Searching Field Values

Finding issues via Text, Options, Labels, and Users

Searching on Text Fields

Text fields can be searched with the following operators

- **~** (fuzzy match, match all words similar to this)

  `summary ~ need`   *Matches words like need, needing, needed*
  `summary ~ "te?t"`   *Matches words like text and test*
  `summary ~ "t*n"`   *Matches words like turn, trainer, testing*

- **!~** (does not fuzzy match, exclude all matches similar to this word)

  `summary !~ need`   The inverse of the previous example

- **is EMPTY** (has no value)

  `description is EMPTY`

- **is not EMPTY** (has any value)

  `description is not EMPTY`

- The comments section is considered a text field for searching

  `comment ~ changes`

- You can search ALL text fields by using the word "text" instead of a field name

  `text ~ need`

- The ScriptRunner plugin enables searching for issues with <u>no comments</u>

  `issuefunction not in hasComments()`

Searching on Option Fields

Option fields can be searched with the following operators

= (equals, match one value)

```
project = HQHD AND component = Intranet
project = GPHD AND status = Open
```
All open issues in GPHD
```
"affected server" = GPHDITSM-6
```
All issues affecting the sap-erp-prod server

!= (does not equal, exclude one value)

```
status != done AND component != jira
```

in (in the set of, match multiple values)

```
project in (HQHD, GPHD) AND component in (Intranet, WiFi)
```

not in (not in the set of, exclude multiple values

```
project in (HQHD, GPHD) AND component not in (Intranet, WiFi)
```

is EMPTY (has no value)

```
component is EMPTY
resolution is NULL
```

is not EMPTY (has any value)

```
component is not EMPTY
resolution is not NULL
```

Searching on Multi-Select Option Fields

Watch out for these nuances regarding Multi-Select Option fields

"=" or "in" operators will match as long as their value/s match one of the issue's values

T	Key	Status	Created	Updated ↓	Components	Assignee	Summary
❗	HQHD-19	OPEN	21/Jan/22	07/Feb/22	Intranet, Public Website	Mitch Davis	Typo on the public website

✅ `component = Intranet`
✅ `component in (Intranet, "Public Website")`

The "!=" and "not in" match issues that have a different value, but not issues with no component

T	Key	Status	Created	Updated	Components	Assignee	Summary
🔲	HQHD-11	WAITING FOR SUPPO...	22/Jan/22	07/Feb/22	Intranet	Mitch Davis	Can't access Intranet on mobile phone

✅ `component != Jira`
✅ `component not in (Jira, WiFi)`

Find issues with component other than Jira *and include issues with no component*

`component != Jira OR component is EMPTY`

Searching on Labels Fields

- Supports **=**, **!=**, **in**, **not in**, **is EMPTY**, **is not EMPTY**
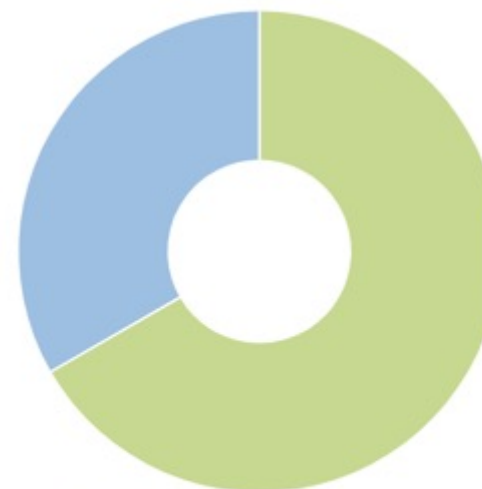- Matching in the issue navigator is NOT case sensitive, but reporting is case sensitive

When assigning labels, it is easy to make a typo or change case. Unfortunately, labels do not support wildcards or fuzzy search (~) without resorting to a custom plugin such as JQL Search Extensions for Jira.

Searching on User Fields

- Supports **=**, **!=**, **in**, **not in**, **is EMPTY**, **is not EMPTY**
- "!=" and "not in" will only match non-Empty values

T	Key	Status	Assignee	Summary		Created	Updated	Resolved
📄	HQHD-47	RESOLVED	*Unassigned*	require phone		06/Feb/22	06/Feb/22	06/Feb/22

❌ `assignee != vwong`

- *currentUser()* function can be used with "=" and "!="

    `reporter != currentUser()`   *All the issues that were reported by someone other than the current user*

    `assignee != currentUser() OR assignee is EMPTY`   *All issues not assigned to the current user*

- *membersOf()* function can be used with "in" and "not in" to search members of a specified group

    `assignee in membersOf("vip-support") OR "request participants" in membersOf("vip-support")`

    Any issue being worked on in any capacity by a member of the vip-support team

Searching Links

- Supports **=**, **!=**, **in**, **not in**
- *issueLinkType* is useful to find issues with links of a given type

```
issueLinkType = "relates to"
```
All the issues that have a "related to" link

```
issueLinkType = "is blocked by"
```
All issues that are blocked by another issue

- "!=" and "not in" **will match** Empty values

T	Key	Status	Links	Assignee	Summary
⊘	HQHD-48	**PLANNING**		Anton Dry	Upgrade Confluence to new version

✅ `issueLinkType != "relates to"`
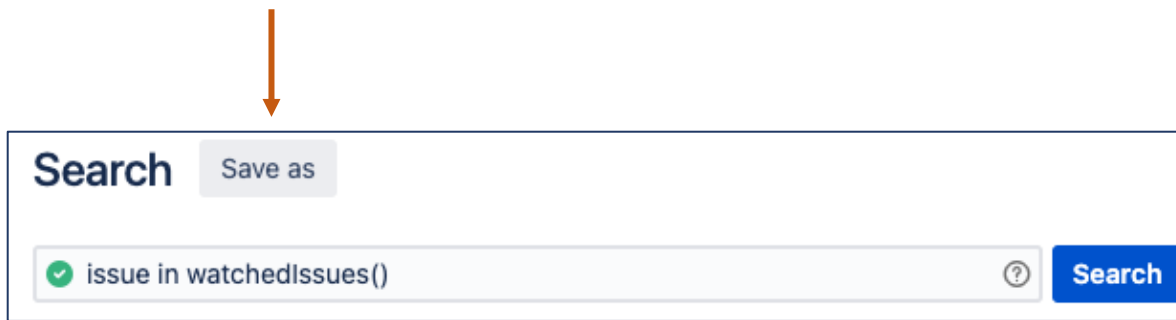
- The ScriptRunner plugin enables searching for issues with <u>any links</u>

```
issuefunction in hasLinks()
```

Visualizing Results

Filters
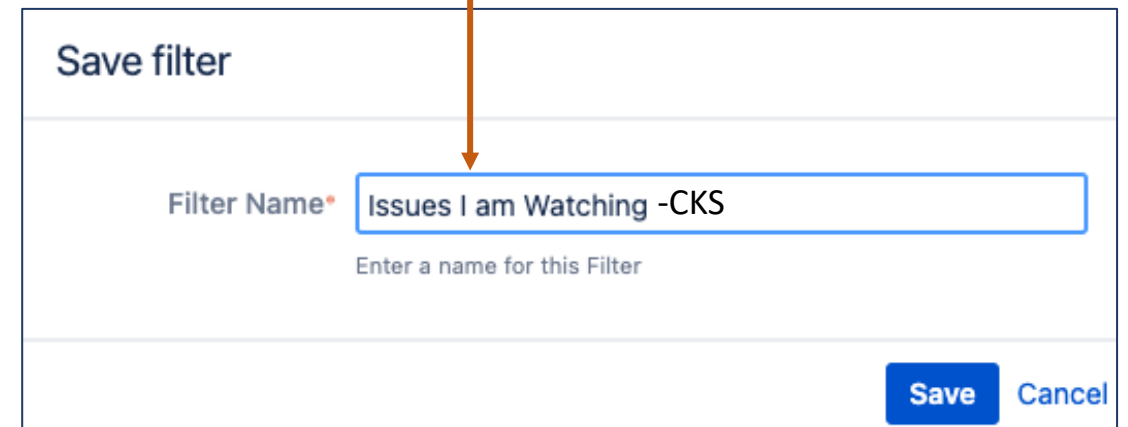
- If you have a search that you will need to use frequently, save it!
- In Jira, a saved search is called a **filter**.
- Filters save both their query and the column configuration.
- Filters can be shared with other users.
- You can subscribe to a filter to have its results emailed to you periodically.
- Filters are critical for building dashboards.

Save the current search as a filter

Give the filter a descriptive name

Search Save as

issue in watchedIssues()   ⑦ **Search**

Save filter

Filter Name*   Issues I am Watching -CKS

Enter a name for this Filter

Save Cancel

Exercise: Save a Filter

Make sure to add your initials to the name, otherwise you will overwrite another student's filter

- Navigate to the Issue Navigator
- Type in the following JQL

```
issue in watchedIssues()
```

- Execute the search

- Click "Save As" and name the filter "Issues I am Watching-CKS"
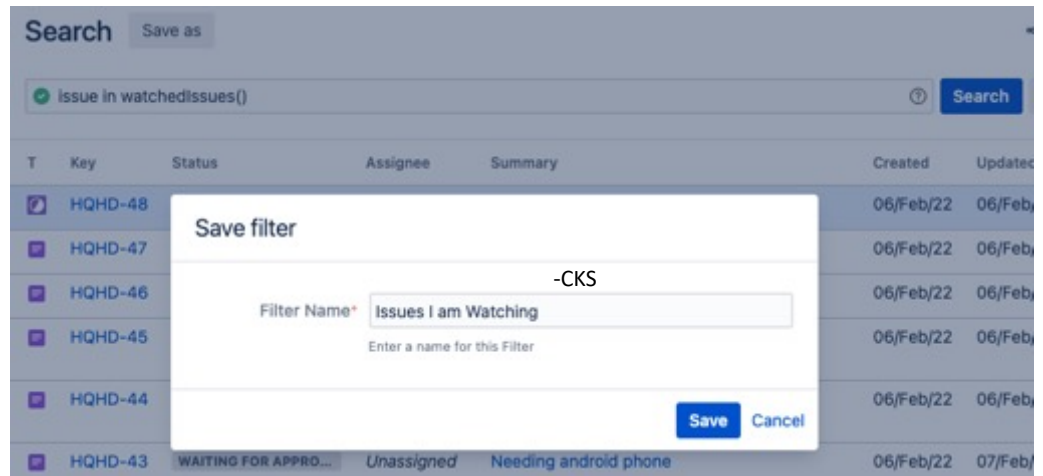
- Where does the Filter go?

- Try clicking on some of the other Filters

View Result In Dashboard

1. Select "Workshop Dashboard" from the "Dashboards" menu
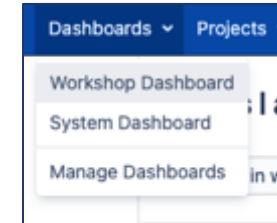
2. Copy the dashboard. Name it with your own name.
3. See the gadget at the top. It has a blue bar labeled "Filter Results: Closed Requests"
4. Click on the ". . ." menu at right hand side of the blue bar and select "Edit"

5. Change the Saved Filter to "Issues I am Watching-CKS"

6. Observe the updated list

Searching History, Dates, and SLAs

Searching on Date Fields

Searching with **fixed** dates

- Date format: yyyy/mm/dd
- Date-time format: yyyy/mm/dd hh:mm

> ✅ created >= "2020/06/19"   ⑦

Created on or after June 19, 2020

Searching with **relative** dates

- "w", "d", "h", or "m"
  (weeks, days, hours, or minutes)

> ✅ created > -5d   ⑦

Created after 5 days ago

Searching with date **functions** (also relative)

- now()
- startOfDay(), startOfWeek(), startOfMonth(), startOfYear()
- endOfDay(), endOfWeek(), endOfMonth(), endOfYear()

Other examples: endOfYear(-1y), endOfMonth(-1M)

> ✅ due < now()   ⑦

Anything now overdue

> ✅ resolved >= startOfWeek()   ⑦

Resolved since the start of this week

> ✅ created < startOfMonth(-1)   ⑦

Created before the start of last month

Special Fields that Allow Searching Their History

Fields: Assignee, Reporter, Resolution, Status

Was - value was ever this, including currently

```
assignee was jira                          Issues currently or previously assigned to jira
status was "Waiting for Support"     Issues currently or previously waiting for support
status was "Waiting for support" and status != "waiting for support"
```

Was Not - value was never this at any point

```
status was not "Waiting for approval"
```

Was In – value was in this set of values, including currently

```
Assignee was in (jira, fred.astair)
```

Was Not In – value was never one of these values

```
resolution was not in ("Won't Do", Declined, Duplicate)
```

Changed – this value changed at some point <u>after creation</u>

```
reporter changed
status changed
```

Searching History Fields Part 2

History syntax adds a second set of operators that are not normally available

AFTER – after a given date-time
```
status was "In Progress" AFTER startOfMonth()
```

BEFORE – before a given date-time
```
resolution was not Done BEFORE "2021-06-19"
```

DURING – during a given date-time range
```
status was "Escalated" DURING (startOfWeek(-1), endOfWeek(-1))
```

ON – at a specific moment in time
```
assignee was jira ON -1d
assignee was in (jira, fred.astair) ON "2022-02-06"
```

BY – which user set the field value
```
reporter changed BY currentUser()
```

TO – what the value was set to
```
status changed TO Closed
```

FROM – what the value was changed from
```
resolution changed FROM "Won't Do"
```

These secondary operators can be strung together
```
status changed FROM "waiting for approval" TO "waiting for support" BY currentUser()
AFTER startOfWeek(-3)
```

JQL Functions: Approvals

function(s)	example	example would find
approved()	approvals = approved()	All issues requiring approval that were approved
pending()	approvals = pending()	All issues requiring approval
myPending()	approvals = myPending()	All issues requiring my approval
pendingBy(*user*)	approvals = pendingBy(fred.astair)	All issues requiring approval by Fred Astair
myApproval()	approvals = myApproval()	All issues that require or required my approval
approver(*user*)	approvals = approver(fred.astair)	All issues that require or required approval by Fred Astair
declined()	approvals = declined()	All issues that were declined

Jira Service Management makes approvals easy to query via the *'approvals'* field

HQHD-45

⌄ Past approvals

WAITING FOR APPRO... Approved 2 hours ago

Fred Astair ✓

jira

JQL Functions: Service Level Agreements

function(s)	example	example would find
breached()	"Time to First Response" = breached()	All issues that missed their SLA goal
paused()	"Time to resolution" = paused()	All issues having an SLA that is paused due to some condition
running()	"Time to first response" = running()	All issues having an SLA that is running
elapsed()	"Time to first response" > elapsed("4h")	All issues that have been waiting for a first response for more than four hours
remaining()	"Time to close after resolution" < remaining("4h")	All issues that have less than 4 hours remaining to meet the SLA for closing
completed()	"time to approve normal change" = completed()	All issues having an SLA that is completed

Jira Service Management allows you to configure Service Level Agreements for your project.   Each SLA is treated as a field in JQL that you can query

JQL Functions: Service Level Agreements

??

But wait...how do you know the names of the SLAs?

Reports → SLA Goals

Headquarters Help Desk

- Queues
- Customers
- Reports
- Objects
- Raise a request
- Knowledge base
- Customer channels

TEAM
Workload
SLA goals
Satisfaction

KNOWLEDGE BASE
Article usage
Article effectiveness

CUSTOM
Created vs Resolved

SLA goals

Time to approve normal change

Issues	Goal	Calendar	Success of all ongoing SLAs and last 7 days
issuetype = Change AND "Change type" = Normal	40h	Sample 9-5 Calendar	No issues

0% Time to close after resolution

Issues	Goal	Calendar	Success of all ongoing SLAs and last 7 days
issuetype in (Incident, "Service Request", "Service Request with Approvals")	24h	Sample 9-5 Calendar	0%

7% Time to first response

Issues	Goal	Calendar	Success of all ongoing SLAs and last 7 days
issuetype = Incident	2h	Sample 9-5 Calendar	0%
issuetype in ("Service Request", "Service Request with Approvals")	4h	Sample 9-5 Calendar	12%

7% Time to resolution

Issues	Goal	Calendar	Success of all ongoing SLAs and last 7 days
issuetype = Incident	4h	Sample 9-5 Calendar	0%
issuetype in ("Service Request", "Service Request with Approvals")	8h	Sample 9-5 Calendar	11%

1. Time to first response
2. Time to approve normal change
3. Time to resolution
4. Time to close after resolution

By convention, all SLAs should start with the word "Time"

Putting it All Together

Confidential © Copyright Ariel Partners 2022

Cool JQL Queries

1	`(assignee = currentUser() OR "request participants" = currentUser()) and resolution is EMPTY`
	All issues where you are the assignee or a request participant that are not completed
2	`project = HQHD AND issuetype = Incident AND created >= startOfMonth(-2) and created < startOfMonth()`
	All Incidents created last 2 months
3	`project = HQHD AND component = Intranet AND resolution is EMPTY`
	All of the Intranet-related issues that are currently in process
4	`project = HQHD AND resolved > -30d`
	Everything we finished in the last 30 days
5	`project = HQHD AND type = "Service Request" AND status = "In Progress"`
	All the Service Requests that are currently in progress
6	`project = HQHD AND reporter != agrant`
	Everything that was not reported by Alana Grant

Homework Exercise: Putting it All Together

See if you can create JQL queries for these:

1. All issues that are "waiting for support" but not yet assigned to anyone

2. All HQHD change requests that are not resolved but do not have any request participants assigned

3. All issues that are unassigned, but which met their "Time to first Response" SLA

4. All issues that exceeded their "*Time to first Response*" SLA last month   (*hint: use* `createdDate`)

    [and use the period between 25-26 months ago]

Wrapping Up

In Summary: Where Can JQL Be Used?

- Issue Navigator, Saved Filters

- Dashboards

- JSM Issue Queues

- Many Plugins
  - Automation For Jira (conditionally perform a rule on matching issues)
  - Structure (populate or filter a structure based on a JQL query)
  - JSU Automation Suite for Workflows (workflow conditions can be based on JQL queries)
  - ScriptRunner

Key Takeaways

- Advanced JQL Queries are extremely powerful
- The combination of power and ease of use of JQL is a key market differentiator for Jira
- Advanced JQL underlies nearly every other feature of Jira
- JQL operators are combined with Booleans (AND, OR, NOT)
- Comparison operators can be used for numeric fields (=, !=, <, <=, >, >=)
- Many JQL operators exist to query fields (IS EMPTY, IS NOT EMPTY, IN, NOT IN)
- Many JQL operators exist to query history (WAS, WAS NOT, WAS IN, WAS NOT IN, CHANGED)
- History queries can be extended (AFTER, BEFORE, DURING, ON, BY, TO, FROM)
- Many convenient functions exist (approved(), breached(), currentUser(), membersOf())
- ScriptRunner adds even more functions (hasLinks(), hasComments())

Ariel | Atlassian Services

Training
Ariel offers training on all Atlassian products both on-site and virtually, for private groups and the general public.

Coaching
Ariel provides Agile coaching for teams using popular Agile methods such as Scrum, Kanban, SAFe, and Flight Levels helping them to utilize Jira to help drive continuous improvements

Licensing
As an Atlassian Verified Government Partner we can fulfill all customers' licensing needs while aligning divergent license periods

JumpStart
Ariel offers a bundled set of services including installation, migration, integration, configuration, and training to get organizations up and running quickly

Silver Solution Partner
US GOVERNMENT

ATLASSIAN
team**TOUR**
Government

Sponsored by:
carahsoft

**Ariel Partners
Atlassian Solutions**

Contact Us!

Craeg Strong, *CTO Ariel Partners*

cstrong@arielpartners.com

https://www.linkedin.com/in/cstrong

https://www.arielpartners.com/atlassiansolutions

https://arielpartners.com/upcoming-courses/

ARIEL
P A R T N E R S
MISSION CRITICAL AGILE™