

# Tracking issues with JIRA?

## Let's visualize epics' status and progress

**Sebastian Łopieński**

CERN IT Lightning Talks #25

<https://indico.cern.ch/e/ITLT-25>

8 November 2024

# JIRA is great for issue tracking, work planning, sprints etc.

The image displays a JIRA dashboard with four columns of issue cards. Each card contains the following information:

- Column 1 (TO DO):**
  - MAJ-TEAM-4220: Remove hamburger menu icon. Status: GAMS-5 Q3 25 - Missing features.
  - MAJ-TEAM-4172: Break up Group Creation Page JS. Status: GAMS-5 Q3 25 - Missing features.
  - MAJ-TEAM-3765: Correct redirection URL's to application portal from groups portal. Status: GAMS - Misc.
  - MAJ-TEAM-4191: Deny filtering by non public fields. Status: Stability and Maintainability.
  - MAJ-TEAM-4088: Cold recovery IBM/MSM. Status: Disaster Recovery.
- Column 2 (IN PROGRESS):**
  - MAJ-TEAM-4220: GAMS sync: add endpoint to synchronize single group. Status: GAMS-2 Q2 24 - Mail & Sync.
  - MAJ-TEAM-4234: Work sessions - Christopher Silva.
  - MAJ-TEAM-3295: Export report files to TDR/OTR/NDBox. Status: Consolidate Integration tests.
  - MAJ-TEAM-3808: Prepare a static info page. Status: SSO info page.
  - MAJ-TEAM-3528: Warn SAML application owners about their expiring certificates. Status: Clean-up application registrations.
- Column 3 (IN REVIEW):**
  - MAJ-TEAM-4092: Plan, Test and Deploy HSAs in Certification Authority. Status: ✓ =.
  - MAJ-TEAM-3808: Policy: restrict synchronization type if synchronization broken. Status: GAMS-2 Q2 24 - Mail & Sync.
  - MAJ-TEAM-3254: Identify existing script providers and deploy them correctly if still used. Status: Keycloak Config Clean-up.
  - MAJ-TEAM-3836: Categorize, identify the source and propose actions for identify membership difference between. Status: GAMS-2 Q2 24 - Mail & Sync.
  - MAJ-TEAM-3806: [Title partially obscured]
- Column 4 (DONE):**
  - MAJ-TEAM-4235: Group sync: put actions behind authentication. Status: GAMS-2 Q2 24 - Mail & Sync.
  - MAJ-TEAM-4252: Work sessions - Christopher Silva.
  - MAJ-TEAM-3834: Categorize, identify the source. Status: R.
  - MAJ-TEAM-3844: Work sessions - Mail Silva.
  - MAJ-TEAM-3884: Add "Team" to LifecycleSettings. Status: Eligibility & Lifecycle.
  - MAJ-TEAM-3818: State transition job. Status: M.
  - MAJ-TEAM-3818: State transition job. Status: Eligibility & Lifecycle - Openback.

# JIRA reporting is also **rich** and **powerful**

## Agile



**Burndown Chart**

Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.



**Burnup Chart**

Track the total scope independently from the total work done. This helps your team manage its progress and better understand the effect of scope change.



**Sprint Report**

Understand the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is excessive scope creep.



**Velocity Chart**

Track the amount of work completed from sprint to sprint to determine your team's velocity and estimate the work your team can realistically achieve in future sprints.



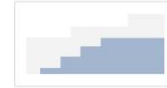
**Cumulative Flow Diagram**

Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.



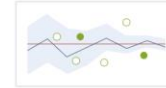
**Version Report**

Track the projected release date for a version. This helps you monitor whether the version will release on time, so you can take action if work is falling behind.



**Epic Report**

Understand the progress towards completing an epic over time. This helps you manage your team's progress by tracking the remaining incompletely/unestimated work.



**Control Chart**

Shows the cycle time for your product, version or sprint. This helps you identify whether data from the current process can be used to determine future performance.



**Epic Burndown**

Track the projected number of sprints required to complete the epic (optimized for Scrum). This helps you monitor whether the epic will release on time, so you can take action if work is falling behind.



**Release Burndown**

Track the projected release date for a version (optimized for Scrum). This helps you monitor whether the version will release on time, so you can take action if work is falling behind.

## Issue analysis



**Average Age Report**

Shows the average age of unresolved issues for a project or filter. This helps you see whether your backlog is being kept up to date.



**Created vs. Resolved Issues Report**

Maps created issues versus resolved issues over a period of time. This can help you understand whether your overall backlog is growing or shrinking.



**Pie Chart Report**

Shows a pie chart of issues for a project/filter grouped by a specified field. This helps you see the breakdown of a set of issues, at a glance.



**Recently Created Issues Report**

Shows the number of issues created over a period of time for a project/filter, and how many were resolved. This helps you understand if your team is keeping up with incoming work.



**Resolution Time Report**

Shows the length of time taken to resolve a set of issues for a project/filter. This helps you identify trends and incidents that you can investigate further.



**Single Level Group By Report**

Shows issues grouped by a particular field for a filter. This helps you group search results by a field and see the overall status of each group.



**Time Since Issues Report**

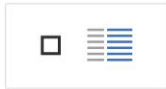
For a date field and project/filter, maps the issues against the date that the field was set. This can help you track how many issues were created, updated, etc. over a period of time.

## Forecast & management



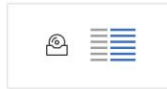
**Time Tracking Report**

Shows the original and current time estimates for issues in the current project. This can help you determine whether work is on track for those issues. You can either analyze the report in your browser or use a CSV file.



**User Workload Report**

Shows the time estimates for all unresolved issues assigned to a user across projects. This helps you understand the user's workload better.



**Version Workload Report**

Shows the time estimates for all unresolved issues assigned to a version, broken down by user and issues. This helps you understand the remaining work for the version.

## Other

Date: 26/Nov/12	Alice	Jimmy	Total
	2h	1h	3h
Another Bug	2h	1h	3h
A Problem	2h	1h	3h

**Project Pivot Report**

Report displaying a time spent by users over a specified period.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Yellow							
Green	1h						
Blue							
Orange							

**Time Sheet Report**

Report displaying users worked time sheet over specified period.

	Timespent
TIME-3	Hindenbug 1h
TIME-4	Mega problem 8h
TP-2	Loch Ness Monster Bug 7h
	2h

**Timespent Report**

Report displaying a time spent during specified period.



**Traceability Matrix Report**

It shows a matrix of a 10,000 (maximum) cells. The columns and rows are limited to 1000. The columns are got first and the rows later to fit ~10,000 cells. Ex (100x100), (250x40), (1000x10), (1x1000)etc. Please, select the filters for the rows and columns.



**Workload Pie Chart Report**

A report showing the issues for a project or filter as a pie chart.

... however, I want something **simpler** and **more flexible!**

# First, get the data from JIRA

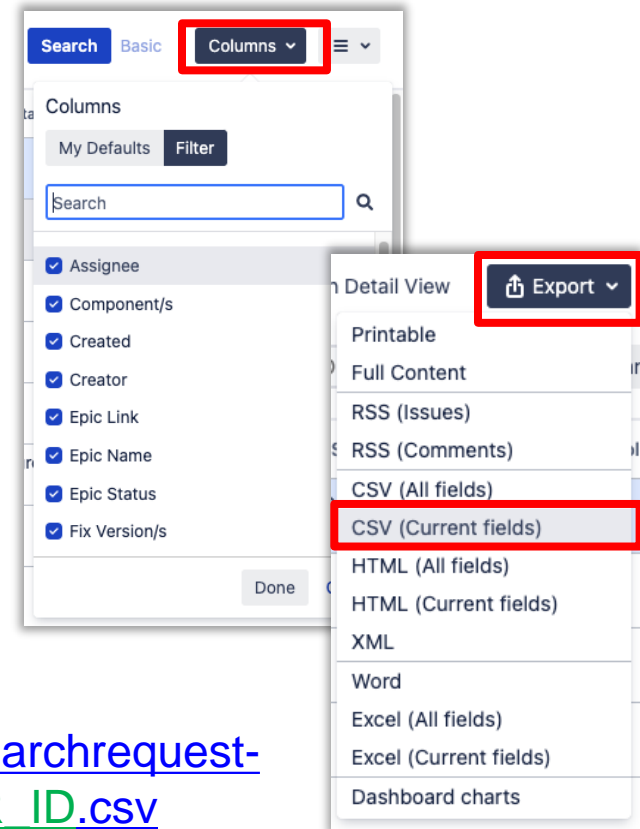
1. [Search](#) for JIRA issues (including epics)
  - can be generic (include more issues than needed)
  - optionally, save that filter for future reuse

2. Select (at least) these columns:

*Created, Epic Link, Epic Name, Epic Status, Issue Type, Key, Resolved, Status, Summary, Updated*

3. Export as “CSV (*current fields*)” to tickets.csv

Direct URL: [https://its.cern.ch/jira/sr/jira.issueviews:searchrequest-csv-current-fields/FILTER\\_ID/SearchRequest-FILTER\\_ID.csv](https://its.cern.ch/jira/sr/jira.issueviews:searchrequest-csv-current-fields/FILTER_ID/SearchRequest-FILTER_ID.csv)



# DEMO



<https://gitlab.cern.ch/slopiens/jira-progress-visualisation>

# Let's start with a single JIRA Epic

Input:

```
test1.txt
1 https://its.cern.ch/jira/browse/MALTIAM-3345
```

Outputs:

```
$ ./show-progress.py test1.txt
https://its.cern.ch/jira/browse/MALTIAM-3345: ✓ ✓ ✓ ✓ ✗ ✓ — □ ✓
$ ./show-progress.py test1.txt txt
Guest accounts improvements : ✓ ✓ ✓ ✓ ✗ ✓ — □ ✓
$ ./show-progress.py test1.txt md
[Guest accounts improvements](https://its.cern.ch/jira/browse/MALTIAM-3345): ✓ ✓ ✓ ✓ ✗ ✓ — □ ✓
```

# Displaying multiple Epics

Input:

```
test2.md > abc # My project
1 # My project
2 (Markdown headers are used)
3
4 Any format is OK:
5
6 https://its.cern.ch/jira/browse/MALTIAM-2176
7 MALTIAM-3210
8 [Guest accounts improvements](https://its.cern.ch/jira/browse/MALTIAM-3345)
9 * Keycloak theme improvements: MALTIAM-3344
10
11 Lnes with no JIRA issue key (XXXXXXX-dddd) - matching \b[A-Z\d]{2,20}-\d{1,5}\b
12 are ignored...
```

# Displaying multiple Epics

```
$ ./show-progress.py test2.md txt
```

```
# My project
```

```
Application registration
```

```
Application management
```

```
Guest accounts improvements
```

```
Keycloak theme improvements
```

```
: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓  
: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓  
: ✓ ✓ ✓ ✓ ✗ ✓ - ✓ ✓ ✓ ✓  
: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ - - ✓ ✓ ✓ ✓ ✓ ✓
```



## Bonus: animate progress over time!

```
$ ./show-progress.py test2.md animate
```

```
# My project
```

```
Add internationalization support :
```

```
Update backend stack :
```

```
General usability improvements :
```

```
Frontend fixes : 
```

# Thank you for your attention

Feel free to use, and adapt this code:



<https://gitlab.cern.ch/slopiens/jira-progress-visualisation>