Búsqueda avanzada en Jira

Además de la **búsqueda rápida**, accesible en cualquier pantalla en la zona superior a la derecha, o la **búsqueda básica**, accesible pulsando en la barra superior Tareas-> Buscar incidencias, en Jira podemos ejecutar la **búsqueda avanzada**.

¿Qué es una búsqueda avanzada?

La búsqueda avanzada permite utilizar consultas estructuradas para buscar asuntos de JIRA. Los resultados de búsqueda se muestran en el "Navegador de Asuntos", donde el listado se puede exportar a múltiples formatos accediendo al submenú "Exportar" arriba a la derecha. También es posible guardar esta consulta como un **filtro** y suscribirse a esta búsqueda avanzada.

Cuando se realiza una búsqueda avanzada, se está usando el lenguaje de consultas de JIRA (JQL o "JIRA Query Language").

Una consulta simple en JQL (también conocida como una "cláusula") consiste en un **campo**, seguido por un **operador**, seguido de uno o más **valores o funciones**. Por ejemplo, la siguiente consulta sencilla encontrará todos los temas en el proyecto de "TEST":

```
project = "TEST"
```

(Este ejemplo usa el campo Proyecto ("project"), el operado IGUAL y el valor TEST)

> ⚠ Se debe tener en cuenta que **no** es posible comparar dos campos.

JQL proporciona una sintaxis parecida a SQL, tal como el ORDER BY (para la ordenación) y la función de SQL "ISNULL()". De todas formas, JQL no es un lenguaje de consultas a bases de datos. Por ejemplo, JQL no dispone de una sentencia SELECT.

> ⓘ **Documentación oficial**
>
> La documentación más actualizada en relación a búsqueda avanzada se puede encontrar en: https://confluence.atlassian.com/jiracorecloud/advanced-searching-operators-reference-765593718.html

Cómo realizar una búsqueda avanzada

1. Desde el interfaz de Jira, navegar en "Tareas" -> "Buscar incidencias". Se mostrará el **Navegador de asuntos**.
    - Si se muestra el enlace de "Búsqueda avanzada", pulsar para cambiar.
2. Teclear la consulta JQL usando los campos, operadores y valores de campo o funciones.
3. Pulsar sobre el botón de búsqueda para ejecutar la consulta.
4. Una vez ejecutada la consulta, Jira nos permite guardar la consulta como un **filtro**. Utilizaremos el botón "Guardar como".

Uso de la función de autocompletado

A la vez que se escribe la consulta, Jira reconocerá el contexto y ofrecerá una lista de sugerencias de "autocompletado" de la siguiente forma:

La lista de sugerencias de **autocompletado** se muestra por orden alfabético e incluye las 15 primeras coincidencias. Anotar aquí que las sugerencias de autocompletado no se ofrecen para las funciones.

Cambio entre la búsqueda "avanzada" y la "simple"

En general, una consulta creada usando una búsqueda simple estará disponible para ser traducida a una "Búsqueda avanzada" (o JQL), y al revés.

De todas formas, una consulta creada usando la "Búsqueda avanzada" podría no estar disponible para su traducción a una búsqueda, particularmente si:

- la consulta contiene un operador OR (anotar que se puede tener un operador IN y éste será traducido, por ejemplo project in (A, B))
    - p.e. incluso cuando se tiene una consulta como: `(project = JRA OR project = CONF)` que es equivalente a esta query:`(project in (JRA, CONF))`, únicamente la secunda consulta será traducida.
- la consulta contiene un operador NOT
- la consulta contiene un operador EMPTY
- la consulta contiene cualquiera de los operadores de comparación: !=, IS, IS NOT, >, >=, <, <=
- la consulta especifica un campo y valor que está relacionado con un proyecto (p.e. version, component, custom fields) y el proyecto no está explicitamente incluido en la consulta (p.e.

`fixVersion = "4.0"`, sin el `AND project=JRA`). Esto es especialmente difícil con los campos personalizados, ya que se pueden configurar e n base a un proyecto / asunto tipo. La regla general es que si la consulta no se puede crear en el formulario 'Búsqueda Simple', entonces si se crea utilizando "Búsqueda avanzada" no podrá ser traducido a "Búsqueda simple".

Configuración de la precedencia de los operadores

Se puede utilizar paréntesis en las declaraciones JQL complejos para hacer cumplir la precedencia de operadores.

Por ejemplo, si desea buscar todos los problemas resueltos en el proyecto SysAdmin, así como todos los asuntos (cualquier estado, cualquier proyecto) a signadas al administrador del sistema (hmora), puede utilizar paréntesis para hacer cumplir la precedencia de los operadores lógicos en su consulta, p.e.:

```
(status=resolved AND project=SysAdmin) OR assignee=aperez
```

Anotar que si no se usan paréntesis, la declaración será evaluada de izquierda a derecha.

También se pueden usar paréntesis para agrupar consultas, tal que puedes aplicar el operador NOT al grupo

Palabras clave

Una palabra clave en JQL es una palabra o frase que hace (o es) cualquiera de estas:

- une dos o más cláusulas juntas para formar una consulta JQL compleja
- altera la lógica de una o más cláusulas
- altera la lógica de los operadores
- tiene una definición explícita en una consulta JQL
- realiza una función específica que altera los resultados de una consulta JQL.

Listado de palabras clave

- AND
- OR
- NOT
- EMPTY
- NULL
- ORDER BY

Operadores

Un operador en JQL es uno o más símbolos o palabras que comparan el valor de un campo en su lado izquierdo con uno o más valores (o funciones) en s u lado derecho, de tal manera que sólo los resultados verdaderos son recuperados por la cláusula. Algunos operadores pueden utilizar la palabra clave NOT.

Listado de Operadores

- EQUALS: =
- NOT EQUALS: !=
- GREATER THAN: >
- GREATER THAN EQUALS: >=
- LESS THAN: <
- LESS THAN EQUALS: <=
- IN
- NOT IN
- CONTAINS: ~
- DOES NOT CONTAIN: !~
- IS
- IS NOT
- WAS
- WAS IN
- WAS NOT IN
- WAS NOT
- CHANGED

Refencia de Campos

Un campo en JQL es un palabra que representa un campo JIRA (o un campo personalizado que se haya definido). En una cláusula, un campo es seguido por un operador, que a su vez es seguido por uno o más valores (o funciones).

El operador compara el valor del campo con uno o más valores o funciones de la parte derecha, tal que únicamente resultados ciertos o "verdaderos" son devueltos por la cláusula.

Listado de campos

Campo	Traducción	Descripción
assignee	Asignado	Usuario de JIRA que tiene asignado el asunto
category	Categoría	Categoría del proyecto
comment	Comentario	
component	Componente	
created	Fecha de Creación	
description	Descripción	Campo con la descripción
due	Fecha de Entrega	
level	Nivel de Seguridad	
parent	Asunto Padre	
priority	Prioridad	
project	Proyecto	Clave o nombre del proyecto
reporter	Informador	Usuario de JIRA que abre la incidencia
resolution	Resolución	Código de resolución
status	Estado	Estado en el que se encuentra el asunto dentro del flujo de trabajo
summary	Resumen	Resumen del asunto
updated	Fecha de Actualización	Última fecha en la que el asunto fue modificado