JQL CHEAT SHEET ATLASSIAN

JQL Cheat Sheet Atlassian is an essential resource for anyone looking to maximize their use of Jira, Atlassian's powerful project management and issue-tracking tool. Jira Query Language (JQL) allows users to create complex queries to filter and manipulate data within their Jira projects. This cheat sheet serves as a comprehensive guide to understanding and utilizing JQL effectively, providing you with the tools to extract valuable insights from your Jira data.

What is JQL?

Jira Query Language (JQL) is a powerful querying language that enables users to perform searches and filter issues in Jira. Unlike the basic search functionality, which limits users to simple queries, JQL allows for more complex searches with multiple criteria, helping users find exactly what they need quickly and efficiently.

Key Features of JQL

- Flexibility: JQL provides a flexible syntax that allows users to combine multiple criteria in a single query.
- Custom Fields: Users can search not just standard fields (like status or priority) but also custom fields defined within their Jira instance.
- Sorting and Ordering: You can sort results based on various fields to prioritize the information that is most relevant to your needs.
- Date Functions: JQL includes powerful date functions that allow users to filter issues based on specific timeframes.

Basic JQL Structure

JQL queries consist of three main components: fields, operators, and values.

- Fields: These are the attributes of the issue you want to search. Examples include `status`, `assignee`, `priority`, and custom fields.
- Operators: Operators define the relationship between the field and the value. Common operators include `=`, `!=`, `>`, `<`, `IN`, and `IS`.
- Values: These are the actual data points you are searching for, such as a specific user, status, or date.

Basic Query Examples

1. Find all issues assigned to a specific user:

```
assignee = "username"
```

2. Find all issues that are open:

```
status = "Open"
```

3. Find issues with a specific priority:

```
priority = "High"
```

~~~
```
~~~

4. Find issues created in the last 7 days:

```
created >= -7d
```

JQL Operators

JQL supports a variety of operators that allow for nuanced searching. Here are some of the most commonly used operators:

Comparison Operators

- `=`: Equals
- `!=`: Not equal
- `>`: Greater than
- `<`: Less than
- `>=`: Greater than or equal to
- `<=`: Less than or equal to

Logical Operators

- `AND`: Combines two conditions; both must be true.
- `OR`: Combines two conditions; at least one must be true.
- `NOT`: Negates a condition.

Inclusion Operators

- `IN`: Matches any value in a specified list.
- `IS`: Checks for the presence of a value (e.g., checking if a field is empty).

Advanced JQL Queries

Once you're comfortable with the basics, you can start creating more complex queries. Here are some advanced query techniques:

Combining Conditions

You can combine multiple conditions using logical operators to refine your search:

```
project = "MyProject" AND status = "Open" AND priority IN ("High", "Medium")
```

This query finds all open issues in "MyProject" that have a priority of either High or Medium.

Using Functions

JQL includes several built-in functions that can enhance your queries:

- currentUser(): Returns the currently logged-in user.
- now(): Represents the current date and time.
- startOfDay(), startOfMonth(), startOfYear(): These functions allow you to work with relative dates.

Example using a function:

```
assignee = currentUser() AND created >= startOfDay(-1)
```

This query fetches issues assigned to the current user that were created in the last 24 hours.

Sorting and Ordering Results

You can sort your results by adding an `ORDER BY` clause at the end of your query:

```
project = "MyProject" ORDER BY priority DESC, created ASC
```

This query lists all issues in "MyProject," sorting them first by priority (highest to lowest) and then by creation date (oldest to newest).

Common Use Cases for JQL

JQL can be used for a variety of purposes in Jira. Here are some common scenarios:

Tracking Progress

To monitor the status of tasks in a project, you might use:

```
project = "ProjectName" AND status != "Done"
```

This query shows all tasks that are not yet completed.

Identifying Bottlenecks

To find issues that have been in the same status for too long, you can use:

```
status = "In Progress" AND updated <= -7d
```

This query returns issues that have been in progress for more than a week without updates.

Assigning Tasks

To view tasks assigned to a specific team member:

```
assignee = "username" AND status != "Closed"
```

This query will show all open tasks assigned to a particular user.

Tips for Writing JQL Queries

- Use Autocomplete: Jira provides autocomplete suggestions while typing your query. This can help avoid syntax errors.
- Test Queries: Before finalizing your queries, test them to ensure they return the expected results.
- Save Filters: If you frequently run the same queries, save them as filters for quick access in the future.
- Use Parentheses: When combining multiple conditions, use parentheses to control the order of operations.
- Refer to Documentation: Atlassian's official documentation provides additional examples and explanations of JQL functions and capabilities.

Conclusion

The JQL Cheat Sheet Atlassian is an invaluable tool for anyone using Jira. By mastering JQL, you can unlock the full potential of Jira's search capabilities, allowing you to filter and manipulate data to meet your project management needs. Whether you are a project manager, team leader, or developer, understanding JQL can greatly enhance your productivity and efficiency in managing tasks and issues. With practice and experimentation, you can create complex queries that provide insights into your projects, helping you make informed decisions and drive your team's success.

Frequently Asked Questions

What is JQL in Atlassian and why is it important?

JQL, or Jira Query Language, is a powerful query language used in Atlassian Jira to create complex queries for filtering issues. It is important because it allows users to retrieve specific sets of issues based on various criteria, enhancing project management and tracking.

What are some common JQL functions I should know?

Common JQL functions include 'currentUser()', 'now()', 'startOfDay()', 'endOfWeek()', and 'membersOf()'. These functions help filter issues based on the current user, dates, and group memberships.

How can I filter issues by status using JQL?

To filter issues by status in JQL, you can use the syntax: 'status = "Status Name"'. For example, to find all issues that are 'In Progress', you would write: 'status = "In Progress"'.

Can I combine multiple JQL statements? If so, how?

Yes, you can combine multiple JQL statements using logical operators like AND, OR, and NOT. For instance, to find issues that are 'Open' or 'In Progress', you would write: 'status = Open OR status = "In Progress"'.

Is there a way to save my JQL filters for future use?

Yes, in Jira, after you run a JQL query, you can save it as a filter by clicking on 'Save as' at the top of the issue navigator. You can name your filter and access it later from the 'Filters' menu.

Jql Cheat Sheet Atlassian

Jql Cheat Sheet Atlassian