# structure() JQL function

Structure adds `structure()` JQL function that lets you search for issues that are added to a structure, with the possibility to add constraints on their relationships.You can use this function in any place in JIRA where you can use JQL: in the Issue Navigator, in a Saved Filter, as an Agile Board query etc. For more information, see JIRA documentation on Advanced Searching and Advanced Searching Functions.

If a user does not have access to structure, they will not be able to create new queries with the `structure()` function and existing queries will have `structure()` function return an empty set. However, the user will still see `structure()` function offered in the JQL completion drop-down.

To specify a structure condition in JQL, use the following format:

**issue in structure(***structureName*$_{opt}$, *structureQuery*$_{opt}$**)**

Function arguments:

| | | |
|---|---|---|
| **structureName** | *Optional* | The name of the structure. If you omit the structure name, system-wide Default Structure will be searched. |
| **structureQuery** | *Optional* | Use this parameter to select only a part of the structure. This parameter specifies a *Structure Query* in a language similar to JQL, Structured JQL. |

You can use structure ID instead of the structure name. You can see structure ID in the URL of the Structure Board if you open **Manage Structure** page and click structure name.

## Function arguments need to be quoted if they contain spaces or non-letters

As dictated by the syntax of JQL, you'll need to enclose structure name or structure query in 'single quotes' or "double quotes" if they contain spaces or non-letters.

### What if *structure name* or *structure query* itself contains quotes?

If structure name or structure query contains quotes of one kind, you need to enclose them with a different kind of quotes. That is, if structure query contains double quote, you'll need to enclose it in single quotes. Alternatively, you can escape quote with a backslash: `\"`.

#### Example 1

Suppose you need to find all issues that are directly under issues in status *Awaiting Deployment.*

In plain JQL, issues in this status can be found via this query: `Status = "Awaiting Deployment"`. Note that since status name contains spaces, JQL requires us to enclose it in quotes.

According to S-JQL Reference, the corresponding Structure query would be `child of [Status = "Awaiting Deployment"]`.

That means that you need to enclose this Structure query with single quotes:

```
issue in structure("My personal structure", 'child of [Status = "Awaiting Deployment"]')
```

Note that the following will **not** work:

```
issue in structure("My personal structure", "child of [Status = "Awaiting Deployment"]")
```

#### Example 2: escaping with backslash

In the following example, the query returns issues that are directly under issues assigned to fix version named *3.0 "Armageddon"*.

```
issue in structure("My personal structure", "child of [fixVersion = '3.0 \"Armageddon\"']")
```

## Backward compatibility with **`structure()`** JQL function prior to Structure 2.4

Prior to Structure 2.4, `structure()` JQL function did not take structure query as an argument; you could specify only one issue key or ID, and you would get the referenced issue along with all of its children at all levels. As you might have noticed, this old-style usage can be interpreted as a structure query, but according to the rules of S-JQL, it would return just the referenced issue without its children. To maintain backward compatibility, any structure query in Structure 2.4 that consists of a single basic constraint that references issues by their keys or IDs matches not only these issues, but all of their children as well.

That means that if you were using JQL of the form

```
issue in structure("My personal structure", TS-129)
```

then in Structure 2.4 this query will still return `TS-129` and all of its children at all levels (provided that `TS-129` is added to the structure.)

If this backward compatibility bites you (if, say, you need to check whether an issue is added to a structure), prepend the structure query with `issue in`:

```
issue in structure("My personal structure", "issue in TS-129")
```

This JQL will match only `TS-129` if it is in the structure.