

# Automation conditions

**Conditions** provide the necessary features to turn a rule into a structured vehicle to automate even the most complex processes.

Conditions can **limit** overall rule processing to specific issues meeting condition criteria, or define different kinds of processing depending on issue content.

Conditions can be added to:

- [triggers](#)
- [selectors](#), and even
- [conditions](#) to turn them into **nested conditions**.



Think of a condition as an "IF - THEN" clause. **Nested conditions** can act as "IF -THEN -ELSE" clauses.

## Available automation conditions

The following automation conditions are currently available:

- [Boolean condition](#)
- [JQL condition](#)
- [User condition](#)

The options actually available **depend** on which other elements (**triggers** and/or **selectors**) have been selected upfront.

## Nested conditions

Combining multiple similar conditions with different actions is the most common use case for a **nested condition**. Each rule can have **multiple nested conditions** (a condition **in** another condition). This makes customizing your rules very intuitive and helps you get things done in no time.



## When the priority changes



IF the priority has changed to **"Highest"**



**Assign** the issue to the product owner



**Update** all subtasks' priorities



IF the priority has changed to **"High"**



**Send an email** to the product owner



IF the current assignee is **NOT** a **developer**



**Assign** the issue to a developer



**Add a comment** and mention the new assignee

Added an action or condition at the wrong place? Just use these [awesome ui hacks](#) to **rearrange all elements** as needed!

If you still have questions, feel free to refer to our [support](#) team.