

Workflow use cases

The following page contains use cases for the **JWT workflow functions**. The number of use cases will grow over time, so make sure to visit this page once in a while.

 Use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Add a hyperlink to an issue in an email		Send email		
Add assignee as watcher to every blocking issue		Update linked issue or sub-task		
Add a table with the elements of a text to an email		Send email	toString() textOnStringList() toStringList()	
Add attachments from current issue to cloning issues		Update linked issue or sub-task		
Add comment with the request participants' display name		Add comment	toString() textOnStringList() toStringList() userDisplayName()	
Add days skipping weekends and holidays to a Date Picker field		Update or copy field values	addTime()	
Add internal Jira Service Management comment		Add comment		
Add or remove request participants		Update or copy field values		
Add percentaged profit margin when closing issue		Log work		
Add sub-tasks to an automatically created issue	 	Create issue	getMatchingValue()	
Add user to field depending on selected options		Update field based on rules		
Add watchers based on issue type		Add or remove watchers		
Add watchers ignoring inactive users		Update or copy field values	usersInGroup() isActive() toString() filterByPredicate()	

A developer must not execute the transition		User is not in project role
Alert the assignee of important issues		Add comment
Alert the reporter		Add comment
All blocking issues must be resolved	 	Condition on linked issues Validation of linked issues
All sub-tasks in the Closed status must have a specific resolution	 	Condition on sub-tasks Validation of sub-tasks
All sub-tasks must be Done or Closed	 	Condition on sub-tasks Validation of sub-tasks
All sub-tasks must be resolved	 	Condition on sub-tasks Validation of sub-tasks
All sub-tasks with a resolution of Done must be in a specific status	 	Condition on sub-tasks Validation of sub-tasks
An issue must have at least 3 resolved Test Cases	 	Condition on sub-tasks Validation of sub-tasks
Assignee may only have a restricted number of assigned issues	 	Condition based on JQL query Validation based on JQL query
Assign important issues to the project lead		Update or copy field values
Assign issue to current user		Update or copy field values
Assign issue to the default user of a role		Assign to project role
Automatically assign a tester during your development process		Assign to project role

STAFF PICK

Automatically create a version when starting the release		Execute remote action	
Automatically fill an insight custom field after a transition		Update or copy field values	
Automatically link an issue to an external one		Execute remote action	
Automatically log work on a Jira issue		Execute remote action	dateTimeToString()
Automatically log work spent in a specific status		Log work	timeDifference()
Block an Epic's transition depending on linked issues status and due date		Logical validator	count()
		Logical condition	filterByPredicate() linkedIssues()
Block an Epic's transition until all the issues under that Epic are resolved		Logical validator	count()
		Logical condition	issuesUnderEpic() filterByResolution()
Block a transition until all sub-tasks have certain fields populated		Logical validator	count()
		Logical condition	filterByPredicate() subtasks()
Calculate custom sequence numbers		Update or copy field values	toUpperCase() substring()
		Set or create JWT project property	length() projectProperty() toNumber()
Change the assignee to the next evaluator		Update or copy field values	first()
			toStringList()
Clone epic, tasks and sub-tasks		Create issue	toString() toStringList() issueKeysToIssueList() replaceFirst() first() fieldValue() textOnStringList() find Modify() subtasks() issuesUnderEpic()
Clone issue in external instance		Execute remote action	
Close parent issue only when all sub-tasks are closed		Condition on sub-tasks	
		Validation of sub-tasks	
Copy formatted description of issue into email		Send email	wikiToHTML()
Copy highest priority from linked issues		Copy field values from linked issues or subtasks	

STAFF PICK

Copy latest due date from sub-tasks		Copy field values from linked issues or subtasks	
Create a comment on an external Jira ticket		Execute remote action	
Create a component to group issues related to UI design		Execute remote action	
Create a new employee account during an onboarding process		Execute remote action	
Create an external project for a new employee during an onboarding process		Execute remote action	
Create an internal Service Management comment on linked issues		Update or copy field values	
Create an issue in the current project		Create issue	STAFF PICK
Create an issue with a custom summary		Create issue	previousValue()
Create an overview page for a software release		Execute remote action	
Create a personal space for a new employee		Execute remote action	
Create a sub-task for each component		Create issue	toStringList()
Create a sub-task for every recurring deadline within a task		Create issue	addMonths() getMatchingValue() dateToString()
Create a sub-task for every sub-task closed		Create issue	findReplaceFirst() first() findPattern() toString() sum() toNumber()
Create a sub-task for every user in a field and assign them		Create issue	getMatchingValue() toStringList()
Create child issue within Advanced Roadmaps hierarchy		Create issue	
Create Confluence page with links to issues		Execute remote action	
Create issues randomly		Create issue	modulus() round() random()
Create issue under epic		Create issue	toString() epic()

Create several issues combining fields		Create issue	nthElement() toStringList() modulus() count() ceil()	
Create several sub-tasks depending on the component		Create issue		
Create sub-tasks depending on selected values in custom field		Create issue	toStringList()	STAFF PICK
Create three issues with individual summaries		Create issue	getMatchingValue()	
Current user must be reporter		Compare two values condition		
		Compare two values validator		
Ensure that all issues linked with a certain issue link type have "Due Date" field set		Logical validator	count()	
		Logical condition	linkedIssues() fieldValue()	
Ensure that an issue has at least one attachment		Logical validator		STAFF PICK
		Logical condition		
Escalate issue to assignee's supervisor		Copy JWT user property	userProperty()	STAFF PICK
Evaluate Assets objects		Logical validator	findPattern() findReplaceAll() replaceAll() toStringList()	
		Logical condition		
Extract an email address from the last comment		Copy excerpted value		
Extract the issue priority from the summary		Copy excerpted value		
Fast-track transition issues assigned to the project lead		Transition issue		STAFF PICK
Format the issue's summary according to specified rules		Format field value		STAFF PICK
Get attributes of Assets objects		Execute remote action		
Get Checklist		Execute remote action		
Get Hubspot contact information		Execute remote action		

Halt a transition if an element is not contained in a list		Logical validator Logical condition	toStringList()
Hide transition from issue creator		Except users in field	
Hide transition to a previous status		Logical condition	
Inline images in a generated email		Send email	replaceAll()
Keep parent issue's priority in sync		Update or copy field values	
Keep the status of an issue and its linked issues in sync		Transition issue	
Keep the status of parents and sub-tasks in sync (post function use case)		Transition issue	
Keep track of important status updates		Update or copy field values Fields required or changed	dateTimeToString()
Link a Jira issue with the corresponding release ticket		Execute remote action	
Link issue to issue keys in its description		Create issue link	findPattern() toString()
Link to JQL query with information from issue			toString() textOnStringList() toStringList()
Link to the linked epic from the current issue			toString() fieldValue() issueKeysToIssueList()
Log absence time on another issue		Log work	stringToDate()
Make "Time spent" field required		Logical validator	
Make the assignee required		Fields required Fields required or changed	

Make the description required		Fields required Fields required or changed	
Match several values of a list		Update or copy field values	toString() distinct() filterByPredicate()
Move an issue to another issue type		Move issue	
Move an issue to another project		Move issue	
Notify the reporter of an issue about its status by a Telegram message		Execute remote action	
Obtain a value from an Elements Connect custom field		Send email	findPattern() toString()
Only users in a project role can execute a transition		Users are/aren't in project role (validator) Users are/aren't in project role (condition)	
Only user specified in project property is allowed to execute transition		Condition on JWT project property Validation of JWT project property	projectProperty()
Only watchers can execute a transition		Only users in a field	
Prevent a closed issue from being reopened after resting 1 week closed		Logical validator Logical condition	isInRole()
Prevent a transition depending of the number of components		Logical validator Logical condition	numberOfSelectedItems()
Prevent creation of issues with a duplicate summary		Validation based on JQL query	
Prevent external users from creating issues		Logical validator	matches() toString() textOnStringList() toStringList() userDisplayName()
Prevent having the same assignee in two sub-tasks			
Prevent issue creation if description contains less than 100 characters		Logical validator	replaceAll() length()

Prevent issue creation with the same field value		Validation based on JQL query Logical validator		STAFF PICK
Prevent issue creation with the same field value (Boolean)		Logical validator Validation based on JQL query	count() issuesFromJQL()	
Prevent issue from being "Closed" if blocking issues are not closed yet.		Validation of linked issues		
Prevent setting due dates outside business hours		Logical validator Logical condition	withinCalendar()	STAFF PICK
Prevent the creation of sub-tasks unless the parent issue is in a certain status		Logical validator		
Prevent transitioning when there is a blocking issue		Validation of linked issues		
Prevent users from selecting specific fix versions	 	Condition based on regular expression Validation based on regular expression		
Prioritize the issues globally		Update or copy field values	indexOf() previousValue() issuesFromJQL()	
Proceed with a task only when all sub-tasks are completed	 	Condition on sub-tasks Validation of sub-tasks		
Read the information from a Trello card		Execute remote action		
Reject duplicated file names in attachments	 	Logical validator Logical condition	count() toStringList() distinct()	
Remove blocking links when blocking issue is closed		Delete issue link		
Remove remaining estimate when an issue is closed		Update field action		
Reopen parent issue, if a sub-task is reopened		Transition issue		
Resolution must be empty	 	Fields required Fields required or changed		

Restrict issue creation per issue type and project role		Logical validator	isInRole()
Restrict the issue creation with specific issue types to certain project roles		Users are/aren't in project role (validator)	
Retrieve the assets of an issue in Jira cloud		Execute remote action	
Send email after transitioning to specific status		Send email	
Send email with the URL of the attachments included in the description		Send email	replaceAll() toString() attachmentUrls() findPattern()
Set a condition in a global transition which only applies in a certain status		Logical condition	
Set a date on the same week day on alternate weeks		Update or copy field values	addDays() nextDayOfTheWeek() modulus() weekOfTheYear() datePart()
Set a Date Picker field to the nth day of the month		Update or copy field values	dayOfTheMonth() lastDayOfTheMonth() addDays()
Set assignee based on a former assignee		Update or copy field values	previousValue()
Set assignee based on priority		Update field based on rules	priority()
Set assignee depending on issue type		Update field based on rules	issueType()
Set a watcher based on custom field value		Add or remove watchers	
Set Checklist		Execute remote action	
Set custom IDs for each issue		Set or create JWT project property	length() projectProperty()
		Update or copy field values	substring() toNumber() year()
Set Due Date with latest value among sub-tasks		Update or copy field values	fieldValue() max() siblingSubtasks()
Set fix version based on its start and release date		Update or copy field values	toString() textOnStringList() unreleasedVersions() startDates() releaseDates() first()
Set issue security level depending on reporter		Update field based on rules	issueSecurityLevel()

STAFF PICK

Set reporter as assignee if a User Picker field is empty		Update field based on rules	
Set specific default assignee if not set		Copy JWT project property	
Set the assignee of an external issue same as the transitioned issue		Execute remote action	
Set the next fix version		Update or copy field values	<code>floor()</code> <code>toNumber()</code> <code>substring()</code> <code>length()</code>
Shorten the summary to a maximum number of characters		Update or copy field values	<code>substring()</code> <code>length()</code>
Specific sub-tasks must be resolved		Condition on sub-tasks	
		Validation of sub-tasks	
Start Progress on parent, if sub-tasks are started		Transition issue	STAFF PICK
There must be at least one related Service Management request		Condition on linked issues	
		Validation of linked issues	
Transition an external Jira ticket based on the linked internal one		Execute remote action	
Translate the description		Execute remote action	<code>replaceAll()</code> <code>wikiToHTML()</code> <code>htmlToTxt()</code>
Triage issues created by email		Move issue	<code>toUpperCase()</code>
			<code>toString()</code>
			<code>findPattern()</code>
			<code>isInGroup()</code>
Validate a Select List (cascading) custom field		Logical validator Logical condition	
Validate only issue links created in transition screen		Logical validator	<code>count()</code>
			<code>transitionLinkedIssues()</code>
			<code>filterByProject()</code>
			<code>filterByIssueType()</code>
Validation based on the value of a date type project property		Logical validator	<code>stringToDate()</code>
		Logical condition	<code>projectProperty()</code>

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