Validation and condition based on time expressions

On this page

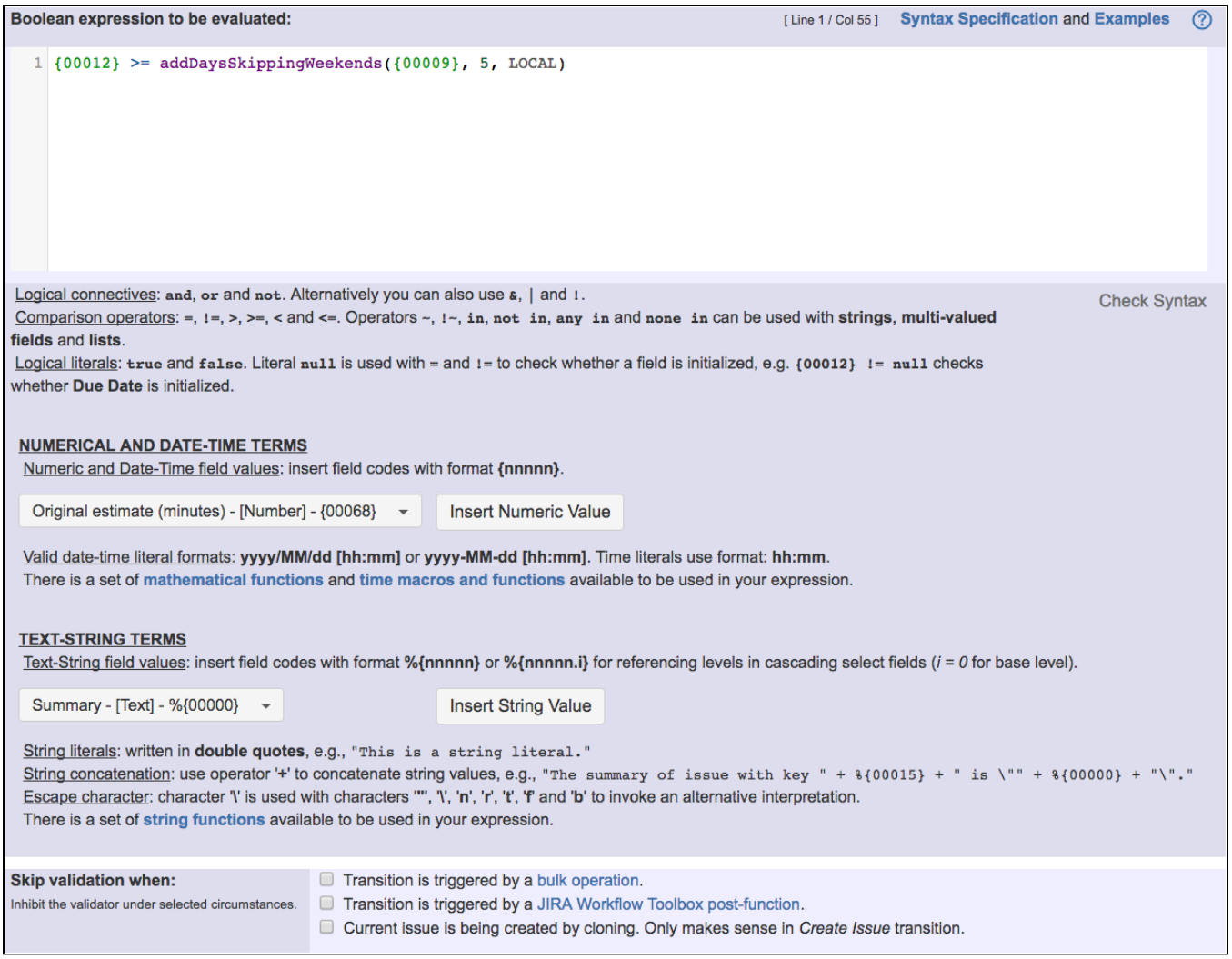
- Features used to implement the examples
- Example 1: Due Date should be at least 5 workdays posterior to Creation Date and Time
- Example 2: Define a time frame where transition execution is allowed
- Other examples of that function
- Related Usage Examples

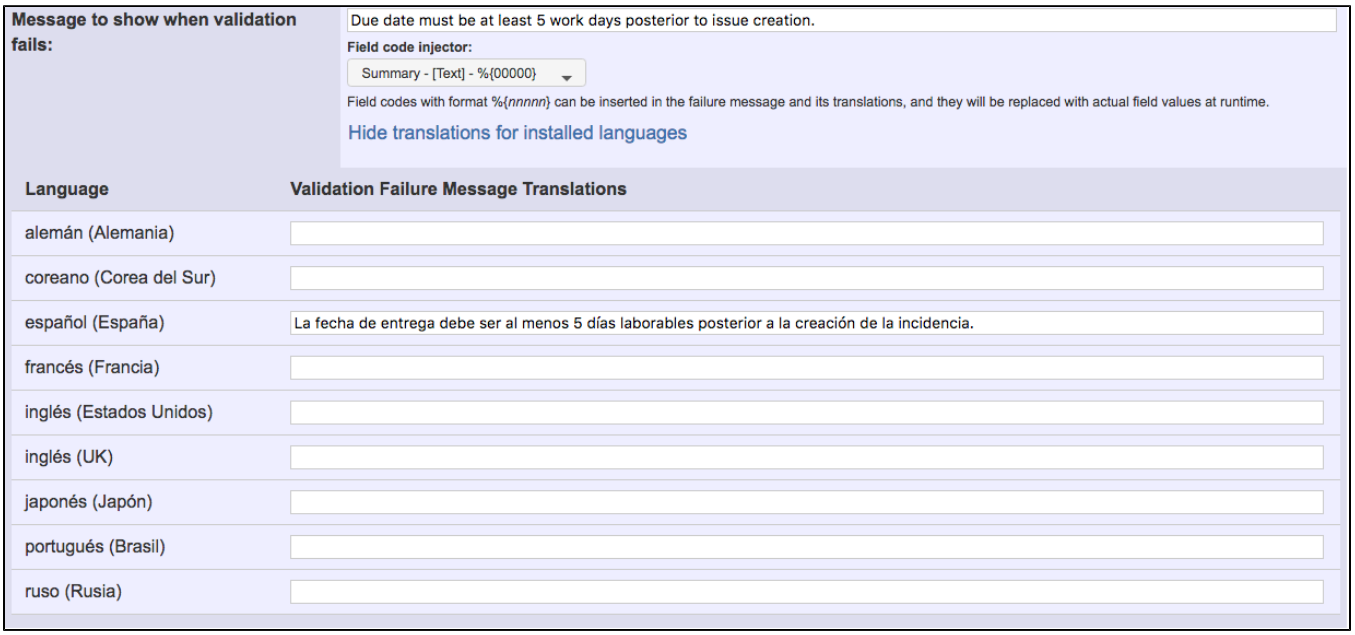
Features used to implement the examples

- **Boolean validator with math, date-time or text-string terms**

Example 1: Due Date should be at least 5 workdays posterior to Creation Date and Time

We want to ensure that "**Due date**" is at least 5 workdays later than issue date and time of creation. To do it we use validator **Boolean validator with math, date-time or text-string terms** with the following configuration:



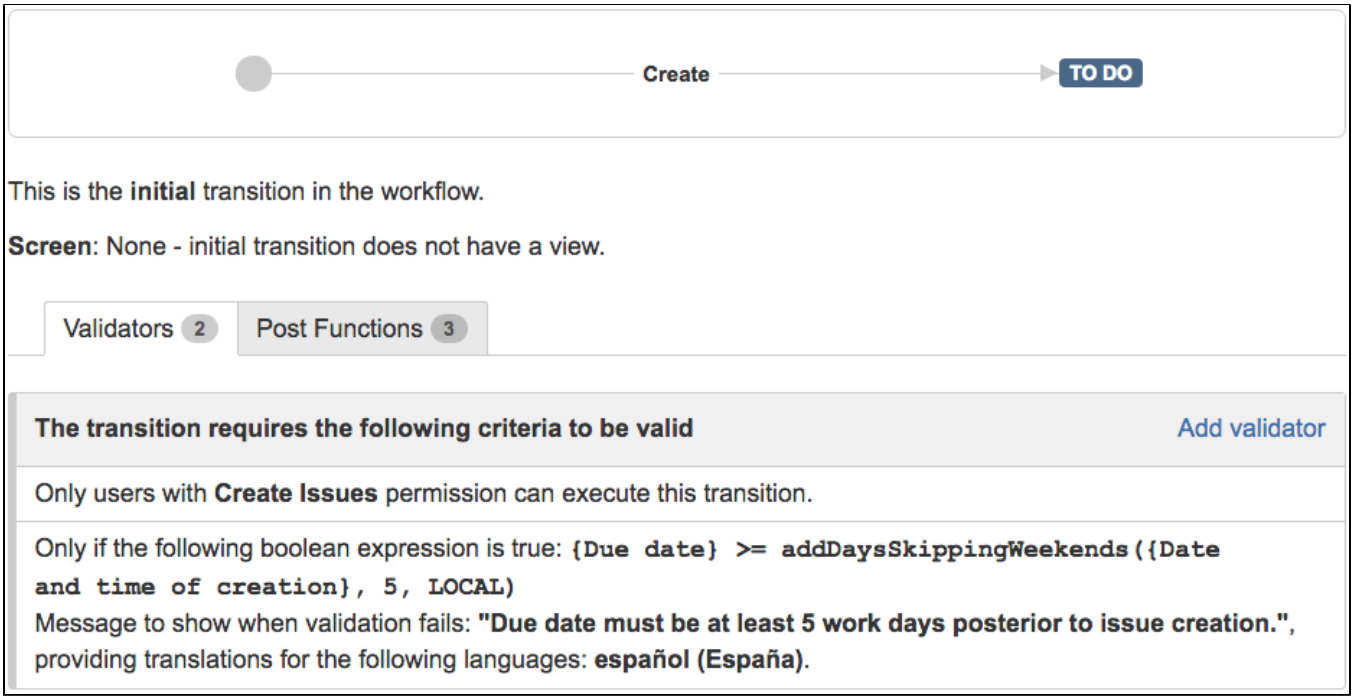


Boolean expression used is: **`{00012} >= addDaysSkippingWeekends({00009}, 5, LOCAL)`**

Note that:

- **{00012}** is code for numeric value of "**Due date**"
- **{00009}** is code for numeric value of "**Date and time of creation**"
- **LOCAL** represents server's timezone. Timezone is needed to determine weekend time periods. Use **USER_LOCAL** for using current user's timezone

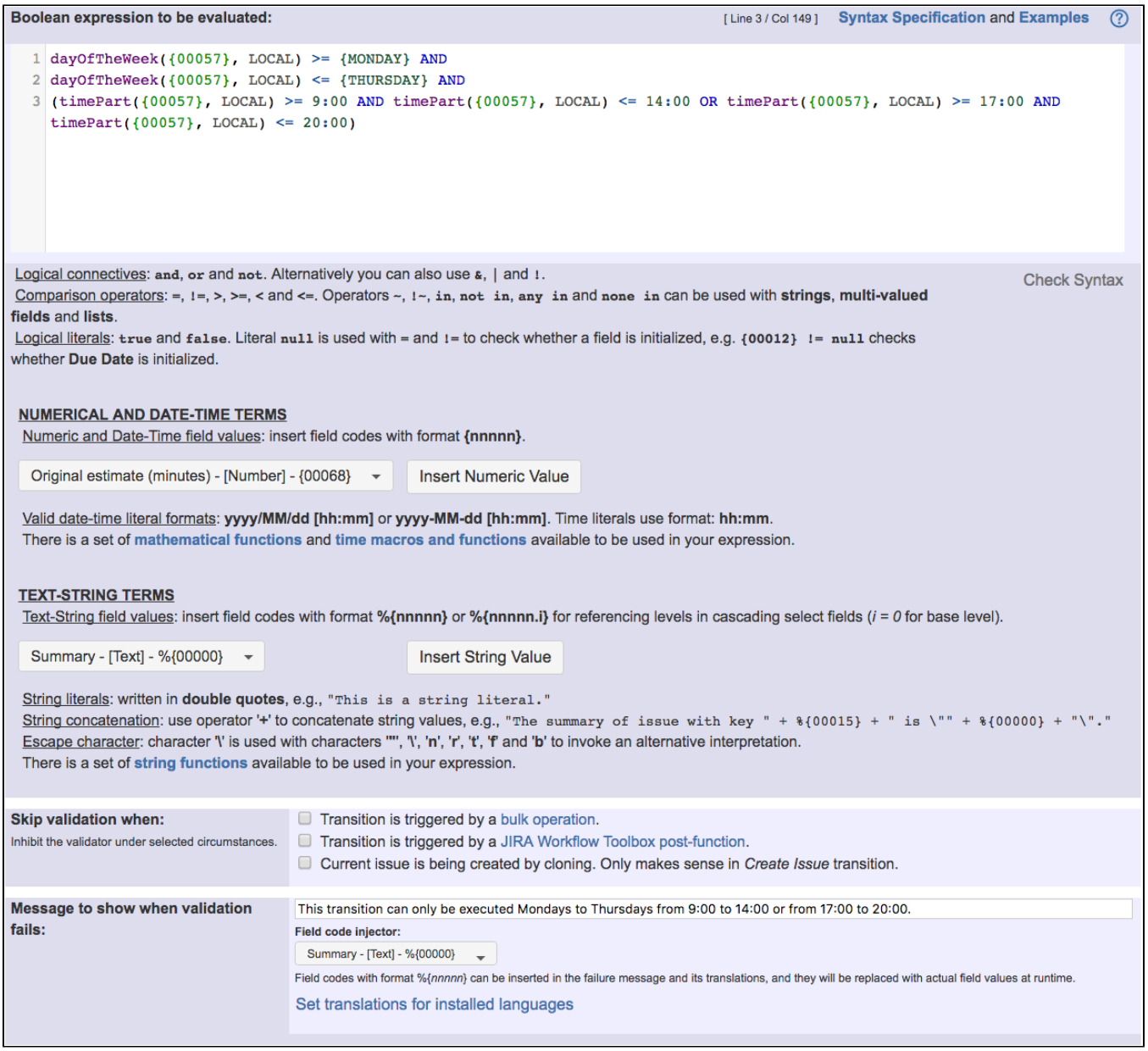
Once configured, the transition "**Create Issue**" will look like this:



Example 2: Define a time frame where transition execution is allowed

We want to allow the execution of a transition only **Mondays to Thursdays: from 9:00 to 14:00 and from 17:00 to 20:30**.

We can use **Boolean condition and validator with math. date-time or text-string terms** condition or validator depending on whether we want to **hide the transition**, or **show a custom error message** to the user when transition execution was not done in a valid time frame. In this example we are going to use the validator:

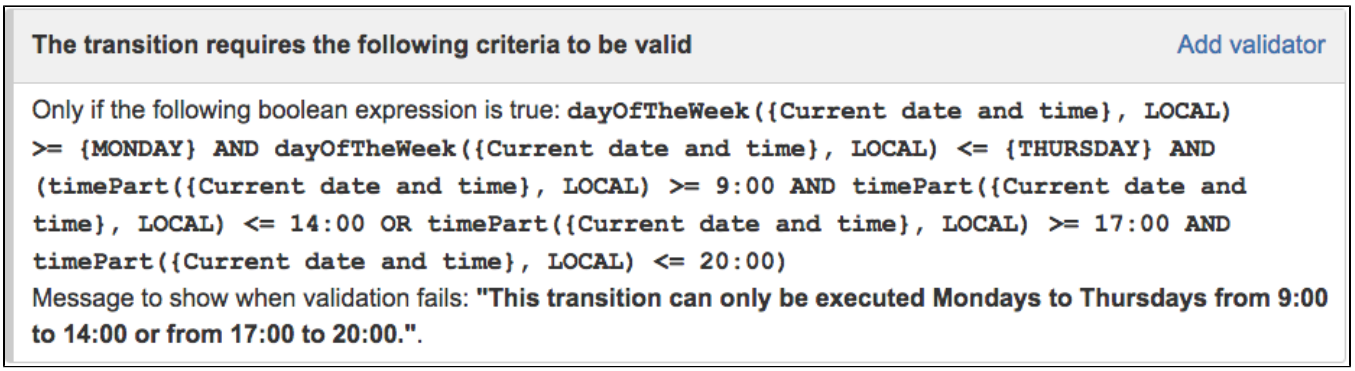


Boolean expression used is: **`dayOfTheWeek({00057}, LOCAL) >= {MONDAY} AND dayOfTheWeek({00057}, LOCAL) <= {THURSDAY} AND (timePart({00057}, LOCAL) >= 9:00 AND timePart({00057}, LOCAL) <= 14:00 OR timePart({00057}, LOCAL) >= 17:00 AND timePart({00057}, LOCAL) <= 20:00)`**

Note that:

- **{00057}** is code for numeric value of "**Current date and time**"
- **LOCAL** represents server's timezone. Use **USER_LOCAL** for using current user's timezone

Once configured, the transition will look like this:



Calculation Using Custom Schedules

Since version **2.2.39** time calculations on custom defined schedules can be carried out using **Schedules Definition Grammar**.

For example, you can define your work-schedule as Mondays to Fridays from 8:00 to 15:00, and from 16:00 to 20:00. Then you can calculate time differences, or add time durations to a base instant within that particular custom schedule.

Much more complex schedules than the one described in the example above are possible. You can define common and personal holidays, you can define different work-times for summer and winter seasons, change your work-times from a certain date on, keeping the older work-time definitions for previous instants. Almost any requirement you might think of is possible.

Other examples of that function

Page: Block a transition until all sub-tasks have certains fields populated
Page: Block an epic's transition depending on linked issues status and due date
Page: Block or hide a transition for an issue depending on its issue links
Page: Block or unblock a transition after an issue rested a specific time in a status
Page: Block transition until all sub-tasks are in a specific status category
Page: Close parent issue when all sub-tasks are closed
Page: Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)
Page: Ensure that all issues linked with a certain issue link type have "Due Date" field set
Page: If field A is populated then, field B must also be populated
Page: Limit issue creation per role and issue type
Page: Limit the number of hours a user can log per day
Page: Limit valid dates for work logs
Page: Make "Time Spent" field required when there is no time logged in the issue
Page: Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"
Page: Make attachment mandatory depending on the value of certain custom field
Page: Make different fields mandatory depending on the value of a Select List custom field
Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows
Page: Make parent issue progress through its workflow
Page: Prevent issue creation if another issue with same field value already exists
Page: Reject duplicated file names in attachments
Page: Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field
Page: Require issue link when resolving as duplicate
Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
Page: Restrict sub-task type creation depending on parent issue status
Page: Restrict sub-task type creation depending on parent issue type
Page: Set a condition in a global transition which only applies in a certain status
Page: Validate a custom field "Story Points" has been given a value in Fibonacci sequence

Related Usage Examples

- Block or unblock a transition after an issue rested a specific time in a status
  - example
  - condition
  - validator
  - transition
- Block transition until all sub-tasks are in a specific status category
  - example
  - transition
  - condition
- Validation and condition based on time expressions
  - example
  - condition
  - validator
  - transition
- Set a condition in a global transition which only applies in a certain status
  - example
  - condition
  - transition
- Validation on sibling sub-tasks depending on issue type and status
  - example
  - validator
  - sub-task
  - transition
- Block a transition until all sub-tasks have certains fields populated
  - example
  - condition
  - validator
  - sub-task
  - transition
- Block an epic's transition depending on linked issues status and due date
  - example
  - validator
  - issue-links
  - transition
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" (Transition issues)

Page: Validate compatible values selection among dependent custom fields
Page: Validate only issue links created in transition screen
Page: Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B
Page: Validation and condition based on time expressions
Page: Validation based on the value of a date type project property
Page: Validation on issue attachments
Page: Validation on MIME types of issue attachments
Page: Validation on sibling sub-tasks depending on issue type and status
Page: Validation on the value of a Cascading Select field

- example
- post-function
- transition

- Transition sub-tasks when parent is transitioned
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Transition only a sub-task among several ones
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Change parent's status depending on sub-task's summary (Transition issues)
  - example
  - post-function
  - transition
- Automatically close resolved sub-tasks when parent issue is closed
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Change parent's status depending on sub-task's summary
  - example
  - post-function
  - sub-task
  - transition
  - outdated