

Numbers, Dates and Times

Fixed values

| Input | Format | Example |
|-------------------------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Valid numerical values | | <ul style="list-style-type: none">• 1• 3.0• .5• -400• -1.1• -.02 |
| Valid Date-time values | yyyy/MM/dd [hh:mm] or yyyy-MM-dd [hh:mm] | <ul style="list-style-type: none">• 2018/03/25 23:15• 2018-03-25 23:15• 2018/03/25• 2018-03-25 |
| Valid Time values | hh:mm | <ul style="list-style-type: none">• 08:15• 23:59• 00:00 |

On this page

- [Fixed values](#)
- [Variable values \(field values\)](#)
- [Math Functions](#)
- [Date-Time Functions](#)
- [Time Macros](#)
- [Examples](#)

Variable values (field values)

Numeric values of **Number**, **Date**, **Date-Time** and **Priority** [data type](#) fields can be inserted in expressions with following notation `{...somefield}`, e.g., use `{...duedate}` for **Due Date**, and `{...numberOfAttachments}` for **Number of attachments**.



Pro tip

For checking if a field has a value you can use `{...somefield} = null` or `{...somefield} != null`

Math Functions

| Function | Input | Returned value |
|---------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| abs (number x) | NUMBER | Returns the absolute value of x , i.e., if x >0 it returns x , otherwise it returns -x . |
| acos (number x) | NUMBER | Returns the arc cosine of x ; the returned angle is in the range 0.0 through pi. |
| asin (number x) | NUMBER | Returns the arc sine of x ; the returned angle is in the range 0.0 through pi. |
| atan (number x) | NUMBER | Returns the arc tangent of x ; the returned angle is in the range 0.0 through pi. |
| ceil (number x) | NUMBER | Returns the smallest (closest to negative infinity) value that is larger than or equal to x and is equal to a mathematical integer. |
| cbrt (number x) | NUMBER | Returns the cube root of x . |
| cos (number x) | NUMBER | Returns the trigonometric cosine of angle x expressed in radians. |
| cosh (number x) | NUMBER | Returns the hyperbolic cosine of x . |
| floor (number x) | NUMBER | Returns the largest (closest to positive infinity) value that is less than or equal to x and is equal to a mathematical integer. |

| | | |
|--------------------------------------------------------------------|--------|-----------------------------------------------------------------------------------------------------------------------------------|
| log (number x) | NUMBER | Returns the natural logarithm (base e) of x . |
| log10 (number x) | NUMBER | Returns the base 10 logarithm of x . |
| max (number x , number y) | NUMBER | Returns the larger of two numeric values. |
| min (number x , number y) | NUMBER | Returns the smaller of two numeric values. |
| modulus (number dividend , number divisor) | NUMBER | Returns dividend - (divisor * floor (dividend / divisor)) . |
| pow (number x , number y) | NUMBER | Returns x raised to the power y . |
| random () | NUMBER | Returns a value with a positive sign, greater than or equal to 0.0 and less than 1.0. |
| remainder (number dividend , number divisor) | NUMBER | Returns dividend - divisor * n , where n is the closest integer to dividend / divisor . |
| round (number x) | NUMBER | Returns the closest integer to x . |
| sin (number x) | NUMBER | Returns the trigonometric sine of angle x expressed in radians. |
| sinh (number x) | NUMBER | Returns the hyperbolic sine of x . |
| sqrt (number x) | NUMBER | Returns the square root of x . |
| tan (number x) | NUMBER | Returns the trigonometric tangent of angle x expressed in radians. |
| tanh (number x) | NUMBER | Returns the hyperbolic tangent of x . |
| toDegrees (number x) | NUMBER | Converts an angle x measured in radians to an approximately equivalent angle measured in degrees. |
| toRadians (number x) | NUMBER | Converts an angle x measured in degrees to an approximately equivalent angle measured in radians. |

Date-Time Functions

Fields of type **Date** and **Date-Time** contain a **numeric value** with the **milliseconds elapsed since January 1, 1970, 00:00:00 GMT**. We usually need to get significant numbers from this numeric value, like YEAR, MONTH, DAY, HOUR, MINUTE, etc.

To achieve this, **Automation Toolbox for Jira** provides a comprehensive set of functions, most of them with **TIMEZONE** as input argument, since any significant number relative to a timestamp depends on the timezone.

| Function | Input | Returned value |
|----------------------------------------------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timePart (number t , timeZone time_zone) | TIMEZONE | Returns the time part of timestamp represented by numeric value t in time_zone time zone. Example: for timestamp March, 25th 2011 23:15 this function returns a NUMBER representing time 23:15 in milliseconds |
| datePart (number t , timeZone time_zone) | TIMEZONE | Returns the date part of timestamp represented by numeric value t in time_zone time zone. Example: for timestamp March, 25th 2011 23:15 this function returns a NUMBER representing date March, 25th 2011 00:00 in milliseconds |

| | | |
|---------------------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| second (number t , timeZone time_zone) | TIMEZONE | Returns the seconds figure of timestamp represented by numeric value t in time_zone time zone. Example: for timestamp March, 25th 2011 23:15:30 this function returns a NUMBER representing 30 seconds in milliseconds. |
| minute (number t , timeZone time_zone) | TIMEZONE | Returns the minutes figure of timestamp represented by numeric value t in time_zone time zone. Example: for timestamp March, 25th 2011 23:15:30 this function returns a NUMBER representing 15 minutes in milliseconds. |
| hour (number t , timeZone time_zone) | TIMEZONE | Returns the hours figure of timestamp represented by numeric value t in time_zone time zone. Example: for timestamp March, 25th 2011 23:15:30 this function returns a NUMBER representing 23 hours in milliseconds. |
| dayOfTheWeek (nu mber t , timeZone ti me_zone) | NUMBER | Returns the day of the week of timestamp represented by numeric value t in time_zone time zone, with Sunday = 1, Monday = 2, ... Saturday = 7. Example: for timestamp March, 25th 2011 23:15 this function returns 6 for Friday as a NUMBER , represented also by macro {FRIDAY} . |
| dayOfTheMonth (nu mber t , timeZone ti me_zone) | NUMBER | Returns the day of the month of timestamp represented by numeric value t in time_zone time zone. Example: for timestamp March, 25th 2011 23:15 this function returns 25 as a NUMBER . |
| month (number t , timeZone time_zone) | NUMBER | Returns the month of a timestamp represented by numeric value t in a certain time zone, with January = 1, February = 2, ... December = 12. Example: for timestamp March, 25th 2011 23:15 this function returns 3 for March as a NUMBER , represented also by macro {MARCH} . |
| year (number t , timeZone time_zone) | NUMBER | Returns the year of a timestamp represented by numeric value t in a certain time zone. Example: for timestamp March, 25th 2011 23:15 this function returns 2011 as a NUMBER . |
| addDays (number t , number n , timeZone time_zon e) | NUMBER | Returns a timestamp as a NUMBER resultant of adding n days to timestamp t . You should use this function instead of simply adding n * {DAY} , since {DAY} is a macro equivalent to 24 * {HOUR} , not taking into account that once in a year we have a day with 25 or 23 hours due to DST transition. Negative values for n are used in order to subtract instead of adding. Example: addDays(2018/03/27 01:00, -2, LOCAL) returns 2018/03/25 01:00 . |
| addMonths (number t , number n , timeZone time_zone) | NUMBER | Returns a timestamp resultant of adding n months to timestamp t . You should use this function instead of simply adding n * {MONTH} , since {MONTH} is a macro equivalent to 30 * {DAY} , not taking into account that some months has more or less than 30 days. Negative values for n are used in order to subtract instead of adding. Example: for timestamp t with value March, 25th 2011 23:15 calling to addMonths(t, 3, LOCAL) will return a timestamp as a NUMBER with value June, 25th 2011 23:15 |

| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| addYears (number t , number n , timeZone time_zone) | NUMBER | Returns a timestamp resultant of adding n years to timestamp t . You should use this function instead of simply adding $12 * \{\text{MONTH}\}$ or $365 * \{\text{DAY}\}$, since that won't take into account that some years have 366 days. Negative values for n are used in order to subtract instead of adding. Example: for timestamp t with value March, 25th 2011 23:15 calling to addYears(t, 10, LOCAL) will return a timestamp as a NUMBER with value March, 25th 2021 23:15 |
| addTimeSkippingWeekends (number t , number timeToBeAdded , timeZone time_zone) | NUMBER | Adds timeToBeAdded to t and returns a NUMBER with the difference that weekends don't count in the sum, e.g., if t represents a date-time which coincides with a Saturday, adding timeToBeAdded = 2 * {HOUR} will return a date-time for next Monday at 02:00 . Use negative values at timeToBeAdded for subtracting time from t . |
| addTimeSkippingWeekends (number t , number timeToBeAdded , timeZone time_zone , number beginning_of_weekend , number end_of_weekend) | NUMBER | Same as previous function, returns a NUMBER but with a custom defined weekend. Arguments beginning_of_weekend and end_of_weekend take values {MONDAY}, {TUESDAY} ... {SUNDAY} . Example of usage for adding 12 hours to Current date and time using Israeli weekend: addTimeSkippingWeekends({...currentDateTime}, 12 * {HOUR}, LOCAL, {FRIDAY}, {SATURDAY}) |
| addDaysSkippingWeekends (number t , number n , timeZone time_zone) | NUMBER | Returns a timestamp as a NUMBER equivalent of $t + n * \{\text{DAY}\}$ with the difference that weekends don't count in the sum, e.g., if t represents a timestamp which coincides with a Friday, adding n = 1 will return a date-time for next Monday. Negative values for n are used in order to subtract days to t . |
| addDaysSkippingWeekends (number t , number n , timeZone time_zone , number beginning_of_weekend , number end_of_weekend) | NUMBER | Same as previous function, returns a NUMBER but with a custom defined weekend. Arguments beginning_of_weekend and end_of_weekend take values {MONDAY}, {TUESDAY} ... {SUNDAY} . Example of usage for adding 10 workdays to Due date using Israeli weekend: addDaysSkippingWeekends({...duedate}, 10, LOCAL, {FRIDAY}, {SATURDAY}) |
| subtractDatesSkippingWeekends (number minuend_date , number subtrahend_date , timeZone time_zone) | NUMBER | Returns a timestamp as a NUMBER equivalent "minuend_date - subtrahend_date" subtracting weekend periods from the result, i.e., you get the elapsed working time from subtrahend_date to minuend_date . |
| subtractDatesSkippingWeekends (number minuend_date , number subtrahend_date , timeZone time_zone , number beginning_of_weekend , number end_of_weekend) | NUMBER | Same as previous function, returns a NUMBER but with a custom defined weekend. Arguments beginning_of_weekend and end_of_weekend take values {MONDAY}, {TUESDAY} ... {SUNDAY} . Example of usage calculating the worktime from Creation to Resolution using Israeli weekend: subtractDatesSkippingWeekends({...resolutiondate}, {...created}, LOCAL, {FRIDAY}, {SATURDAY}) |
| dateToString (number t , timeZone time_zone , language) | NUMBER | Returns a STRING representing the date-time value at t , in a certain time zone , and in a certain language . This function is useful in the Action Update Field to represent as a string the result of a time expression. |
| dateTimeToString (number t , timeZone time_zone , language) | NUMBER | Returns a STRING representing the date-time value at t , in a certain time zone , and in a certain language . This function is useful in the Action Update Field to represent as a string the result of a time expression. |

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dateTimeToString (number t , string date_time_pattern , language) | NUMBER | Returns a STRING representing the date-time value at t with a certain custom format defined by date_time_pattern string parameter, using a certain language when using words for months, days of the week, etc. This function is useful in Action Update Field to represent as a string the result of a time expression. Example: <code>dateTimeToString(2011-03-25 11:30, "yyyy.MM.dd 'at' HH:mm:ss", USER_LANG)</code> returns string <code>"2011.03.25 at 11:30:00"</code> . |
| dateTimeToString (number t , string date_time_pattern , timeZone time_zone , language) | NUMBER | Returns a STRING representing the date-time value at t with a certain custom format defined by date_time_pattern string parameter, in a certain timeZone time_zone , using a certain language when using words for months, days of the week, etc. This function is useful in the Action Update Field to represent as a string the result of a time expression. Example: <code>dateTimeToString(0, "yyyy.MM.dd 'at' HH:mm:ss", GMT, USER_LANG)</code> returns string <code>"1970.01.01 at 00:00:00"</code> . Example: <code>dateTimeToString(0, "yyyy.MM.dd 'at' HH:mm:ss", MST, USER_LANG)</code> returns string <code>"1969.12.31 at 17:00:00"</code> . |
| monthToString (number t , timeZone time_zone , language) | NUMBER | Returns a STRING with the name of the month for a date-time t , in a certain time zone time_zone , and in a certain language . This function can be used in the Action Update Field to write the name of the month of a date-time field or expression. |
| dayOfTheWeekToString (number t , timeZone time_zone , language) | NUMBER | Returns a STRING with the day of the week for a date-time t , in a certain time zone time_zone , and in a certain language . This function is useful in the Action Update Field to write the day of the week of a date-time field or expression. |
| weekOfTheYear (number t , number firstDayOfTheWeek , number minimalDaysInFirstWeek , timeZone time_zone) Available since version 1.1.0 | NUMBER NUMBER NUMBER TIMEZONE | Returns the week of the year of the date-time t in a certain time_zone as NUMBER . The parameter firstDayOfTheWeek represents the first day of the week, e.g.: { SUNDAY } in the U.S., and { MONDAY } in Germany. The parameter minimalDaysInFirstWeek represents the minimal number of days required in the first week of the year, e.g., if the first week is defined as the one that contains the first day of the first month of the year, value 1 should be used. If the minimal number of days required must be a full week (e.g. all days of the week need to be in that year), value 7 should be used. Example: <code>weekOfTheYear(2023/01/03, {SUNDAY}, 1, LOCAL)</code> returns 1 . Example: <code>weekOfTheYear(2023/01/03, {MONDAY}, 1, LOCAL)</code> returns 2 . Example: <code>weekOfTheYear(2023/01/03, {MONDAY}, 7, LOCAL)</code> returns 1 . |
| dayOfTheYear (number t , timeZone time_zone) Available since version 1.1.0 | NUMBER TIMEZONE | Returns the day of the year of date-time t in a certain time_zone as NUMBER , e.g. for January 1st the value returned will be 1 . Example: <code>dayOfTheYear(2019/02/01, LOCAL)</code> returns 32 . |
| stringToDate (string s , timeZone time_zone) | TIMEZONE STRING | Returns a NUMBER with the date-time represented by string s . The numeric value returned corresponds to the milliseconds elapsed since January 1, 1970, 00:00:00 GMT . Valid input string formats are yy/MM/dd HH:mm , yyyy-MM-dd HH:mm , yyyy/MM/dd , yyyy-MM-dd , also formats relative to current time like in JQL queries: "w" (weeks), "d" (days), "h" (hours) or "m" (minutes), or format defined at system property jira.date.time.picker.java.format . |

| | | |
|---------------------------------------------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stringToDate (string s , string date_time_pattern) | STRING | <p>Returns a NUMBER with the date-time represented by string s. Expected format of value at parameter "s" is defined by date_time_pattern string parameter. The numeric value returned corresponds to the milliseconds elapsed since January 1, 1970, 00:00:00 GMT.</p> <p>Example: stringToDate("2011.03.25 at 11:30:00", "yyyy.MM.dd 'at' HH:mm:ss") returns a date-time numeric value that can be used for setting a Date Time picker custom field.</p> |
| stringToDate (string s , string date_time_pattern , string language , string country) | STRING | <p>Returns a NUMBER with the date-time represented by string s. Expected format of value at parameter "s" is defined by date_time_pattern string parameter for a specific language (language code ISO 639-2) and country (country code ISO 3166 alpha-2). The numeric value returned corresponds to the milliseconds elapsed since January 1, 1970, 00:00:00 GMT.</p> <p>Example: stringToDate("Dec 7, 2016 2:10:25 AM PST", "MMM d, yyyy h:mm:ss a z", "eng", "US") returns a date-time numeric value that can be used for setting a Date Time picker custom field.</p> |
| formatDuration (number duration) | DURATION | <p>Returns a STRING with the pretty representation of a time duration, i.e. a subtraction of 2 date-time values, using the language of current user's profile.</p> <p>Example: formatDuration(2017-01-31 11:30 - 2017-01-30 00:00) returns "1 day, 11 hours, 30 minutes" .</p> |
| shortFormatDuration (number duration) | DURATION | <p>Returns a STRING with the most compact representation possible of a time duration, i.e. a subtraction of 2 date-time values, using the language of current user's profile.</p> <p>Example: shortFormatDuration(2017-01-31 11:30 - 2017-01-30 00:00) returns "1d 11h 30m" .</p> |
| formatWorkDuration (number duration) | DURATION | <p>Returns a STRING similar to function formatDuration() but using the workday and workweek defined at time tracking configuration , instead of 24 hours per day and 7 days per week.</p> <p>Example: formatWorkDuration(5 * 8 * {HOURL} + 2 * 8 * {HOURL} + 3 * {HOURL}) returns "1 week, 2 days, 3 hours", with 8 hours per workday and 5 days per workweek.</p> |
| shortFormatWorkDuration (number duration) | DURATION | <p>Returns a STRING similar to function shortFormatDuration() but using the workday and workweek defined at time tracking configuration , instead of 24 hours per day and 7 days per week.</p> <p>Example: formatWorkDuration(5 * 8 * {HOURL} + 2 * 8 * {HOURL} + 3 * {HOURL}) returns "1w 2d 3h" , with 8 hours per workday and 5 days per workweek .</p> |
| timeZone (string timeZone_name) | TIMEZONE | <p>Returns the timeZone whose name is represented by string timeZone_name. This function is useful to obtain a timeZone from a string, like the value of a Project Properties.</p> <p>Example: timeZone("DST") returns DST timeZone.</p> |

| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>timeInValue(string field field, boolean expression predicate)</p> <p>Available since version 1.1.0</p> | <div>STRING</div> <div>BOOLEAN</div> | <p>Returns the <div>NUMBER</div> of milliseconds a string field with code %{nnnnn} of the current issue has had a value satisfying a boolean expression predicate, where the string value of the field with code %{nnnnn} is represented by ^%.</p> <p>Example: timeInValue(%{00000}, ^% ~~ "ERROR" OR ^% ~~ "WARNING") returns the number of milliseconds the field summary (field code %{00000}) of the current issue has contained any of the words "ERROR" or "WARNING", ignoring the case.</p> <p>Example: timeInValue(%{00094}, count(toStringList(^%, ",") > 1) returns the number of milliseconds the field components (field code %{00094}) of the current issue has contained more than one selected component.</p> <p>Example: timeInValue(%{00017}, ^% in ["Critical", "High"]) returns the number of milliseconds the field priority (field code %{00017}) of the current issue has had a value of Critical or High.</p> |
| <p>timeInValue(number field field, boolean expression predicate)</p> <p>Available since version 1.1.0</p> | <div>NUMBER</div> <div>BOOLEAN</div> | <p>Returns the <div>NUMBER</div> of milliseconds a number or date-time field with code {nnnnn} of the current issue has had a value satisfying a boolean expression predicate, where the numeric value of the field with code {nnnnn} is represented by ^.</p> <p>Example: timeInValue({00012}, ^ != null) returns the number of milliseconds the field Due date (field code {00012}) of the current issue has had a value.</p> <p>Example: timeInValue({10001}, ^ >= 5 AND ^ <= 10) returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) of the current issue has remained between 5 and 10.</p> <p>Example: timeInValue({10001}, modulus(^, 2) = 0) returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) of the current issue has had an even value (2, 4, 6,...).</p> |
| <p>timeInValue(string field field, issue list issues, boolean expression predicate)</p> <p>Available since version 1.1.0</p> | <div>STRING</div> <div>ISSUE []</div> <div>BOOLEAN</div> | <p>Returns the sum of milliseconds a string field with code %{nnnnn} has had a value satisfying a boolean expression predicate in distinct issues as <div>NUMBER</div>, where the string value of the field with code %{nnnnn} is represented by ^%.</p> <p>Example: timeInValue(%{00000}, subtasks(), ^% ~~ "ERROR" OR ^% ~~ "WARNING") returns the sum of milliseconds the summary fields (field code %{00000}) of all subtasks of the current issue have contained any of the words "ERROR" or "WARNING", ignoring the case.</p> <p>Example: timeInValue(%{00094}, epic(), count(toStringList(^%, ",") > 1) returns the number of milliseconds the components fields (field code %{00094}) in a linked Epic issue have contained more than one selected component.</p> <p>Example: timeInValue(%{00017}, filterByIssueType(linkedIssues(), "Bug, New Feature"), ^% in ["Critical", "High"]) returns the sum of milliseconds all linked Bugs and New Features of the current issue have had a priority (field code %{00017}) value of Critical or High.</p> |

| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>timeInValue(number field field, issue list issues, boolean expression predicate)</p> <p>Available since version 1.1.0</p> | <div>NUMBER</div> <div>ISSUE []</div> <div>BOOLEAN</div> | <p>Returns the sum of milliseconds a number or date-time field with code {nnnnn} has had a value satisfying a boolean expression predicate in distinct issues as <div>NUMBER</div>, where the numeric value of the field with code {nnnnn} is represented by ^.</p> <p>Example: <code>timeInValue({00012}, subtasks(), ^ != null)</code> returns the number of milliseconds the field due date (field code {00012}) of all subtasks of the current issue have had a value.</p> <p>Example: <code>timeInValue({10001}, epic(), ^ >= 5 AND ^ <= 10)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) of an Epic issue has had a value between 5 and 10.</p> <p>Example: <code>timeInValue({10001}, filterByIssueType(linkedIssues(), "Bug, New Feature"), modulus(^, 2) = 0)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) has had an even value in any linked Bug or New Feature.</p> |
| <p>timeInValue(string field field, boolean expression predicate, string schedule_name, timeZone time_zone)</p> <p>Available since version 1.1.0</p> | <div>STRING</div> <div>BOOLEAN</div> <div>STRING</div> <div>TIMEZONE</div> | <p>Returns the <div>NUMBER</div> of milliseconds a string field with code %{nnnnn} of the current issue has had a value satisfying a boolean expression predicate, where the string value of the field with code %{nnnnn} is represented by ^%. The time being calculated by this function is only counted during a defined schedule with name schedule_name for time zone time_zone.</p> <p>Example: <code>timeInValue(%{00000}, ^% ~~ "ERROR" OR ^% ~~ "WARNING", "schedule_name", LOCAL)</code> returns the number of milliseconds the field summary (field code %{00000}) of the current issue has contained any of the words "ERROR" or "WARNING", ignoring the case, within a schedule named schedule_name for the server's default time_zone.</p> <p>Example: <code>timeInValue(%{00094}, count(toStringList(^%, ",") > 1, "schedule_name", LOCAL)</code> returns the number of milliseconds the field components (field code %{00094}) of the current issue has contained more than one selected component, within a schedule named schedule_name for the server's default time_zone.</p> <p>Example: <code>timeInValue(%{00017}, ^% in ["Critical", "High"], "schedule_name", LOCAL)</code> returns the number of milliseconds the current issue has had a priority value of Critical or High (field code %{00017}), within a schedule named schedule_name for the server's default time_zone.</p> |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>timeInValue(number field field, boolean expression predicate, string schedule_name, timeZone time_zone)</p> <p>Available since version 1.1.0</p> | <div>NUMBER</div> <div>BOOLEAN</div> <div>STRING</div> <div>TIMEZONE</div> | <p>Returns the NUMBER of milliseconds of a number or date-time field with code {nnnnn} of the current issue has had a values satisfying a boolean expression predicate, where the numeric value of the field with code {nnnnn} is represented by ^. The time being calculated by this function is only counted during a defined schedule with name schedule_name for time zone time_zone.</p> <p>Example: <code>timeInValue({00012}, ^ != null, "schedule_name", LOCAL)</code> returns the number of milliseconds the field due date (field code {00012}) of the current issue has had a value, ignoring the case, within a schedule named "my_schedule" for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, ^ >= 5 AND ^ <= 10, "schedule_name", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) of the current issue has had a value between 5 and 10, within a schedule named schedule_name for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, modulus(^, 2) = 0, "schedule_name", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) in current issue has had an even value, within a schedule named schedule_name for the server's default time_zone.</p> |
| <p>timeInValue(string field field, issue list issues, boolean expression predicate, string schedule_name, timeZone time_zone)</p> <p>Available since version 1.1.0</p> | <div>STRING</div> <div>ISSUE []</div> <div>BOOLEAN</div> <div>STRING</div> <div>TIMEZONE</div> | <p>Returns the NUMBER of milliseconds a string field with code %{nnnnn} has had a value satisfying a boolean expression predicate in distinct issues, where the value of the field with code %{nnnnn} is represented by ^%. The time being calculated by this function is only counted during a defined schedule with name schedule_name for time zone time_zone.</p> <p>Example: <code>timeInValue(%{00000}, subtasks(), ^% ~~ "ERROR" OR ^% ~~ "WARNING", "my_schedule", LOCAL)</code> returns the sum of milliseconds the fields summary (field code %{00000}) of all subtasks of the current issue have have contained any of the words "ERROR" or "WARNING", ignoring the case, within a schedule named schedule_name for the server's default time_zone.</p> <p>Example: <code>timeInValue(%{00094}, epic(), count(toStringList(^%, ",") > 1, "my_schedule", LOCAL)</code> returns the number of milliseconds the field components (field code %{00094}) in the linked Epic issue has contained more than one selected component, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue(%{00017}, filterByIssueType(linkedIssues(), "Bug, New Feature"), ^% in ["Critical", "High"], "my_schedule", LOCAL)</code> returns the sum of milliseconds all linked Bugs and New Features of the current issue have had a priority (field code %{00017}) value of Critical or High., within a schedule named my_schedule for the server's default time_zone.</p> |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>timeInValue(number field field, issue list issues, boolean expression predicate, string schedule_name, timeZone time_zone)</p> <p>Available since version 1.1.0</p> | <div>NUMBER</div> <div>ISSUE []</div> <div>BOOLEAN</div> <div>STRING</div> <div>TIMEZONE</div> | <p>Returns the NUMBER of milliseconds number or date-time field with code {nnnnn} has had a value satisfying a boolean expression predicate in distinct issues, where the numeric value of the field with code {nnnnn} is represented by ^. The time being calculated by this function is only counted during a defined schedule with name schedule_name for time zone time_zone.</p> <p>Example: <code>timeInValue({00012}, subtasks(), ^ != null, "schedule_name", LOCAL)</code> returns the number of milliseconds the field due date (field code {00012}) of all subtasks of the current issue have had a value, within a schedule named "my_schedule" for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, epic(), ^ >= 5 AND ^ <= 10, "schedule_name", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) in the linked Epicissue has had a value between 5 and 10, within a schedule named "schedule_name" for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, filterByIssueType(linkedIssues(), "Bug, New Feature"), modulus(^, 2) = 0, "schedule_name", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) has had an even value in any linked Bug or New Feature, within a schedule named schedule_name for the server's default time_zone.</p> |
| <p>fieldChangeTimes(string field field, boolean expression predicate)</p> <p>Available since version 1.1.0</p> | <div>STRING</div> <div>BOOLEAN</div> | <p>Returns the timestamps as NUMBER [] of when a string value of field with code %{nnnnn} has changed satisfying a certain predicate that depends on the values of the field before and after the value change. The string value before the change is represented by ^0%, and after the change by ^1%. The timestamps are returned as a number list sorted in ascending order.</p> <p>Example: <code>fieldChangeTimes(%{00000}, ^0% !~~ "IMPORTANT" AND ^1% ~~ "IMPORTANT")</code> returns the list of timestamps when word "IMPORTANT" has been added to the current issue's summary (field code %{00000}) ignoring the case.</p> <p>Example: <code>fieldChangeTimes(%{00017}, ^0% = null AND ^1% != null)</code> returns the list of timestamps of when the issue priority (field code %{00017}) of the current issue has been set.</p> <p>Example: <code>fieldChangeTimes(%{00017}, ^0% not in ["Critical", "High"] AND ^1% in ["Critical", "High"])</code> returns the list of timestamps when current issue's priority (field code %{00017}) has become Critical or High.</p> |

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fieldChangeTimes (number field field , b oolean expression p redicate) Available since version 1.1.0 | <div>NUMBER</div> <div>BOOLEAN</div> | <p>Returns the timestamps as <div>NUMBER []</div> of when a numeric / date-time value of field with code {nnnnn} h as changed satisfying a certain predicate that depends on the values of the field before and after the value change. The numeric value before the change is represented by ^0, and after the change by ^1. The timestamps are returned as a number list sorted in ascending order.</p> <p>Example: fieldChangeTimes({00012}, ^0 < ^1) r eturns the timestamps of when the Due date (field code { 00012}) has been edited to a higher value.</p> <p>Example: fieldChangeTimes({10001}, abs(^0 - ^1) / ^0 >= 0.25) returns the timestamps of when a hypothetical numeric field called Passengers(field code {10001}) has been edited with a variation of at least 25% over its previous value.</p> |
| fieldChangeTimes (string field field , issue list issues , boolean expression predicate) Available since version 1.1.0 | <div>STRING</div> <div>ISSUE []</div> <div>BOOLEAN</div> | <p>Returns the timestamps as <div>NUMBER []</div> of when a string value of field with code %{nnnnn} in distinct parameter issues have changed satisfying certain predic ate that depends on the values of the fields before and after the value change. The string value before the change is represented by ^0%, and after the change by ^1%. The timestamps are returned as a number list containing a sequence of sorted numeric values in ascending order for each parameter issue.</p> <p>Example: fieldChangeTimes(%{00000}, subtasks(), ^0% !~~ "IMPORTANT" AND ^1% ~~ "IMPORTANT") returns the list of timestamps of when the word "IMPORTANT" has been added the the summary (field code %{00000}) of all current issue's subtasks, ignoring the case.</p> <p>Example: fieldChangeTimes(%{00017}, epic(), ^0% = null AND ^1% != null) returns the list of timestamps of when the issue priority (field code % {00017}) of the current issue's epic has been set.</p> <p>Example: fieldChangeTimes(%{00017}, linkedIssues("is blocked by"), ^0% not in ["Critical", "High"] AND ^1% in ["Critical", "High"]) returns the list of timestamps of when the priority(field code %{00017}) in all blocking linked issues has become Critical or High.</p> |
| fieldChangeTimes (number field field , is sue list issues , boolean expression predicate) Available since version 1.1.0 | <div>NUMBER</div> <div>ISSUE []</div> <div>BOOLEAN</div> | <p>Returns the timestamps as <div>NUMBER []</div> of when a numeric value of field with code {nnnnn} in distinct parameter issues have changed satisfying a certain pred icate that depends on the values of the fields before and after the value change. The numeric value before the change is represented by ^0, and after the change by ^1 . The timestamps are returned as a number list containing a sequence of sorted numeric values in ascending order for each parameter issue.</p> <p>Example: fieldChangeTimes({00012}, subtasks(), ^0 < ^1) returns the timestamps of when the due date (field code {00012}) has been edited to a higher value in any of the current issue's subtasks.</p> <p>Example: fieldChangeTimes({10001}, epic(), abs(^0 - ^1) / ^0 >= 0.25) returns the timestamps when a hypothetical numeric field called Passengers (field code {10001}) in the current issue's epic has been edited with a variation of at least 25% over its previous value</p> |
| lastFieldChangeTi me (string field field) Available since version 1.1.0 | <div>STRING</div> | <p>Returns the timestamp as <div>NUMBER</div> of most recent value update of a field with code %{nnnnn}.</p> <p>Example: lastFieldChangeTime(%{00000}) returns the timestamp of the last update of an issue's summary (field code {00000}).</p> |

Time Macros

Date-Time values are numeric values representing the number of **milliseconds** elapsed since **January 1, 1970, 00:00:00 GMT**.

Macros are **aliases for literal / fixed values**. A comprehensive set of time macros is provided to make your expressions more readable.

| Macro | Equivalent value |
|----------|---------------------------|
| {SECOND} | 1000 |
| {MINUTE} | 1000 * 60 |
| {HOUR} | 1000 * 60 * 60 |
| {DAY} | 1000 * 60 * 60 * 24 |
| {WEEK} | 1000 * 60 * 60 * 24 * 7 |
| {MONTH} | 1000 * 60 * 60 * 24 * 30 |
| {YEAR} | 1000 * 60 * 60 * 24 * 365 |

The following macros are available to be used with function `dayOfTheWeek(t, time_zone)`:

| Macro | Equivalent value |
|-------------|------------------|
| {SUNDAY} | 1 |
| {MONDAY} | 2 |
| {TUESDAY} | 3 |
| {WEDNESDAY} | 4 |
| {THURSDAY} | 5 |
| {FRIDAY} | 6 |
| {SATURDAY} | 7 |

The following macros are available to be used with function `month(t, time_zone)`:

| Macro | Equivalent value |
|-------------|------------------|
| {JANUARY} | 1 |
| {FEBRUARY} | 2 |
| {MARCH} | 3 |
| {APRIL} | 4 |
| {MAY} | 5 |
| {JUNE} | 6 |
| {JULY} | 7 |
| {AUGUST} | 8 |
| {SEPTEMBER} | 9 |
| {OCTOBER} | 10 |
| {NOVEMBER} | 11 |
| {DECEMBER} | 12 |

Examples

| Input | Output |
|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <code>(2 * 6) / 3</code> | Returns the result of a simple calculation: 4 |
| <code>{...duedate} + 2 * {DAY}</code> | Returns a date which is two days in the future of the current Due Date . |
| <code>round(({...duedate} - {...currentDateTime}) / {HOUR})</code> | Returns the number of hours from between the current date and time to Due Date . |