addDaysSkippingWeekends()

This function **adds** or **subtracts natural days** (or working days) to or from **any given date**.

Weekends will **not be counted** as working days.

Syntax

```
addDaysSkippingWeekends(timestamp, numberOfDays, timeZone) #Output: Number
```

Examples

Parser expression	Description
`addDaysSkippingWeekends({issue.dueDate}, 2, LOCAL)`	This example **adds 2 working days** to the issue's due date. • If the due date is set to a **Monday** the function will return a date-time for **Wednesday**. • If the due date is set to a **Friday** the function will return a date-time for **Tuesday**.
`addDaysSkippingWeekends({issue.dueDate}, -2, LOCAL)`	This example **subtracts 2 working days** from the issue's due date. • If the due date is set to a **Monday** the function will return a date-time for **Thursday**. • If the due date is set to a **Friday** the function will return a date-time for **Wednesday**.
`addDaysSkippingWeekends({issue.cf10100}, -6, USER_LOCAL)`	This example **subtracts 6 working days** from of a custom date-time field (with the ID 10100). Instead of the **Jira server's local time**, this example uses the **current user's time zone**.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`timestamp`	**NUMBER**	The parameter must be valid **timestamp**. Usually this value is retrieved from a field (e.g. due date, created date).
`numberOfDays`	**NUMBER**	The **offset** in days. **Negative** values are used to **subtract** days. Internally JWT multiplies this value with the time macro `{DAY}` to add full days to the timestamp. Learn more about time macros.
`timeZone`	**TIMEZONE**	The **time zone** used for the calculation.

Work days might depend on the **time zone** - it might be Sunday on the west coast of the US while at the same time it's already Monday in Australia.

Output

This function returns a [NUMBER] representing a timestamp

Variant of the function where you can additionally define the **start** and the **end** of the **weekend**.

This function is useful when the non-working days **differ from the standard** (Saturday/Sunday).

Syntax
`addDaysSkippingWeekends(timestamp, numberOfDays, timeZone, startOfWeekend, endOfWeekend) #Output: Number`

Examples

Parser expression	Description
`addDaysSkippingWeekends({issue.dueDate}, 10, LOCAL, {FRIDAY}, {SATURDAY})`	This example **adds 10 working days** to the issue's due date. **Fridays** and **Saturdays** are not counted and will be skipped..
`addDaysSkippingWeekends({issue.dueDate}, -2, LOCAL, {SUNDAY}, {TUESDAY})`	This example **subtracts 2 working days** from the issue's due date. **Sundays, Mondays** and **Tuesdays** are not counted and will be skipped..

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
timestamp	[NUMBER]	The parameter must be valid **timestamp**. Usually this value is retrieved from a field (e.g. due date, created date).
numberOfDays	[NUMBER]	The **offset** in days. **Negative** values are used to **subtract** days. Internally JWT multiplies this value with the time macro `{DAY}` to add full days to the timestamp. Learn more about time macros.
timeZone	[TIMEZONE]	The **time zone** used for the calculation.
startOfWeekend	[NUMBER]	Valid values are `{MONDAY}`, `{TUESDAY}` ... `{SUNDAY}`.
endOfWeekend	[NUMBER]	Valid values are `{MONDAY}`, `{TUESDAY}` ... `{SUNDAY}`.

Work days might depend on the **time zone** - it might be Sunday on the west coast of the US while at the same time it's already Monday in Australia.

Output

This function returns a [NUMBER] representing a timestamp

The output can be written into any Jira field of type **Date Picker** ore **Date Time Picker**.

Another very common use case is to use this function in one of the JWT calculated date-time fields.

If you want to subtract two date-time values you might want to have a look at the function subtractDatesSkippingWeekends().

Use cases and examples

Use case
No content found.