

addTime()

This function **adds time** to a given **timestamp** based on a [JWT calendar specification](#).

Syntax

addTime(timestamp, timeToAdd, calendarName, timeZone) #Output: Number

Examples

Assumption: A custom JWT calendar called "my_calendar" has been defined as follows:

```
MON - THU {
  08:00 - 15:00,
  16:00 - 19:30;
}

FRI {
  08:00 - 15:00;
}
```

Parser expression	Description
<code>addTime(2020/12/01 01:00, 8 * {HOUR} + 31 * {MINUTE}, "my_calendar", LOCAL)</code>	This example returns a timestamp representing: "2017/12/04 10:01"
<code>addTime(2020/04/20 20:30, - 5 * {HOUR}, "my_schedule", LOCAL)</code>	This example returns a timestamp representing: "2017/04/20 13:00"

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
timestamp	<input type="text" value="NUMBER"/>	The parameter must be valid timestamp . Usually this value is retrieved from a field (e.g. due date, created date).
timeToAdd	<input type="text" value="NUMBER"/>	Time to be added in milliseconds. Usually time macros will be used to convert to input to hours, days etc.
calendarName	<input type="text" value="TEXT"/>	The name of the used JWT calendar .
timeZone	<input type="text" value="TIMEZONE"/>	The time zone used for the calculation.

Output

This function returns a representing a timestamp.

Variant of the function where you can define an additional [JWT calendar specification](#).

Syntax

```
addTime(timestamp, timeToAdd, calendarName, additionalSpecifier, timeZone) #Output: Number
```

Examples

Assumption: A custom JWT calendar called **"my_calendar"** has been defined as follows:

```
MON - THU {  
    08:00 - 15:00,  
    16:00 - 19:30;  
}  
  
FRI {  
    08:00 - 15:00;  
}
```

Parser expression	Description
<pre>dateTimeToString(addTime(2020/12/01 9:00, 25 * {HOUR}, "my_calendar", "2020/12/02 {;}", LOCAL), LOCAL, USER_LANG)</pre>	<p>This example returns a timestamp representing:</p> <p>"04/Dec/20 13:00"</p> <p>A day consists of 10.5 hrs according to my_calendar. Adding 25 hrs to the specified timestamp, ignoring December 2nd, results in December 4th, 1PM.</p>

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
firstDate	<input type="text" value="NUMBER"/>	The parameter must be valid timestamp . Usually this value is retrieved from a field (e.g. due date, created date).
dateToSubtract	<input type="text" value="NUMBER"/>	The parameter must be valid timestamp . Usually this value is retrieved from a field (e.g. due date, created date).
calendarName	<input type="text" value="TEXT"/>	The name of the used JWT calendar .
additionalSpecifier	<input type="text" value="TEXT"/>	A text containing an additional JWT calendar specification .
timeZone	<input type="text" value="TIMEZONE"/>	The time zone used for the calculation.

Output



This function returns a 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](#).



Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Add days skipping weekends and holidays to a Date Picker field		Update or copy field values			<code>addTime()</code>
Predicted resolution date based on business hours			Date-time		<code>addTime()</code>