

sumUp formatting templates

On this page

[Velocity template](#) | [The velocity context](#) | [Templates](#) | [Default Format](#) | [Hours Format](#) | [Jira Timespan Format](#) | [Days, Hours and Minutes Format](#)

Within sumUp, **calculation rules** are used in nearly every context to tell the app which fields should be calculated and which **output format** should be used.

To **further customize** the output format, **velocity templates** can be used!

On this page you'll find some **predefined templates** that can be **used right away** to adapt a rule's output format.

Velocity template

To use a custom output format in the rule, select **Velocity template** as shown below:

Add new rule

Field

Time Spent

Mode

Sum up values

Rule name

Σ Time Spent

Σ ∅

Output format

☐ Default

☒ Velocity template

Template

```
#set($H = '#')
#if($value)
#set($v = $value.doubleValue()/3600.0)
$number.format("$H.$H$H", $v)h
#end
```

\$value contains an instance of BigDecimal or null.

[Test your template](#)

[Examples and Documentation](#)

The velocity context

Need help?

If you're unfamiliar with velocity templates, you might want to check out the [official documentation](#).

The Velocity Context is initialized with the following content:

key	Class	Description
\$value	java.lang.BigDecimal	The computed value.
\$number	com.atlassian.jira.util.velocity.NumberTool	A helper class to format numeric values.
SINCE VERSION 3.6.0 \$math	org.apache.velocity.tools.generic.MathTool	A helper class for doing math calculations.
\$jiraDateTimeUtils	<i>internal</i>	A helper class which has one function: <code>getTimeFormattedString(long value)</code> . This function formats timespan according to the time configuration in JIRA.
\$formatter	java.text.SimpleDateFormat	JavaSimpleDate format with format ("EEE, d MMM yyyy h:mm a")

These **keys can be used within the templates** to customize the format.

Templates

- **Default Format**

```
#if($value)
$value
#end
```

- **Hours Format**

```
#set($H = '#')
#if($value)
#set($v = $value.doubleValue()/3600.0)
$number.format("$H.$H$H", $v)h
#end
```

 Assumes that the value is in seconds.

- **Jira Timespan Format**

```
#if($value)
$jiraDateTimeUtils.getTimeFormattedString($value.longValue())
#end
```

 Assumes that the value is in seconds.

- **Days, Hours and Minutes Format**

```

#set($H = '#')
#set($intMin = 999999)
#set($intHour = 999999)
#set($Min = $value.doubleValue())
#set($Hour = 0)
#set($Day = 0)

#if($value)#foreach($numMin in [ 1..$intMin ] )
    #if($Min < 60 )
        #break
    #else
        #set($Min = $Min - 60)
        #set($Hour = $Hour + 1)
    #end
#end#foreach($numHour in [ 1..$intHour ] )
    #if($Hour < 24 )
        #break
    #else
        #set($Hour = $Hour - 24)
        #set($Day = $Day + 1)
    #end
#end

$number.format("$H.", $Day)$number.format("$H.", $Hour)$number.format("$H", $Min)
#end

```

i Assumes that the value is in minutes.

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