

# Calendars



Please bear with us as we improve this feature.

At the moment, there is no good user interface for editing calendars. We are going to ship it in future versions.

Currently, if you need to define a calendar, please refer to the technical documentation about JSON format or write us to [support@almworks.com](mailto:support@almworks.com) so we can help you.

Structure.Gantt allows you to define different calendars for the following:

- [Gantt chart itself](#)
- [Default resource](#) calendar
- [Individual resource](#) calendar

This is especially useful when you need more precise planning for resources working on different schedules.

Schedules are defined as Calendars, and Structure.Gantt comes with two predefined calendars: "Standard" (40-hour work week, 9-17 every workday) and "24 Hour", with all time marked as working time.

Jira administrators can also define additional calendars using the **Administration | Structure | Structure.Gantt | Work Calendars** page.

## Calendar format

A calendar is defined by a code in JSON format, which sets the working periods for each day of the week and lists exceptions.

### Basic structure

Every calendar consists either of a week schedule or list of exceptions, or both:

```
{
  "week": ...,
  "exceptions": ...
}
```



All calendar definitions should be wrapped in curly braces: { }

### Week schedule

Week schedule is defined by specifying work ranges for every weekday (monday, tuesday, wednesday, thursday, friday, saturday, sunday):

```

"week": {
  "monday": [
    {
      "start": 900,
      "finish": 1300
    },
    {
      "start": 1400,
      "finish": 1715
    }
  ],
  "tuesday": [
    {
      "start": 900,
      "finish": 1300
    },
    {
      "start": 1400,
      "finish": 1715
    }
  ]
}

```

Time is written using HHMM format, where HH is hour (0-24) and MM is minutes, for example: `1730` for 17:30 (5:30 pm) and `930` for 9:30.

You may skip a weekday either by skipping its definition completely or providing an empty list of ranges:

```

{"monday": []}

```

## Exceptions

Exceptions are defined for exact dates:

```

"exceptions": [
  {
    "date": 20170301,
    "workPeriods": []
  },
  {
    "date": 20170305,
    "workPeriods": [
      {
        "start": 900,
        "finish": 1300
      },
      {
        "start": 1400,
        "finish": 1715
      }
    ]
  }
]


```

Dates are written using YYYYMMDD format, where YYYY is a 4-digit year, MM is a month (1-12) and DD is a day (0-31).

The workPeriods parameter can be empty, which means it is a day off, or can contain custom start and finish times, which will override the times you have defined in the main scheme.

## Hierarchy


You can also organize calendars into a hierarchy via the "Based on" property, so you can create more specific calendars, based on a general calendar ("Standard", for example).

 In this case, existing calendar-week schedule values are overridden (i.e. the definition for "wednesday" in a child calendar will override the "wednesday" schedule of its parent calendar).

## Time zones

Calendars don't have time zone information and should be written in your local time. Time zones are provided separately and are taken either from the current user profile TZ (for the Gantt chart, configured at the User Profile page) or from default or individual resource settings:

### Resource Settings

Name  admin

Capacity


Default capacity (1) will be used if no capacity is specified.

Time zone

Work calendar

## Deleting Calendars

Calendars can be deleted (even the ones that are used by other calendars as Based On or by Gantt charts). Gantt will continue to work properly, even if a specified calendar cannot be found: a hard-coded calendar definition, similar to a default version of "Standard" calendar, will be used. However, this is not normal and should be fixed.

 A Gantt info popup will show a warning as soon as the Gantt calendar becomes unavailable.  
For a resource calendar issues, there is no notification shown.