

Export/Import JSON with Tom's Planner

Introduction	2
Some basic settings	3
Timeline settings	4
The Columns	5
The legend and Logo	6
Print settings	7
The grid itself	8
Dependencies	11
Appendix A period and symbol codes	13
Appendix B Durations and duration type	14
Appendix C change log	16

Introduction

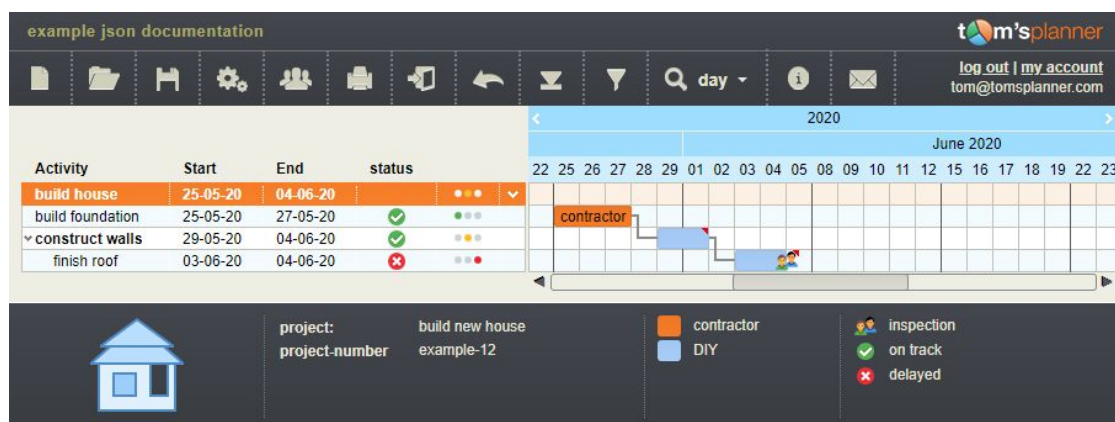
[JSON](#) is the format we use to store the schedules in our database. An export to json of your schedule contains all the information there is (including project information, timeline settings, last used printing settings etc.)

In this document we will describe the structure of version 15. In Appendix C you will find a change log. If no version number is available the JSON is from before version 1. If you open a schedule in the tool and save it again it will automatically update the JSON version to the most recent one.

Since there is no usable standard method to describe JSON structures we use an example JSON document and we will just walk you through it like a developer would do to another developer when they work side by side.

We will mainly explain the details that are not immediately obvious and will try to show with images which part of the JSON is linked to which part in the schedule. The example we are using can be downloaded at: <http://tomsplanner.com/documentation/example-json-version-15.txt>.

If you open this schedule it looks like this:

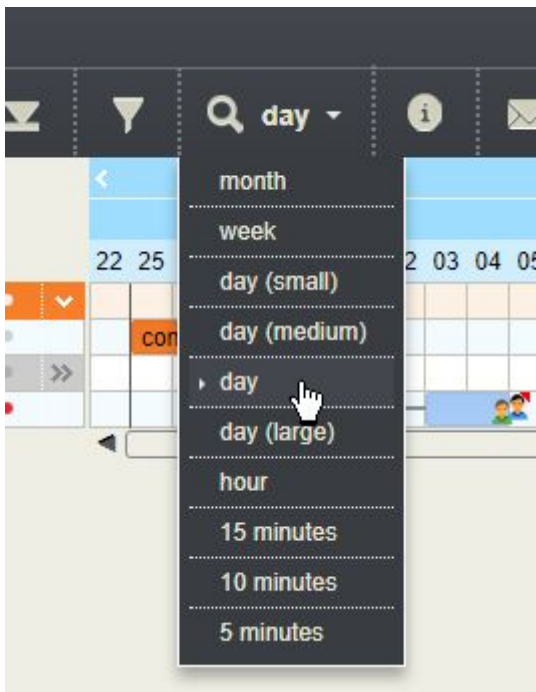


Unfortunately (and a bit embarrassingly) the use of capital letters in the property names is not done consistently throughout the JSON structure. So you will find the property name 'hideWeekendDays' capitalized but for instance the property name 'referencedate' is not capitalized. Apologies for that.

To be able to export to JSON you need a paid account of the 'Unlimited' type. If you however want to play around with the JSON a bit first it's good to know that in our [demo](#) you can save schedules offline and with a free personal account you can open/import these JSON files.

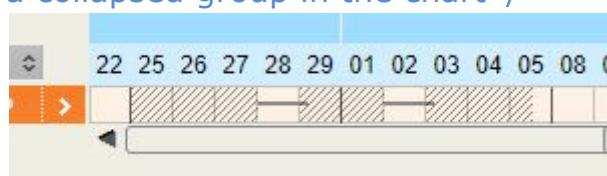
Some basic settings

```
{
"jsonversion":15,/*see Appendix C for the change log*/
"settings":{
  "defaultzoom":9,/*the default zoom level of the schedule*/
```

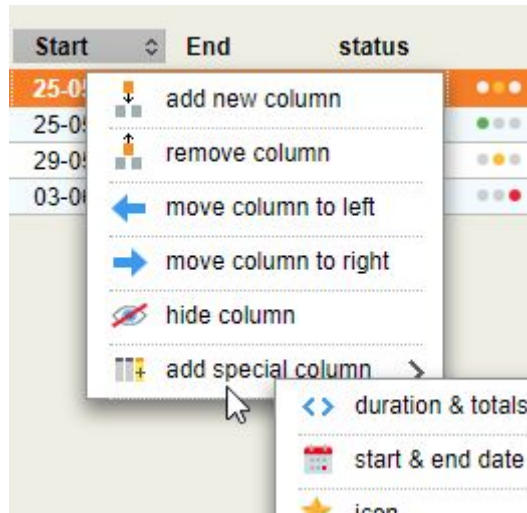


/*the zoom level is online stored in the database per schedule/user combination. If no zoom is available (for instance when you open a schedule that has been stored locally as a file or when you import a schedule) this default value is used. The following zoom levels are available: 3=month, 4=week, 6=day(small), 8=day(medium), 9=day, 10=day(larger), 11=hour, 12=15 minutes, 13=10 minutes, 14=5 minutes */

```
"plugins":{
  "groupduration":{"active":true},/*show the duration of
a collapsed group in the chart*/
```



"specialcolumns":{"active":true}/*enables user to use special column types*/

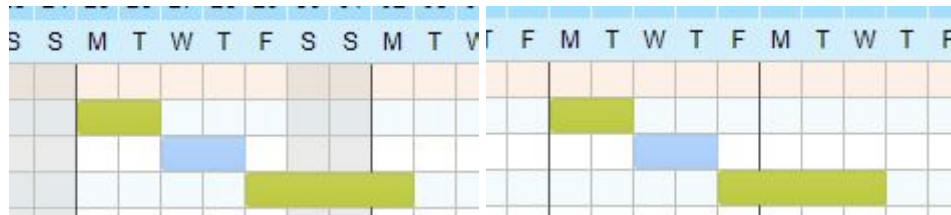


```
    },  
    "nature":"tomspawnermodel",/*Needs to be included! Has no  
    specific meaning*/
```

Timeline settings

```
"timeline":/*the settings of the blue timeline*/  
{  
  "activeHourFrom":9,/*start hour working day (integer)*/  
  "activeHourTo":17,/*end hour working day (integer)*/  
  "type":"day",/*possible values: day, hour, minutes15,  
  minutes10, minutes5. This relates to how you set durations in  
  the chart*/  
  "firstWeekDay":1,/*the start day of the week 0=sunday...  
  6=saturday*/  
  "showYear":true,/*show line with year numbers in timeline*/  
  "showMonth":true,/*show months in blue timeline*/  
  "showWeekNr":false,/*show week numbers in blue timeline*/  
  "showDate":true,/*show dates in blue timeline (1-31)*/  
  "showWeekDay":false,/*show weekdays in blue timeline  
  (mon-sun)*/  
  "showHour":false,/*hours in blue timeline*/  
  "showMinutes":false,/*show minutes in blue timeline*/  
  /*Note: some combinations of visible elements in the timeline  
  are not possible and will cause errors. Just go to the timeline  
  settings panel and see which options are available with  
  different timeline types. For instance if you have a timeline of  
  the type 'week' you will not be able to show the line with  
  hours.* /
```

"hideWeekendDays":true, /*determines if weekenddays should be hidden or shaded*/



/*shaded

hidden*/

"activeDays":[false,true,true,true,true,true,false],/*days in the week that need to be visible in the schedule. The first day is a Sunday and the last one is Saturday*/

"shadeDays":[false,false,false,false,false,false], /*The days that need to be shaded in the schedule. The first day is a Sunday and the last one is Saturday.*/

"timeFormat":"24hour", /* Determines how time looks in the timeline. '24hour' means two o'clock in the afternoon is 14 if this property is set to '12hour' two o'clock in the afternoon is 2*/

"markToday":true,/*determines if today's date is orange in timeline*/

"leftBorderDate":"05/22/2020 00:00:00 GMT"/*the horizontal scroll position of the grid when this chart is being opened. So in this example the left border of the blue timeline will be on the 22nd of may*/

},

The Columns

"columns":[/*Definition of the column area on the left side of your schedule. Each column is an entry in this array and has it's own type (see image below) and a colwidth property: */

```
{
  "colwidth":99,/*every column needs a width in pixels*/
  "indent":true,/*wether or not the texts in the column need to be indented when it's in a subgroup*/
  "type":"text",/*type can be "text","symbol","trafficlight",
  "startdate","enddate" or "duration"*/
  "full":"Activity"/*the header of the columns*/
},
```

```
{
  "colwidth":55,
  "type":"startdate"
```

```
},
{
  "colwidth":55,
```

```

"type":"enddate"
},
{
"colwidth":44,
"type":"symbol",
"full":"status",
"defaultvalue":0 /*determines the symbol that is added in this
column when a new row is inserted*/
},
{
"colwidth":30,
"type":"trafficlight",
"full":"",
"defaultvalue":2/*0=no traffic light, 1=red, 2=orange,
3=green*/
}
],

```

The legend and Logo

/*The metadata refers to this part of the schedule: */

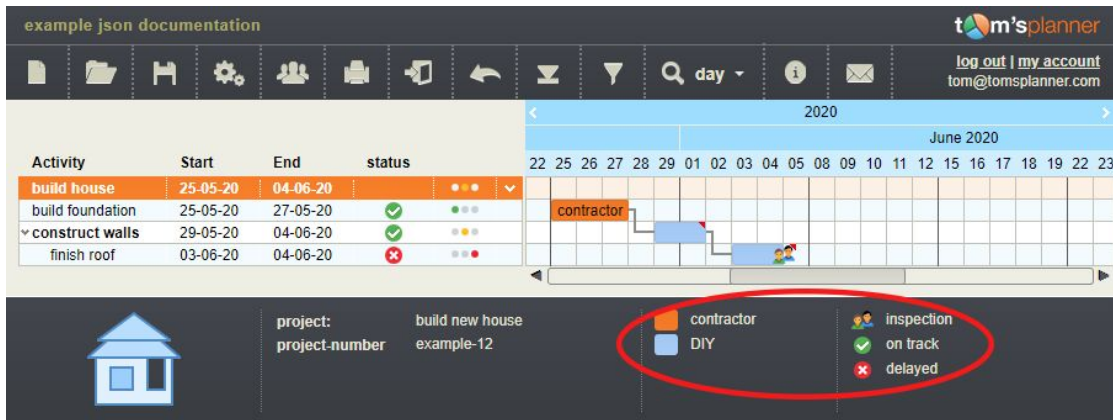
The screenshot shows the 'tom's planner' interface. At the top, there's a navigation bar with icons for home, folders, settings, users, print, copy, undo, redo, search, and a 'day' dropdown. The main area is divided into a table on the left and a Gantt chart on the right. The table has columns for Activity, Start, End, and status. The Gantt chart shows a timeline for June 2020 with tasks like 'contractor' and 'DIY'. At the bottom, there's a metadata section with a house icon, project name 'build new house', and project number 'example-12'. A legend below the metadata shows symbols for contractor (orange square), DIY (blue square), inspection (person icon), on track (green checkmark), and delayed (red X).

```

"metadata":[
  {"label":"project","labelvalue":"build new house"},
  {"label":"project-number","labelvalue":"example-12"}
],

```

/*The legend is marked in the image below. There is one array with the periods and one with symbols. Don't mix them up.*/



```

"legenda":{
  "periods":[
    {"label":"contractor","type":"fa8115"},/*The type is an
    rgb color code.*/
    {"label":"DIY","type":"a1c8f8"}
  ],
  "symbols":[
    {"label":"inspection","type":47},/* See appendix B for
    the symbol codes*/
    {"label":"on track","type":2},
    {"label":"delayed","type":1}
  ],
  "bolDisplay":true/*determines whether the legend (including
  the logo and meta data is shown at the bottom of the tool)*/
},
/*The logo is shown in the left bottom corner of the tool. The file
name is not an url so you can't include logos that are externally
hosted. The height of the logo needs to be given to avoid that the
legend needs to be re-rendered when the logo file is loaded*/
"logo":{
  "bolshow":true,/*is the logo visible in the legend or not*/
  "filename":"dcfdb6e6-9a8a-4cbe-b768-87287ebc9621.png"
  ,"height":80 /*height of image file in pixels*/
},

```

Print settings

```

"printsetup":{
  "format":"A4",/*possible values: letter,A4,legal,A3,ledger, A2,
  A1, A0*/
  "orientation":"landscape",/*possible values: landscape,
  portrait*/
  "colorscheme":"printerfriendly",/*possible values:
  printerfriendly, fullcolor*/

```

```

    "zoomtofit":true,/*determines if page should be
    printed on one page (true) or multiple pages (false)*/
    "range":{/*if no range is defined the schedule will be printed
    from the start to the end. But you can define a range that will
    be printed*/
        "startdate":"02/21/2012 00:00:00 GMT",
        "enddate":"03/11/2012 00:00:00 GMT"
    }
},

```

The grid itself

/*The grid constitutes of five types of objects: "grid", "fase" (group), "activity" (row), "period" and "symbol". The type is defined by the 'nature' property of each object. Every object has a property named "kids" which can contain other objects. A grid can contain one fase, a fase can contain multiple activities and an activity can contain multiple periods and symbols. Subgroups are defined by setting an indentation level for an activity/row*/

```

"visualGrid":{
    "nature":"grid",
    "id":"grid",/*this grid object has always an id: grid*/
    "lastID":37,/*every object in the schedule get's an unique id:
    'grid_'+an integer. Every time a new object is added the
    lastID counter is raised with one and used for the new id */
    "kids":[{
        /*this the root group and is a container which you won't
        see in the schedule itself, this is just a standard piece of
        the json and is always the same*/
        "nature":"fase",
        "id":"grid_1",/*this group object has always a id:
        grid_1*/
        "foldStatus":false,/*this group object has always a
        foldStatus: false*/
        "type":"fe8519",/*this group object has always a type:
        fe8519*/
        "kids":[
            {
                "nature":"fase",/*first group in the schedule*/
                "id":"grid_2",
                "foldStatus":false,/*determines if a group is collapsed or not*/
                "type":"fe8519",/*determines the background color of the group
                header. This is an rgb color code*/
                "kids":[{
                    "nature":"activity",/*the first row and the group header*/

```



```

"id":"grid_3",
"autonumber":"1",/*this property is used to enumerate the
rows. When a schedule is loaded into the tool this value will
be ignored and the value will always be recalculated*/
"indent":0,/*indentation of the row. You can use indentation
to create subgroups in your chart. The first row in the group is
the header of the group so the indent can only be 0. The
second row (and first row under the group header) can not be
indented yet because each subgroup needs it's header row so
the second row also always has an indent of 0. The third row
(and second row under the group header) can be the first row
in a subgroup and so it can have an indentation of 1*/
"kids":[],
"label":[/*the label array contains the column data of the
specific row/activity*/
{"full":"build house"},/*the first text column*/
{}/*the start date column as has an empty object*/
{}/*the end date column as has an empty object*/
{"colvalue":0},/*the status column, see appendix A for
the symbol codes*/
{"colvalue":2}/*the fifth column is of type `trafficlight,
0=no traffic light, 1=red, 2=orange, 3=green */
]
},{
"nature": "activity",/*the second row in the chart right under
the group header row*/
"id":"grid_4",
"autonumber":"1.1",
"indent":0,
"kids":[{"/*the kids array of an activity can contain periods
and symbols*/
"nature":"period",/*the first orange time block in this
chart*/
"kids":[],
"id":"grid_33",
"type":"fa8115",/*RGB code with the color of the time
block. */
"duration":6,/* see appendix B of this
document */
"durationtype":"day",/* see appendix B of this
document */
"startmoment":"05/25/2020 09:00:00 GMT",
"finishmoment":"05/27/2020 17:00:00 GMT",/*the
duration overrides the finishmoment. Finishmoment is
result of startmoment, duration and durationtype*/
"label":"contractor",/*optional property with the text
inside a time block*/

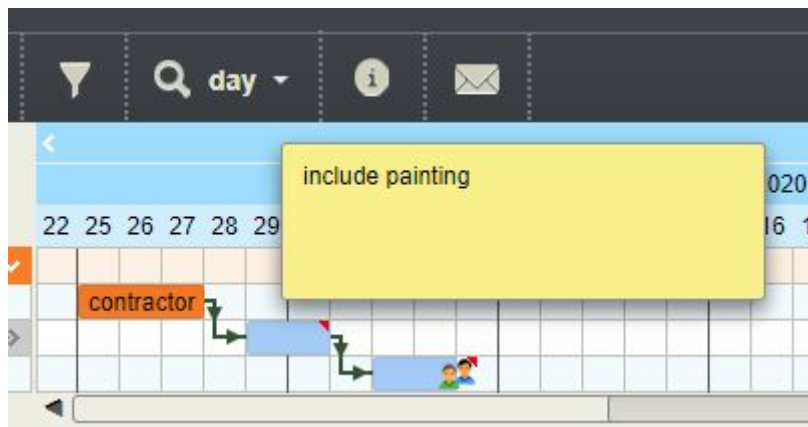
```



```

}],
"label": [
  {"full": "build foundation"},
  {}},
  {}},
  {"colvalue": 2},
  {"colvalue": 3}
]}, {
  "nature": "activity",
  "id": "grid_5",
  "autonumber": "1.2",
  "indent": 0,
  "kids": [
    {"nature": "period",
      "kids": [],
      "id": "grid_34",
      "type": "a1c8f8",
      "duration": 4,
      "durationtype": "day",
      "startmoment": "05/29/2020 09:00:00 GMT",
      "finishmoment": "06/01/2020 17:00:00 GMT",
      "comment": "include painting" /*optional property with a
comment. Time blocks and icons can have comments*/

```



```

}],
"label": [
  {"full": "construct walls"},
  {}},
  {}},
  {"colvalue": 2},
  {"colvalue": 2}
]}, {

```

```
"nature": "activity",
"id":"grid_6",
"autonumber":"1.2.1",
"indent":1,/*first row in a subgroup*/
```

Activity	Start	End	sta
build house	25-05-20	04-06-20	
build foundation	25-05-20	27-05-20	
▼ construct walls	29-05-20	04-06-20	
finish roof	03-06-20	04-06-20	

```
"kids":[
  {"nature":"symbol",
  "kids":[],
  "id":"grid_37",
  "type":47,/*see appendix A for the symbol codes*/
  "moment":"06/04/2020 17:00:00 GMT",
  "comment":"call supervisor"
  },
  {"nature":"period",
  "kids":[],
  "id":"grid_35",
  "type":"a1c8f8",
  "duration":4,
  "durationtype":"day",
  "startmoment":"06/03/2020 09:00:00 GMT",
  "finishmoment":"06/04/2020 17:00:00 GMT"
  }
],
  "label":[
    {"full":"finish roof"},
    {}},
    {}},
    {"colvalue":1},
    {"colvalue":1}
  ]
}}}}},
```

Dependencies

```
"dependencies":[/*This is where the dependencies between tiem
blocks and icons can be defined*/
{
  "el":["grid_33","grid_34"],/*id of the first period/symbol
that needs to be finished before the period/symbol with
the second id starts */
  "type":"fs" /*type is fs "finish-start", currently this is
the only supported type*/
```

```
    },  
    {  
      "el":["grid_34","grid_35"],  
      "type":"fs"  
    }  
  ]}
```

Appendix A icon codes

 1	 2	 3	 4	 5	 6	 7
 8	 9	 10	 11	 12	 13	 14
 15	 16	 17	 18	 47	 48	 49
 83	 84	 85	 86	 87	 88	 89
 90	 91	 50	 51	 52	 53	 54
 55	 56	 57	 58	 59	 60	 61
 62	 63	 64	 65	 66	 67	 68
 69	 70	 71	 72	 73	 74	 75
 76	 77	 78	 79	 80	 81	 82
 19	 20	 21	 22	 23	 24	 25
 26	 27	 28	 29	 30	 31	 32
 33	 34	 35	 36	 37	 38	 39
 40	 41	 42	 43	 44	 45	 46

Appendix B Durations and duration type

Durations are present in two ways in the json.

In the nestTotals properties of activities/rows:

```
{
  "nature": "activity",
  "nestTotals": {
    "duration": 2,
    "last": "03/01/2012 13:00:00 GMT",
    "first": "02/29/2012 13:00:00 GMT"
  },
  .....
}
```

The duration in this context is directly related to the timeline type as defined in the timeline settings:

```
"timeline": {
  ...
  "type": "day",
  ...
}
```

Timeline type;	Duration expressed in:
month	Days
week	Weeks
day	Days
hour	Hours
15minutes	Minutes
10minutes	Minutes
5minutes	Minutes

You will also encounter durations as a property of periods:

```
{
  "nature": "period",
  ...
  "duration": 13,
  ...
}
```

The duration in this context is related to the drag & drop step size in the tool itself. If you are working in a day schedule you can drag & drop your periods and symbols with steps of half a day. So a duration of one is equal to half a day.

Timeline type;	A duration of one is equal to:
month	One day
week	Half a week
day	Half a day
hour	Half an hour
15minutes	7.5 minutes
10minutes	5 minutes
5minutes	2.5 minutes

Appendix C change log

New in version 15 (july 2018)

Colors are not set as integers (pointing to a specific limited set of colors) anymore but as rgb values (for instance: #ff0000 is red). Enabling custom colors for groups and time blocks.

New in version 14

The dependencies section is added to the json (see above)

New in version 12

You can set which day is the first day in the week. The default value is 1 (monday). Setting the first day of the week is done through the property `timeline.firstWeekDay`.

New in version: 8

Added subgroups to Tom's Planner. Each activity has a new property called indentation which indicates the indentation of the rows into subgroups. The indentation property is an integer. The default value is zero.

New in version: 6 (21-5-2012)

1. The timeline types 'month' and 'week' are dismissed.
2. Default zoom is added as a property of settings.

New in version: 5

3. The lastID property has a correct value (can be incorrect in older versions)
4. nestTotals property is added to activities

Version 1 through 4:

In some cases the lastID has an incorrect value. You need to fix this programmatically before you are able to extend the schedule

New in version 4:

1. The property 'settings' is added to the root and has the default value:

```
{plugins:{specialcolumns:{active:true},groupduration:{active:true}}};
```

New in version 3:

The property 'type' is added to each column in the definition.

New in version 2:

1. The properties 'markToday' is added to the `timelinesettings`.

New in version 1:

2. The properties 'hideWeekendDays', 'shadeDays' and 'timeFormat' are added to the timelinesettings.
3. The property 'colorscheme' is added to the print setup.
4. The property 'jsonversion' is added to the root.