Export/Import JSON with Tom's Planner

Introduction	2
Some basic settings	3
Timeline settings	4
The Columns	6
The legend and Logo	7
Print settings	9
The grid itself	10
Appendix A and B period and symbol codes	15
Appendix C background colors codes	15
Appendix D Durations and duration type	16
Appendix E change log	18

Introduction

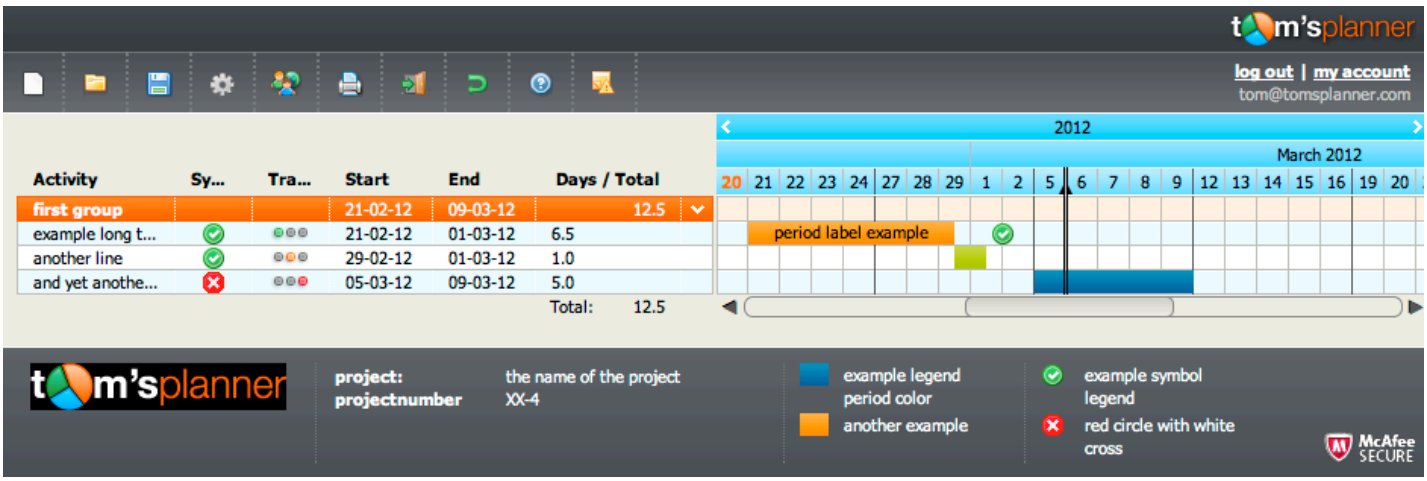
JSON is the format we use to store the schedules in our database. An export to json of your schedule contains all the information there is (including project information, timeline settings, last used printing settings etc.)

In this document we will describe the structure of the most recent version. In Appendix E 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.

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-5.txt.

If you open this schedule it looks like this:

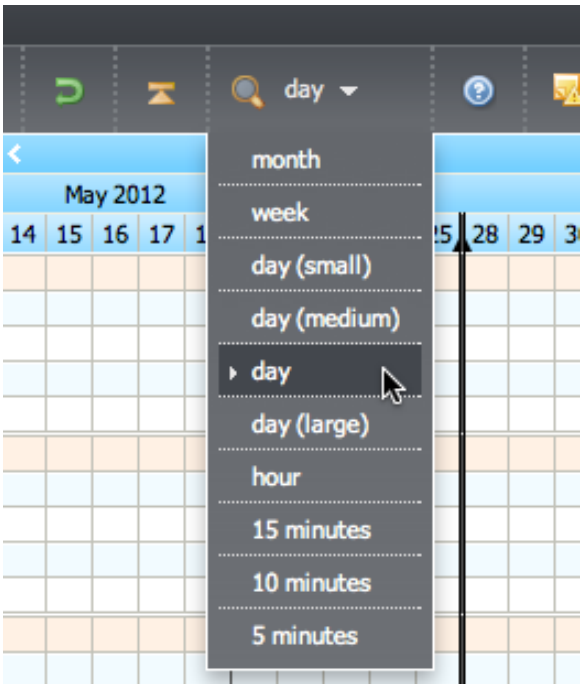


Unfortunately (and a bit embarrassingly) the use of capital letters in the property names is not done consequently throughout the JSON structure. So you will find the property name 'hideWeekendDays' capitalized but for instance the property name 'referencedate' is not capitalized. Apologies for that.

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.

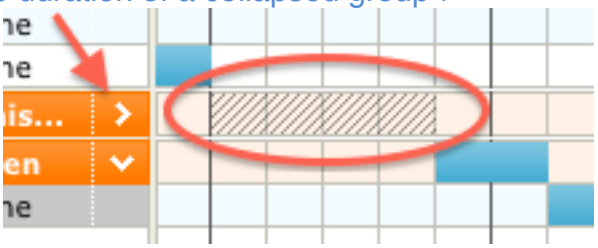
Some basic settings

```
{
    "jsonversion": 6,   /*see Appendix E for the change log*/
    "nature": "tomsplannermodel" /*Needs to be included! Has no specific
                                   meaning*/
     "settings":          { /*these are schedule settings, for now they only contain
                              plugins settings*/
       “defaultzoom”:9, /*the default zoom level of the schedule*/
```

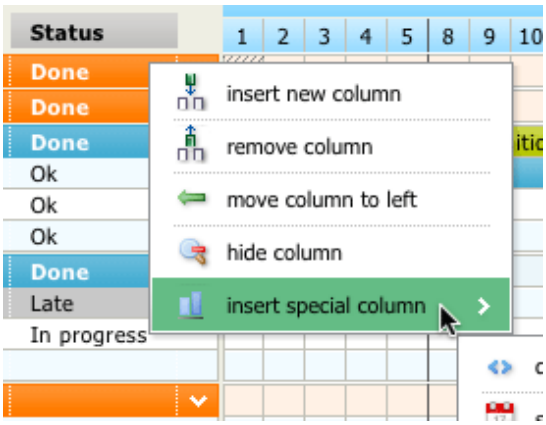


/*the zoom level is stored in the database per schedule/user combination. If no zoom is available (for instance when you open a schedule that has been stored locally as a file or when you import a schedule) this default value is used. The following zoom levels are available: 3=month, 4=week, 6=day(small),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

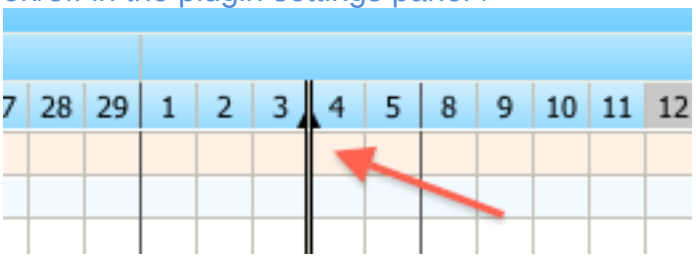
```
    "plugins":{ /*settings for plugins*/

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



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



/*The reference date line refers to the black vertical line in the schedule, which can be turned on/off in the plugin settings panel*/




```
"referencedate": {
        "visible": true,
        "refdate": "03/05/2012 18:07:55 GMT",
},
```

Timeline settings

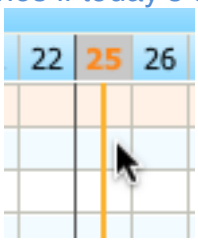
```
"timeline": { /*the settings of the blue timeline*/
        "activeHourFrom": 9, /*start hour working day (integer)*/
        "activeHourTo": 17,  /*end hour working day (integer)*/
        "type": "day",    /*possible values: day, hour, minutes15,
                           minutes10, minutes5*/
        "showYear": true,       /*show line with year numbers in blue timeline*/
        "showMonth": true,      /*show line with month names in blue timeline*/
        "showWeekNr": false, /*show line with week numbers in blue timeline*/
        "showDate": true,        /*show line with dates in blue timeline*/
        "showWeekDay": false, /*show line with names days in blue timeline*/
        "showHour": false, /*hours in blue timeline*/
        "showMinutes": false, /*show minutes in blue timeline*/

          /*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*/
```

weekenddays shaded

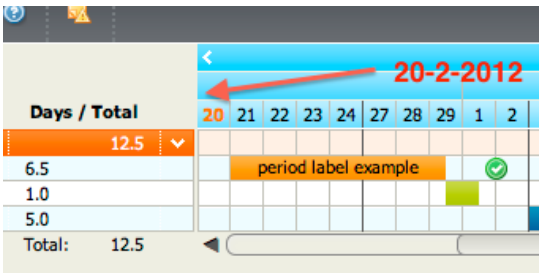
weekenddays hidden

"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*/

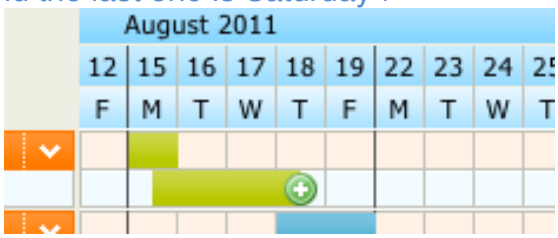
"markToday": true, /*determines if today's date is orange in timeline*/



"leftBorderDate": "02/20/2012 00:00:00 GMT",
/*the position of the left border of the timeline when the schedule is opened in time*/

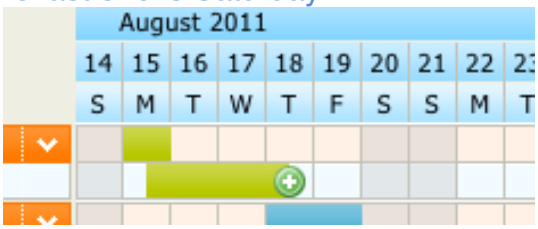


"activeDays": [false, true, true, true, true, true, false],
/*days in the week that need to be visible in the schedule. The first day is a Sunday and the last one is Saturday*/



Sunday and Saturday not active

"shadeDays": [false, 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.*/

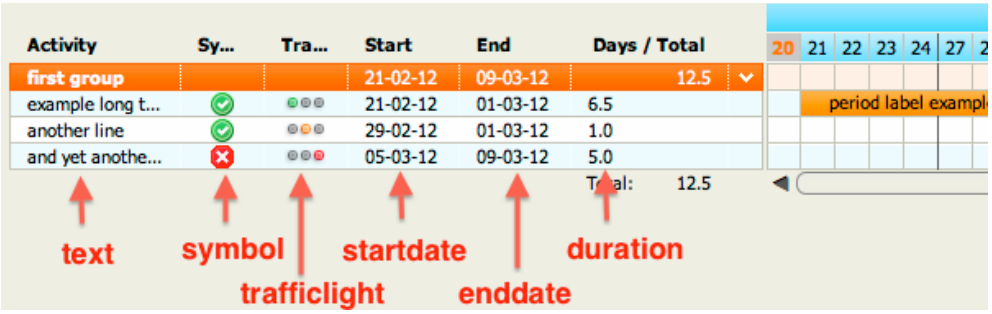


Sunday and Saturday are shaded

},/*end tag timeline settings*/

The Columns

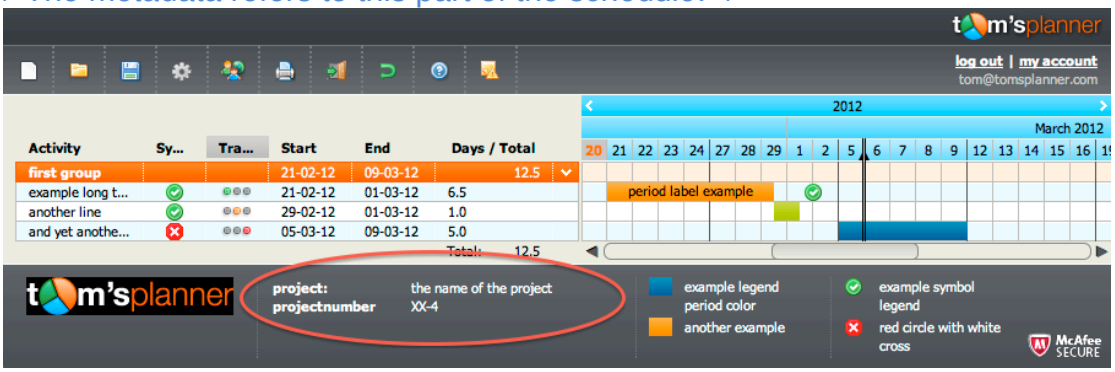
```
"columns": [/*Definition of the column area on the left side of your
            schedule. Each column get's an element in this array and has
            it's own type (see image below) and a colwidth property: */
```




```
{
    "colwidth": 83,/*every column needs a width in pixels*/
    "type": "text",
    "short": "Activity",/*if the column is not wide enough to contain the header
                          this property has the shortened version of the header
                          label*/
    "full": "Activity",/*the complete header of the columns*/
  },{
    "colwidth": 30,
    "type": "symbol",
    "short": "Sy...",
    "full": "Symbol columns",
     "defaultvalue": 0 /*determines the symbol that is added in this column
                          when a new row is inserted*/
  },{
    "colwidth": 30,
    "type": "trafficlight",
    "short": "Tra...",
    "full": "Traffic lights",
    "defaultvalue": 2 /*0=no traffic light, 1=red, 2=orange, 3=green*/
  },{
    "colwidth": 47,
    "type": "startdate"
  },{
    "colwidth": 47,
    "type": "enddate"
  },{
    "colwidth": 75,
    "type": "duration"
  }],
```

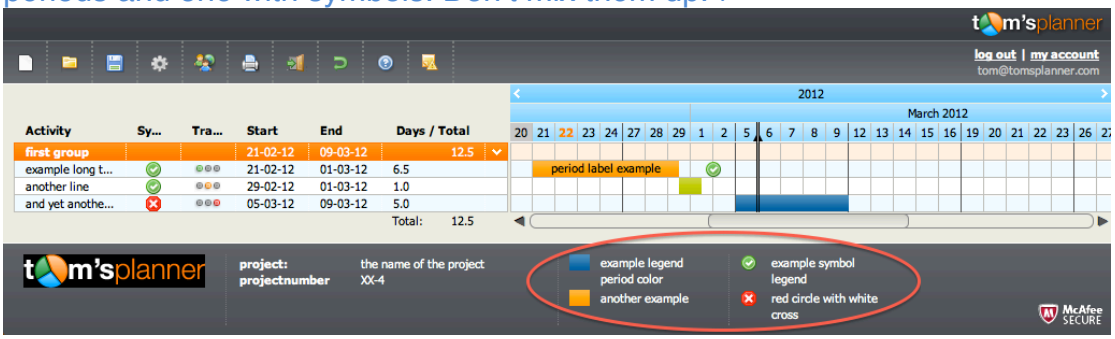
The legend and Logo

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




```
"metadata": [{
        "label": "project:",
        "labelvalue": "the name of the project"
    },
    {
        "label": "projectnumber",
        "labelvalue": "XX-4"
    }],
```

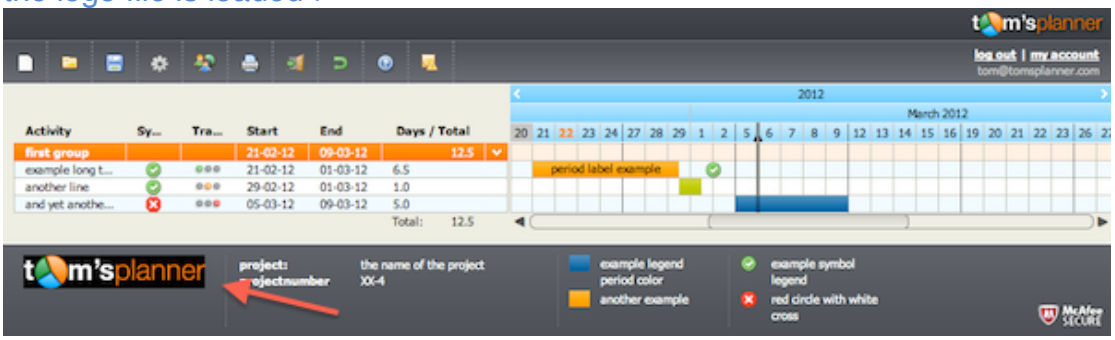
/*The legend is marked in the image below. There is one array with the periods and one with symbols. Don't mix them up.*/




```
    "legenda": {
        "periods": [{
            "label": "example legend period color",
            "type": 23 /* See appendix A for the color codes */
        },
        {
            "label": "another example",
            "type": 19 /* See appendix A for the color codes */
        }],
        "symbols": [{
            "label": "example symbol legend",
            "type": 2 /* See appendix B for the symbol codes*/
```

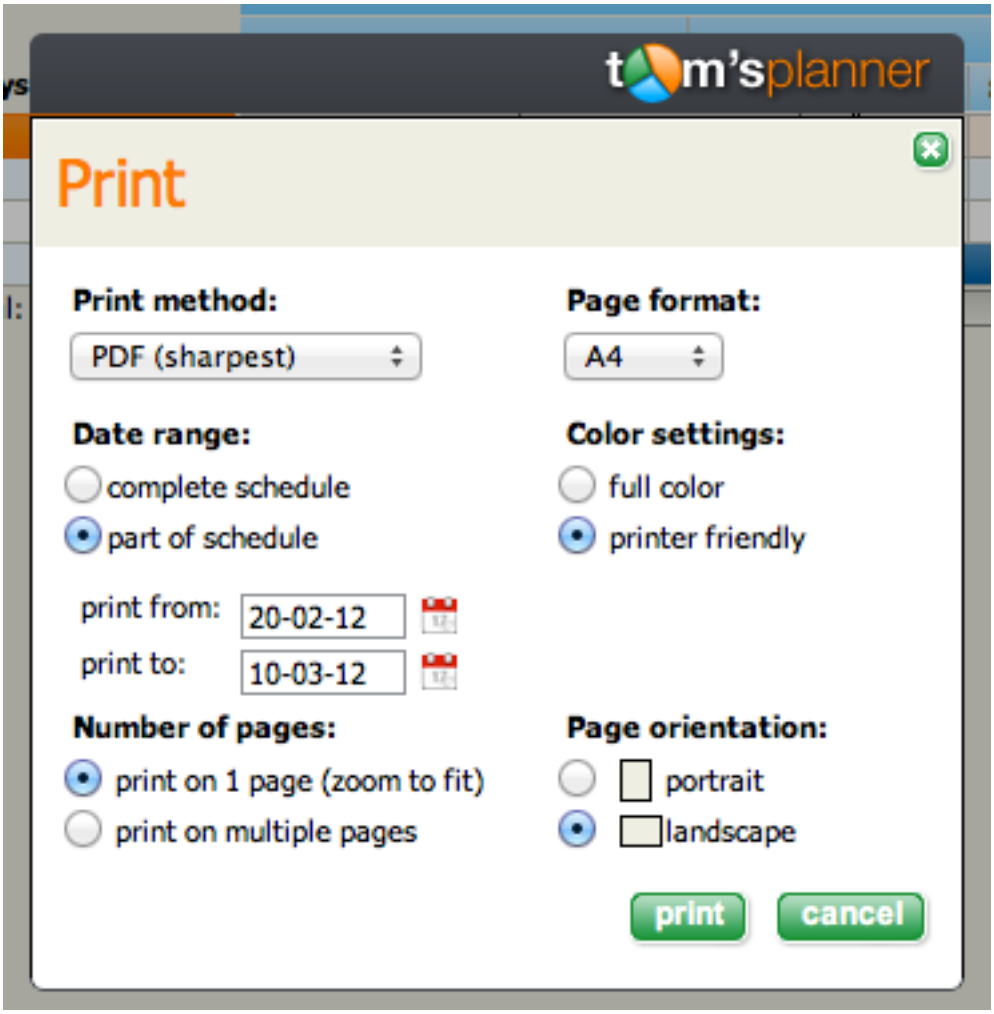
```
        },
        {
            "label": "red circle with white cross",
            "type": 1 /* See appendix B for the symbol codes*/
        }],
        "bolDisplay": true /*determines whether the legend (including the logo
                                and meta data is shown at the bottom of 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 logo's that are externally hosted. The height of the logo needs to be given to avoid that the legend needs to be rerendered when the logo file is loaded*/




```
    "logo": {
        "bolshow": true, /*is the logo visible in the legend or not*/
        "filename": "f0b906b3-da46-47dc-ac33-11b84169c1e2.png",
        "height": 31 /*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. */

```
"visualGrid": {
    "nature": "grid", /*this is like the foundation that the schedule is built
                        upon*/
    "id": "grid_0",/*this grid object has always an id: grid_0*/
    "lastID": 11, /*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": 1, /*this group object has always a type: 1*/
      "kids": [/*this is the array that contains all the groups of the schedule*/
       {
        "nature": "fase",/*top group in the schedule*/
        "id": "grid_2",
        "foldStatus": false,/*determines if a group is collapsed or not*/
        "type": 1, /*determines the background color of the group header.
                     See Appendix C for the color codes*/
        "kids": [{/*top row of the top group in the schedule*/
          "nature": "activity",
          "id": "grid_3",
          "autonumber": "1.0",/*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*/
          "nestTotals": {/*the nestTotals contain information on the objects
                          in it’s kids array. This data is added to the JSON
                          for ease of use when exporting data. When a
                          schedule is opened in Tom’s Planner this data will
                          be ignored and recalculated*/
            "duration": 0,/*integer representing the sum of the total planned
                            durations of all the periods in the kids collection .
                            See appendix d for more information on
                            durations!/
            "last": "",/*  date sting representing the largest value of the
                           endmoment and moment properties of periods and
                           symbols in the kids collection of the activity. If no
```

```
                    periods or symbols are found this property is equal
                    to an empty string */
      "first": ""/* same as the 'last' property but then the smallest
                  value*/

    },
    "kids": [],
    "label": [/*the label array contains the column data of the specific
                row/activity*/
     {/*the first column is of type 'text'*/
      "short": "first group",/*label trimmed to fit the width of
                              column*/
      "full": "first group"/*complete label*/
    },
    {/*the second column is of type 'symbol. See appendix B for the
      symbol codes*/
      "colvalue": 0
    },
    {/*the third column is of type 'trafficlight, 0=no traffic light, 1=red,
      2=orange, 3=green */
      "colvalue": 0
    },
     {/*the fourth column is of type 'startdate*, object has no
     properties
     /},
    {/*the fifth column is of type 'enddate*, object has no properties /},
   {/*the sixth column is of type 'duration*, object has no properties/}]
},
{
    "nature": "activity",
    "id": "grid_4",
    "autonumber": "1.1",
    "nestTotals": {
      "duration": 13,
      "last": "03/01/2012 17:00:00 GMT",
      "first": "02/21/2012 09:00:00 GMT"
    },
    "kids": [/*the kids array of an activity can contain periods and
              symbols*/
     {
      "nature": "symbol",
      "kids": [],/*the kids collection of a symbol is always empty*/
      "id": "grid_5",
      "type": 2, /* See appendix B for the symbol codes */
      "moment": "03/01/2012 17:00:00 GMT"
    },
    {
      "nature": "period",
```

```
                "kids": [],
                "id": "grid_6",
                "type": 19, See appendix A for the color codes
                "duration": 13,
                "durationtype": "day",/* see appendix D of this document */
                "startmoment": "02/21/2012 09:00:00 GMT",
                "finishmoment": "02/29/2012 13:00:00 GMT",/*the duration
overrides the finishmoment. Finishmoment is result of startmoment, duration
and durationtype*/
                "label": "period label example" /*optional property with the label
        of the period (being the text in the colored bar itself)*/
            }],
            "label": [/*the label array contains the column data of the specific
                        row/activity. See earlier in this document*/

             {
                "short": "example long t...",
                "full": "example long text"
            },
            {
                "colvalue": 2
            },
            {
                "colvalue": 3
            },
            {},
            {},
            {}]
        },

/*The following is a repetition of the pervious parts of the json so there's
nothing new to tell. For completeness we have added it to the document*/
        {
            "nature": "activity",
            "id": "grid_7",
            "autonumber": "1.2",
            "nestTotals": {
                "duration": 2,
                "last": "03/01/2012 13:00:00 GMT",
                "first": "02/29/2012 13:00:00 GMT"
            },
            "kids": [{
                "nature": "period",
                "kids": [],
                "id": "grid_8",
                "type": 15,
                "duration": 2,
                "durationtype": "day",
```

```
            "startmoment": "02/29/2012 13:00:00 GMT",
            "finishmoment": "03/01/2012 13:00:00 GMT"
        }],
        "label": [{
            "short": "another line",
            "full": "another line"
        },
        {
            "colvalue": 2
        },
        {
            "colvalue": 2
        },
        {},
        {},
        {}]
    },
    {
        "nature": "activity",
        "id": "grid_9",
        "autonumber": "1.3",
        "nestTotals": {
            "duration": 10,
            "last": "03/09/2012 17:00:00 GMT",
            "first": "03/05/2012 09:00:00 GMT"
        },
        "kids": [{
            "nature": "period",
            "kids": [],
            "id": "grid_10",
            "type": 23,
            "duration": 2,
            "durationtype": "week",
            "startmoment": "03/05/2012 09:00:00 GMT",
            "finishmoment": "03/09/2012 17:00:00 GMT"
        }],
        "label": [{
            "short": "and yet anothe...",
            "full": "and yet another one "
        },
        {
            "colvalue": 1
        },
        {
            "colvalue": 1
        },
        {},
        {},
```

```
                {}]
            }]
        }]
    }]
  }
}
```

Appendix A color codes

22	28	23	25	29
15	21	16	18	31
8	14	9	11	34
27	24	41	26	12
20	17	42	44	45
13	10	43	19	46

Appendix B 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 C group color codes

1	2	3	4	5	6

Appendix D 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 E change log

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

1. The timeline types ‘month’ and ‘week’ are dismissed.
2. Defaultzoom 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.