12,614,663 members (34,413 online)

Sign in

CODE PROJECT For those who code

home articles quick answers discussions features community help

Search for articles, questions, tips

Articles » Platforms, Frameworks & Libraries » Libraries » Code Libraries

Creating your own online Gantt application with dhtmlxGantt

Stas Wolski, 15 Jun 2015 GPL3

★★★★★ 5.00 (8 votes)

Rate this:

In this article I want to tell you how to build your own online Gantt chart application. I'll use dhtmlxGantt for that purpose. I will show you how you can create a basic Gantt chart and what extra features you can add. You should have a local web-server (Apache or nginx, for example) with PHP and database (MySQL in my case) running to make everything work.

What for?

To make a long story short, Gantt charts are pretty useful, if you want to manage a complex project. The main aim is to visualize the working process. But if the existing functionality of Gantt applications is not enough, you may want to create your own one. Well, this article can be pretty handy then.

It's quite easy to initialize a basic dhtmlxGantt chart and add some handy features to it. Definitely easier than to pronounce its name. JavaScript API allows you to customize any chart element you want. There is also a possibility to export your data to different file formats such as PDF or PNG. To see all the features available, check the project's home page. A bunch of demos are available there as well.

Basic chart initialization

Let's now create our application. First, we should create some data to visualize. Then we'll initialize the Gantt chart with this data.

First of all, you need to download the library package, of course. Then you should extract the *codebase* folder to the root directory of your local web server. And now, it's time to code...

Phase one. Preparing the data

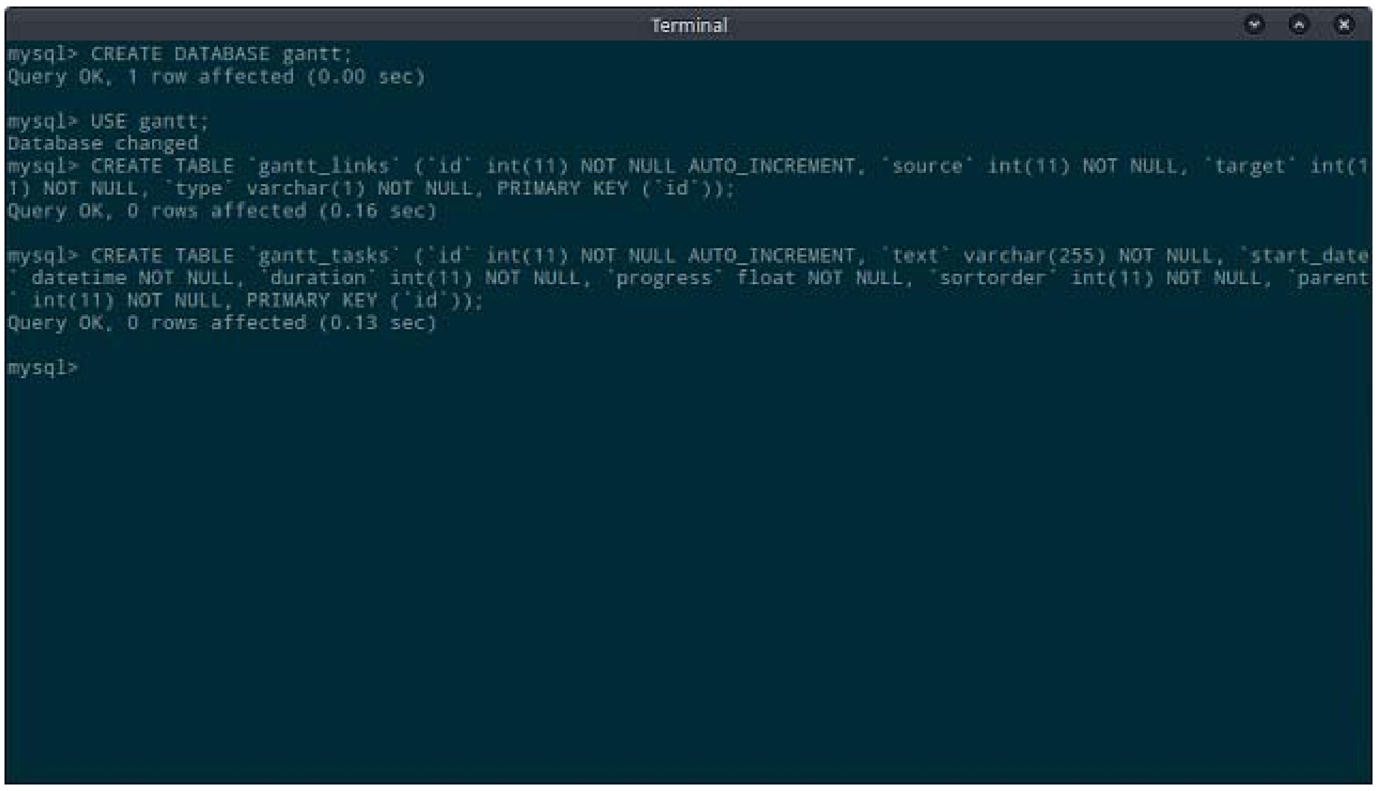
All the boring work first. We should create some data to work with. There is a possibility to use an inline data source. XML and JSON formats are available. But it's useful for demonstration purposes only. To see how the actual application works we'll use a database.

We should create a new database and then add two tables to it:

Hide Copy Code

```
CREATE DATABASE gantt;
USE gantt;
CREATE TABLE `gantt_links` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `source` int(11) NOT NULL,
  `target` int(11) NOT NULL,
  `type` varchar(1) NOT NULL,
  PRIMARY KEY (`id`)
);
CREATE TABLE `gantt_tasks` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `text` varchar(255) NOT NULL,
  `start_date` datetime NOT NULL,
  `duration` int(11) NOT NULL,
  `progress` float NOT NULL,
  `sortorder` int(11) NOT NULL,
  `parent` int(11) NOT NULL,
  PRIMARY KEY (`id`)
);
```

Seems that everything went well:



```
mysql> CREATE DATABASE gantt;
Query OK, 1 row affected (0.00 sec)

mysql> USE gantt;
Database changed
mysql> CREATE TABLE `gantt_links` (`id` int(11) NOT NULL AUTO_INCREMENT, `source` int(11) NOT NULL, `target` int(1
1) NOT NULL, `type` varchar(1) NOT NULL, PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.16 sec)

mysql> CREATE TABLE `gantt_tasks` (`id` int(11) NOT NULL AUTO_INCREMENT, `text` varchar(255) NOT NULL, `start_date
` datetime NOT NULL, `duration` int(11) NOT NULL, `progress` float NOT NULL, `sortorder` int(11) NOT NULL, `parent
` int(11) NOT NULL, PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.13 sec)

mysql>
```

Now we need to use some PHP functionality to handle our new database. We'll use the dhtmlxConnector library for that purpose. Create the `data.php` file and add the proper code into it:

Hide Copy Code

```
<?php

include ('codebase/connector/gantt_connector.php');

$res=mysql_connect("localhost","root","");
mysql_select_db("gantt");

$gantt = new JSONGanttConnector($res);
$gantt->render_links("gantt_links","id","source,target,type");
$gantt->render_table(
    "gantt_tasks",
    "id",
    "start_date,duration,text,progress,sortorder,parent"
);
?>
```

As you can see, it selects our previously created `gantt` database and uses two tables from it.

Data handling is done. Let's move to the `index.html` file and make our chart work.

Phase two. Chart initialization

To make a basic Gantt chart work we need to include the `dhtmlxgantt.js` and `dhtmlxgantt.css` files into our project. You can find them inside the *codebase* folder. I've added some CSS code to make sure everything will work fine in the full-screen mode. Check the `<style></style>` tags content to see what I've done here. All the chart code is placed within the `<body></body>` tags. The `<div></div>` pair of tags contains the chart. I've set its `height` and `width` properties to 100% to make sure that chart will take all available space. Don't forget to set the `id` attribute. We'll use its value later to initialize our chart. Here's my `index.html` file:

Hide Copy Code

```
<!DOCTYPE html>
<html>
<head>
    <title>My Own Gantt Chart</title>
    <script src="codebase/dhtmlxgantt.js"></script>
    <link href="codebase/dhtmlxgantt.css" rel="stylesheet">
    <style type="text/css">
        html, body{ height:100%; padding:0px; margin:0px; overflow: hidden;}
    </style>
</head>
<body>
    <div id="my_gantt" style="width:100%; height:100%;"></div>
    <script>
    //All the Gantt chart code goes here
    </script>
</body>
</html>
```

Now it's time to breathe some life into our project. All the code below should be added inside the `<script></script>` tags of our `index.html` file.

We need to make the format of output data compatible with the format of dhtmlxGantt:

Hide Copy Code

```
gantt.config.xml_date = "%Y-%m-%d %H:%i";
```

Now we can initialize our chart:

Hide Copy Code

```
gantt.init("my_gantt");
```

Load the previously created data:

Hide Copy Code

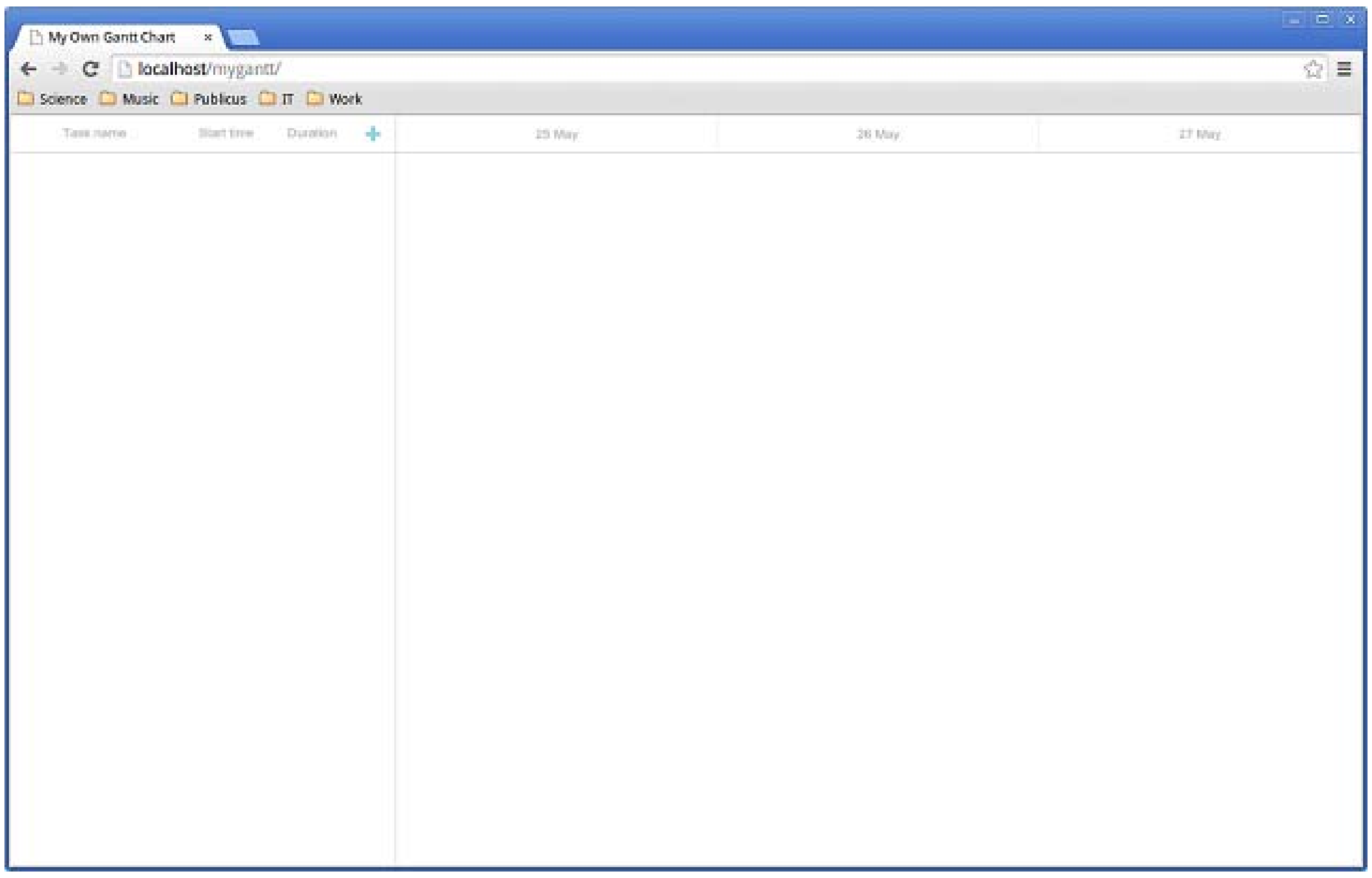
```
gantt.load('data.php');
```

And the final step is to provide the ability to update our data. The dhtmlxDataProcessor helper library is what we need here and now:

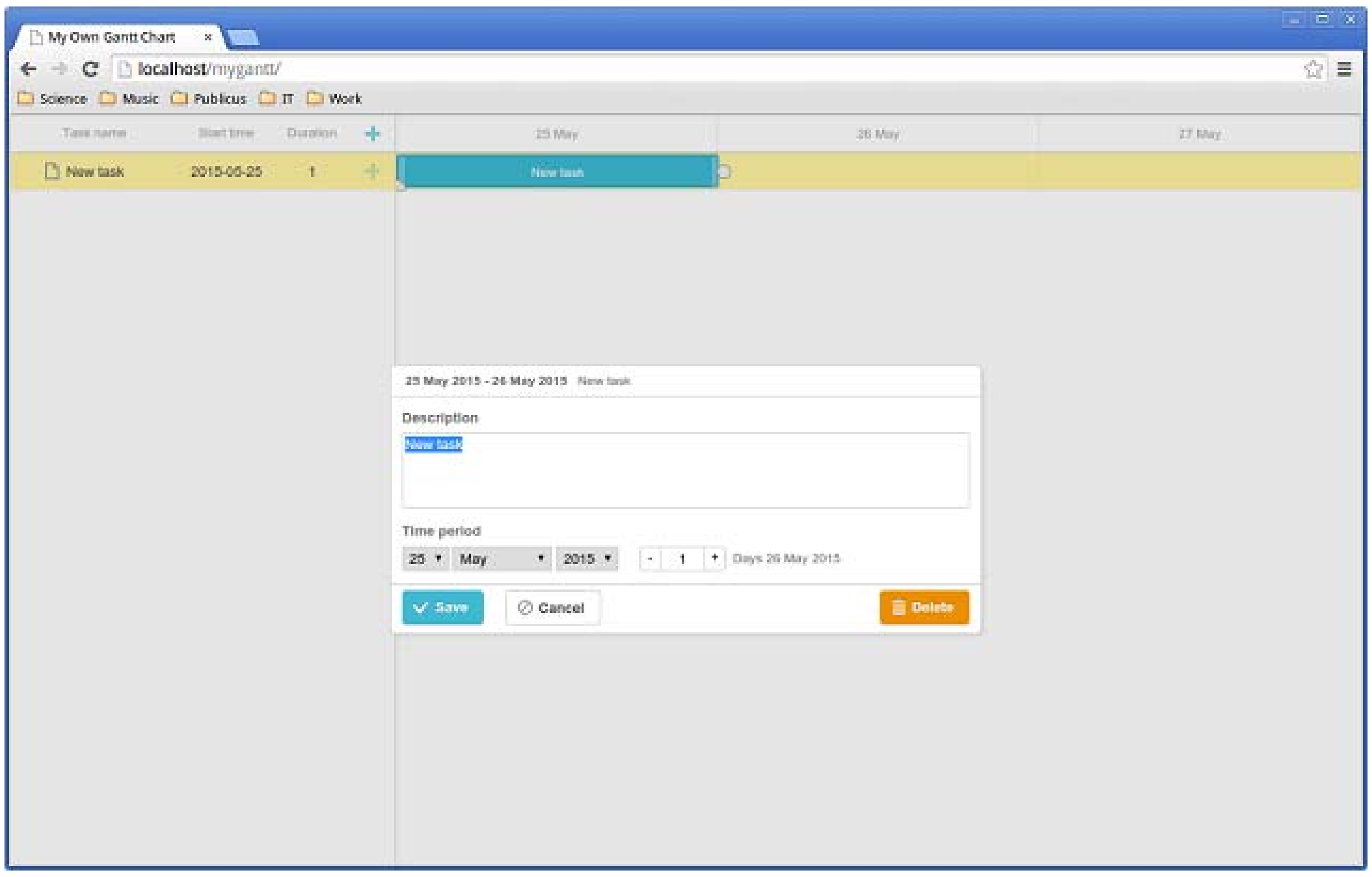
Hide Copy Code

```
var dp=new dataProcessor("data.php");
dp.init(gantt);
```

Well, that's all. These are the five steps that we need to initialize the chart. Let's check the result:



"I can't see any charts!" you may say. Don't worry. As we've created an empty database, there's no data by default. But we can simply create some tasks using the "+" button at the top:



Task name	Start time	Duration
New task	2015-05-25	1

25 May 2015 - 26 May 2015 New task

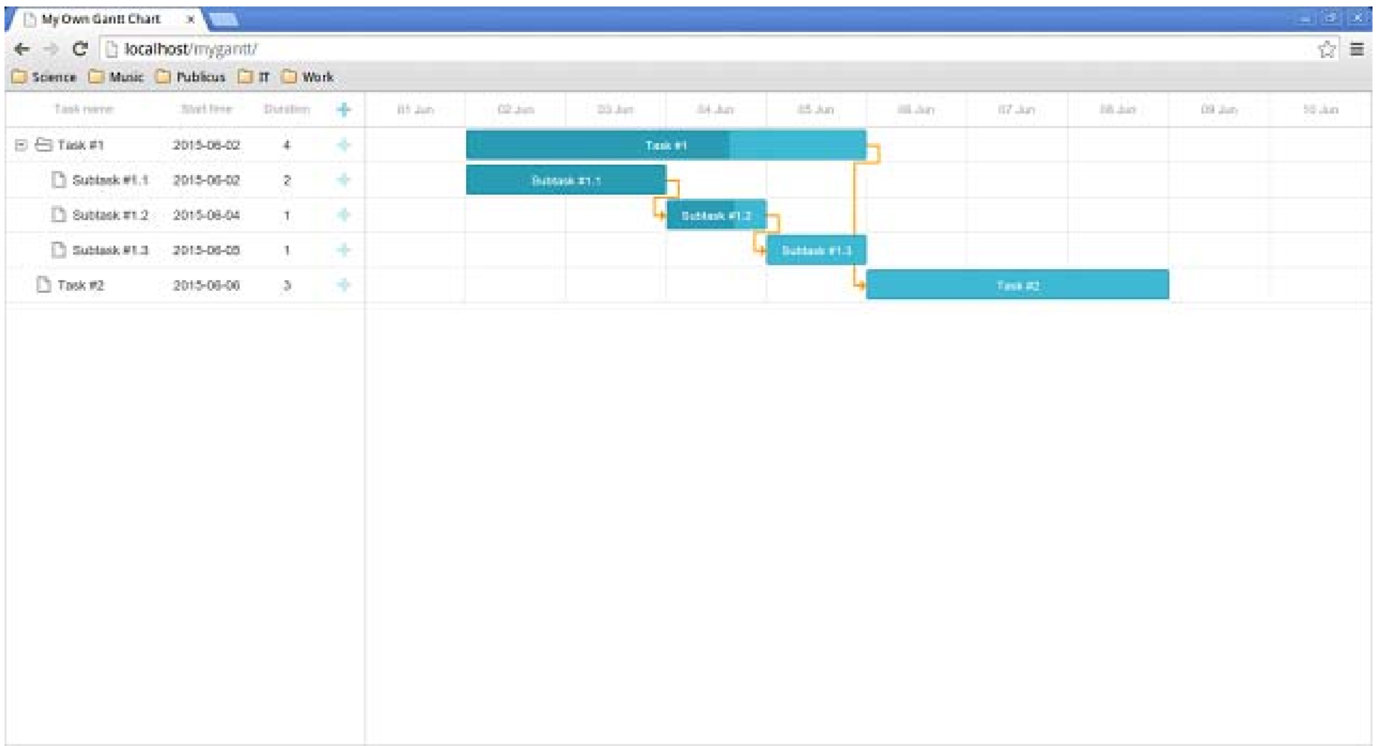
Description

New task

Time period

25 May 2015 - 1 + Days 26 May 2015

The triangle in the bottom left corner of a task will help you to set the current task's status. By dragging the circle at the task's edge, it's possible to set a dependency with another task.
There is another "+" button on the right side of the task description as you can see. It allows you to create a subtask.



Task name	Start time	Duration	+
Task #1	2015-06-02	4	+
Subtask #1.1	2015-06-02	2	+
Subtask #1.2	2015-06-04	1	+
Subtask #1.3	2015-06-05	1	+
Task #2	2015-06-06	3	+

So, what do we have by default? You can edit your tasks by double-clicking them, drag tasks horizontally to change the start date, resize tasks to change their duration. It's easy to define dependencies between the existing tasks. Besides, we've created this chart using less than 50 lines of code.

Adding extra features

There is a guides page that will help you to add new features to your Gantt chart. Let's try a couple of them to see how it all works.

Exporting

With dhtmlxGantt you can export your chart to a file. It may be useful if you want to show someone your project's current progress, for example. There are PDF, PNG, Excel and iCal output formats available. To make the export feature available you should include `api.js` to your file:

Hide Copy Code

```
<script src="http://export.dhtmlx.com/gantt/api.js"></script>
```

Now you're free to use export functions. As an example, I've created a simple drop-down list and a JavaScript handler for it. Add this code before the `<div id="myGantt"></div>` tag:

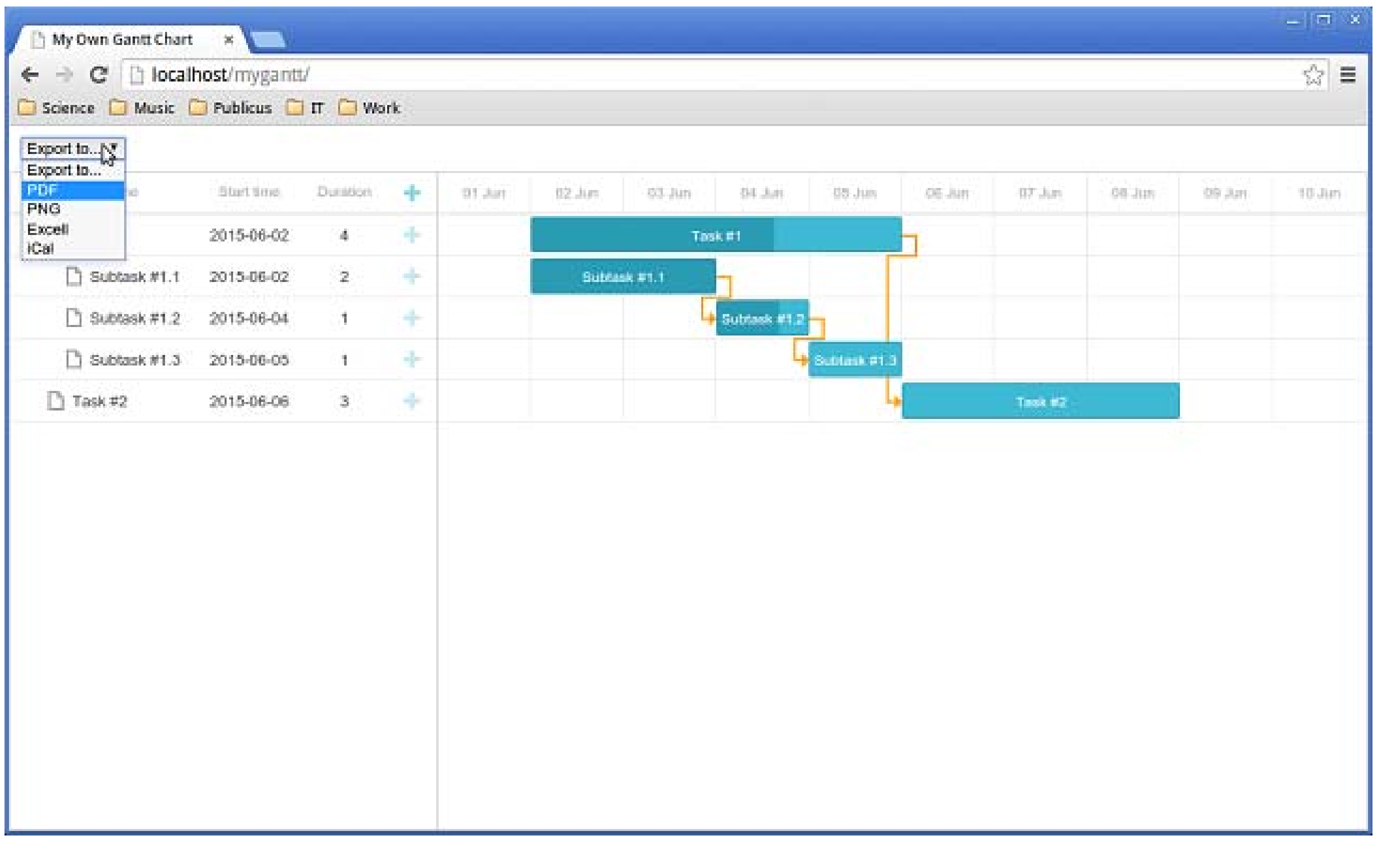
Hide Copy Code

```
<select id="export" style='margin:10px;' onchange='exportTo()'>
  <option>Export to...</option>
```

As you can see, it allows you to choose one of the available file formats. Now, let's handle a user's choice. Here's my even-baby-will-understand handler. Insert it within the `<script></script>` tags.

Hide Copy Code

```
var exportTo = function() {
    var output = document.getElementById("export").value;
    switch (output) {
        case "pdf": gantt.exportToPDF(); break;
        case "png": gantt.exportToPNG(); break;
        case "excel": gantt.exportToExcel(); break;
        case "ical": gantt.exportToICal(); break;
        default: alert("Wrong file format");
        }
};
```



	Start time	Duration	+
	2015-06-02	4	+
Subtask #1.1	2015-06-02	2	+
Subtask #1.2	2015-06-04	1	+
Subtask #1.3	2015-06-05	1	+
Task #2	2015-06-06	3	+

As you can see, all you need to export a Gantt chart to a file is a single method.

Important tasks

You can highlight partially completed tasks. If there are a lot of them, you should work harder. But not on your weekends, off course! So, let’s highlight them too. We’ll use the `gantt.templates` object for this task. It allows us to configure the templates for dates and labels used in the Gantt chart.

Let’s start with the styling work. Add this code between the `<style></style>` tags:

Hide Copy Code

```
.important{
    border:2px solid red;
    color:red;
    background:red;
}
.important .gantt_task_progress{
    background: #ff5956;
}
.weekend{
    background: #f4f7f4 !important;
}
```

This CSS code defines the style to highlight critical tasks and weekends. Now we need to change the chart behaviour. You should add this code before the `gantt.init()` line. First, the weekends:

Hide Copy Code

```
gantt.templates.scale_cell_class = function(date){
    if(date.getDay()==0||date.getDay()==6){
        return "weekend";
    }
};
gantt.templates.task_cell_class = function(item,date){
    if(date.getDay()==0||date.getDay()==6){
        return "weekend" ;
    }
};
```

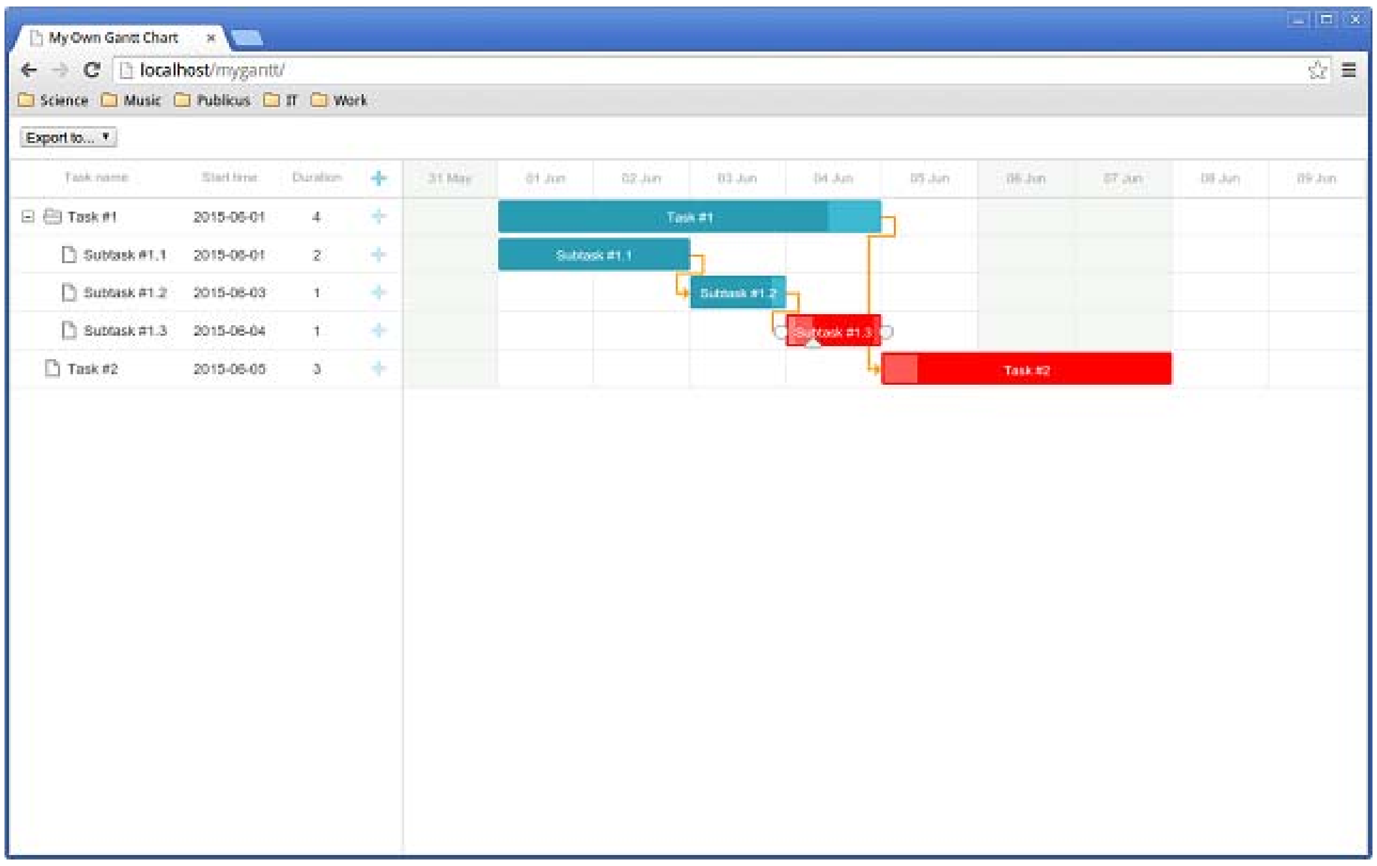
We used the `scale_cell_class` template to highlight the timescale area and the `task_cell_class` for the chart area.

Now, the tasks:

Hide Copy Code

```
gantt.templates.task_class=function(start,end,task){
    if(task.progress > 0.5){
         return "";
    }else{
        return "important";
    }
};
```

Let’s see how it looks like now.



Task name	Start time	Duration
Task #1	2015-06-01	4
Subtask #1.1	2015-06-01	2
Subtask #1.2	2015-06-03	1
Subtask #1.3	2015-06-04	1
Task #2	2015-06-05	3

As you can see, all critical tasks and weekends are highlighted now.

Single line features

Let’s now add some more features. We’ll use the `gantt.config` object.
This code:

Hide Copy Code

```
gantt.config.order_branch = true;
```

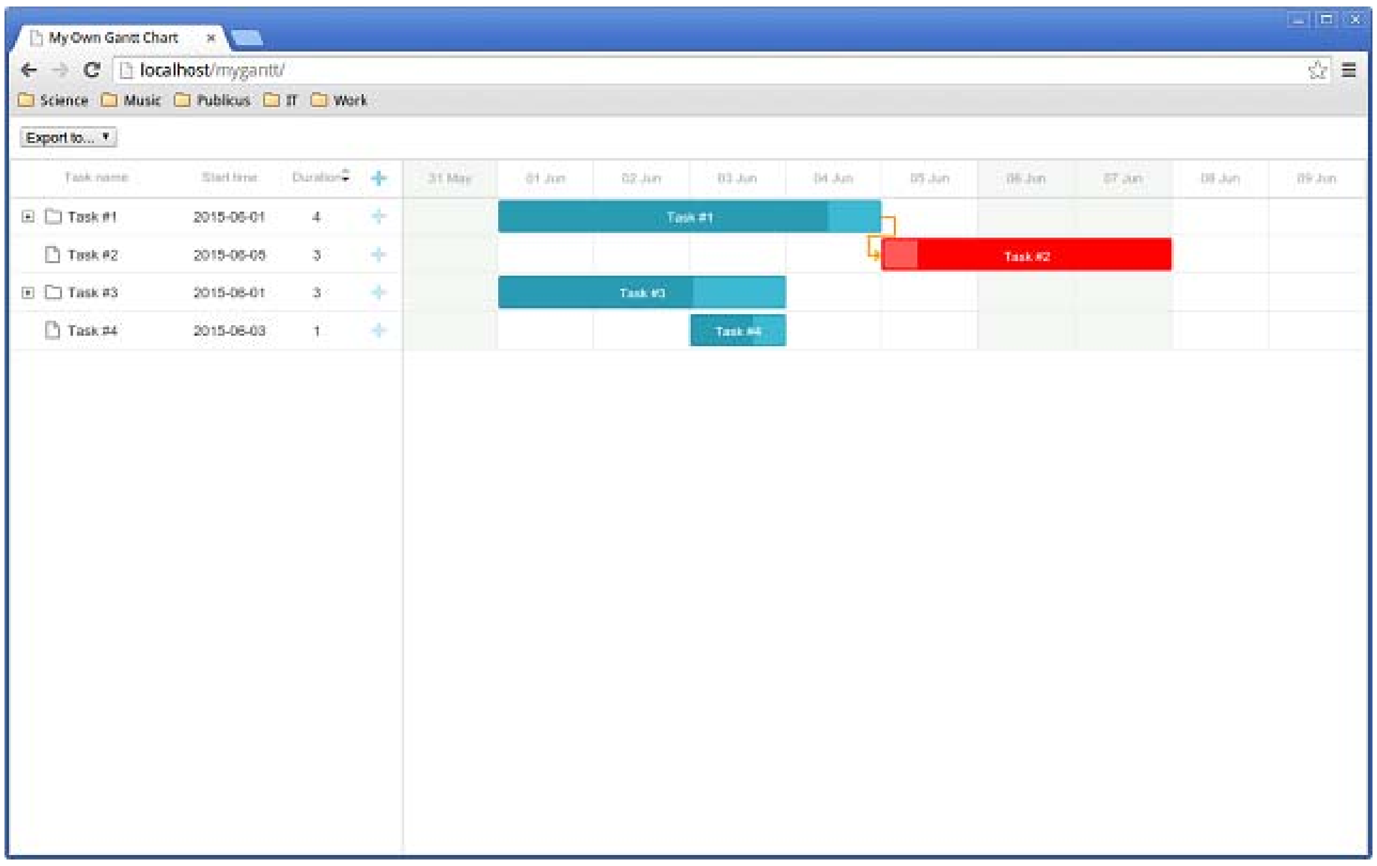
Enables drag-n-drop reordering. So, you can change the order for a single subtask in case of a force majeure of any kind.

To enable sorting you should set the `sort` property to `true`:

Hide Copy Code

```
gantt.config.sort = true;
```

Which will allow you to define the most exhausting tasks:

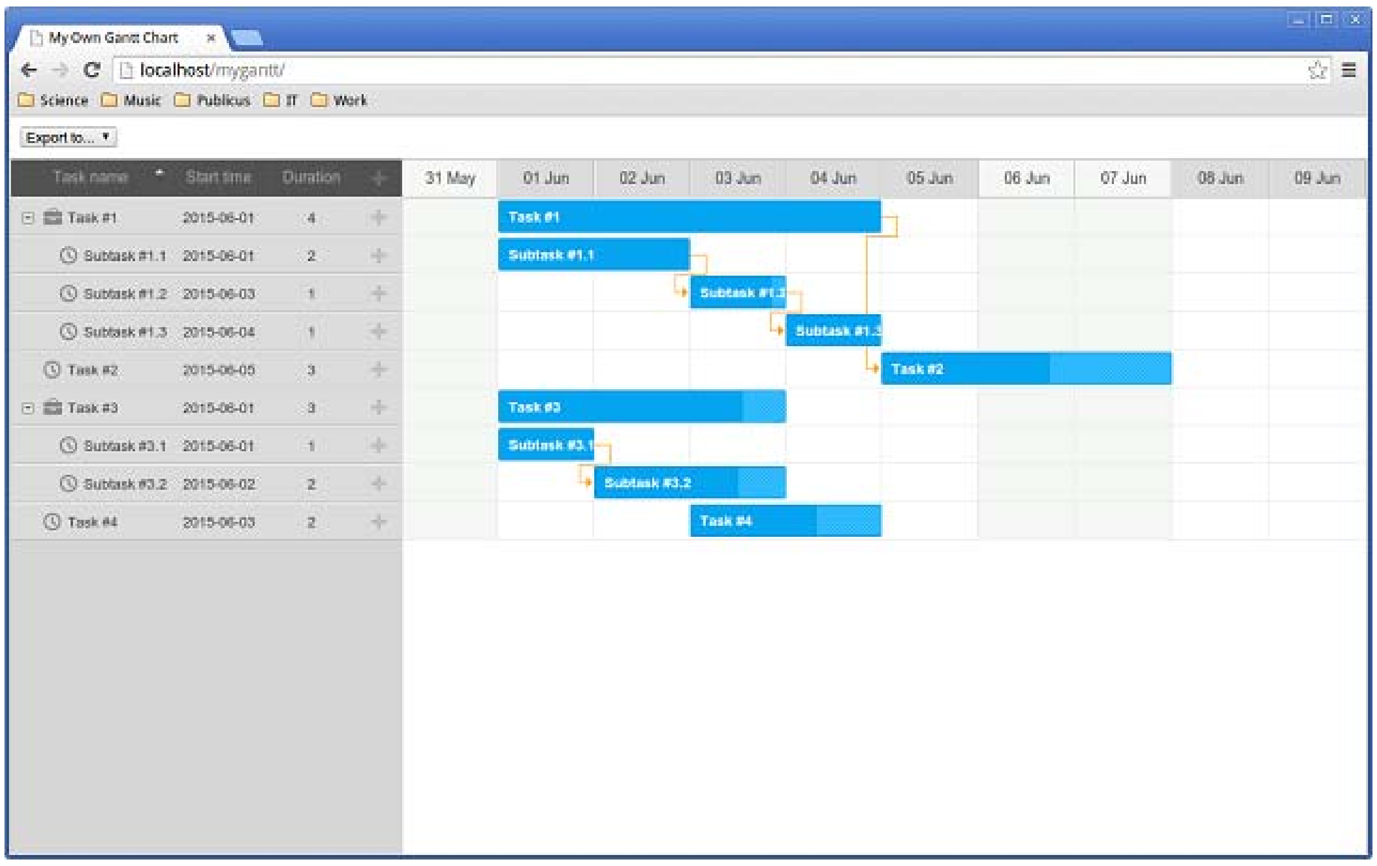


Task name	Start time	Duration
Task #1	2015-06-01	4
Task #2	2015-06-05	3
Task #3	2015-06-01	3
Task #4	2015-06-03	1

And finally, my favorite one. Changing the skin. There are four of them available: Terrace, Skyblue, Meadow and Broadway. To change the skin, you need to include a proper .css file to your project:

```
<link rel="stylesheet" href="codebase/skins/dhtmlxgantt_broadway.css" type="text/css">
```

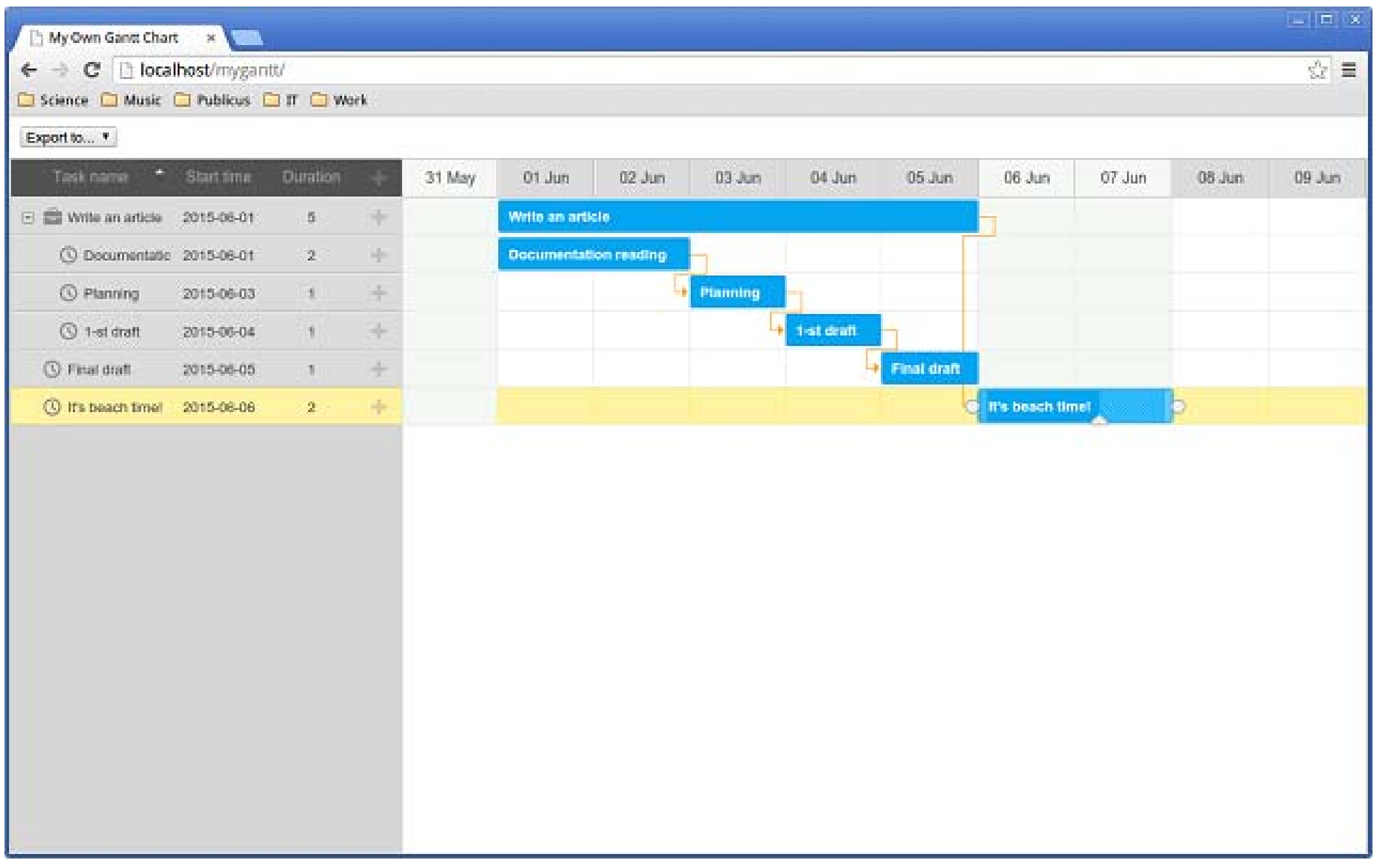
Here's how it looks like now:



Task name	Start time	Duration
Task #1	2015-06-01	4
Subtask #1.1	2015-06-01	2
Subtask #1.2	2015-06-03	1
Subtask #1.3	2015-06-04	1
Task #2	2015-06-05	3
Task #3	2015-06-01	3
Subtask #3.1	2015-06-01	1
Subtask #3.2	2015-06-02	2
Task #4	2015-06-03	2

In total

dhtmlxGantt is an easy coding and good looking application. Default chart is pretty simple and low functional, but there is a lot of handy features that can turn it into an advanced one. Guides page will help you with that. I've created a simple Gantt chart app and it's already allows me to make this:



Task name	Start time	Duration
Write an article	2015-06-01	5
Documentatic	2015-06-01	2
Planning	2015-06-03	1
1-st draft	2015-06-04	1
Final draft	2015-06-05	1
It's beach time!	2015-06-06	2

So, I hope that this tutorial will help those developers who start exploring dhtmlxGantt and need simple explanation on how to implement this project management library to their applications. I've tried to describe it as easy as possible not diving deep into details. You're welcome with your questions. Thanks for reading.

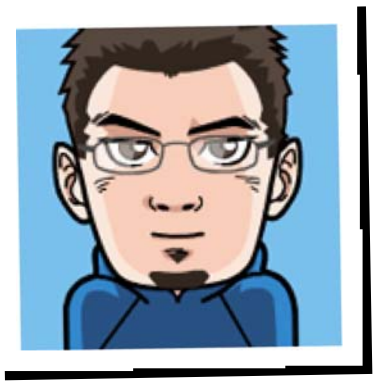
License

This article, along with any associated source code and files, is licensed under The GNU General Public License (GPLv3)

Share

EMAIL TWITTER

About the Author



Stas Wolski

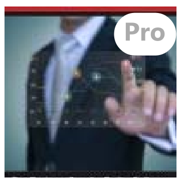
Software Developer (Senior)

Russian Federation 🇷🇺

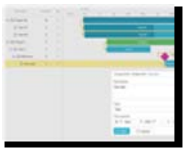
Building web apps since 2001. Rich experience in HTML frontends for different platforms.

You may also be interested in...

Keeping Up With PHP



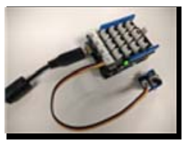
The Business Case for Earlier Software Defect Detection and Compliance



Gantt Chart for ASP.NET MVC with dhtmlxGantt

Mobile Messaging with Twilio

Creating Gantt Chart App with dhtmlxGantt and MeteorJS



Arduino 101 Bluetooth Low Energy

Comments and Discussions