

- 1.
- 2.
- 3.
- 4.
- 5./ API
- 6.
- 7. Just code it!
 - a.
 - b.
 - c.
 - d.
 - e. Felix
 - f.
- 8.

Bean Shell

1.

Tait Brown MITJquery Gantt Chart Jasper Report

2.

Userview JQuery

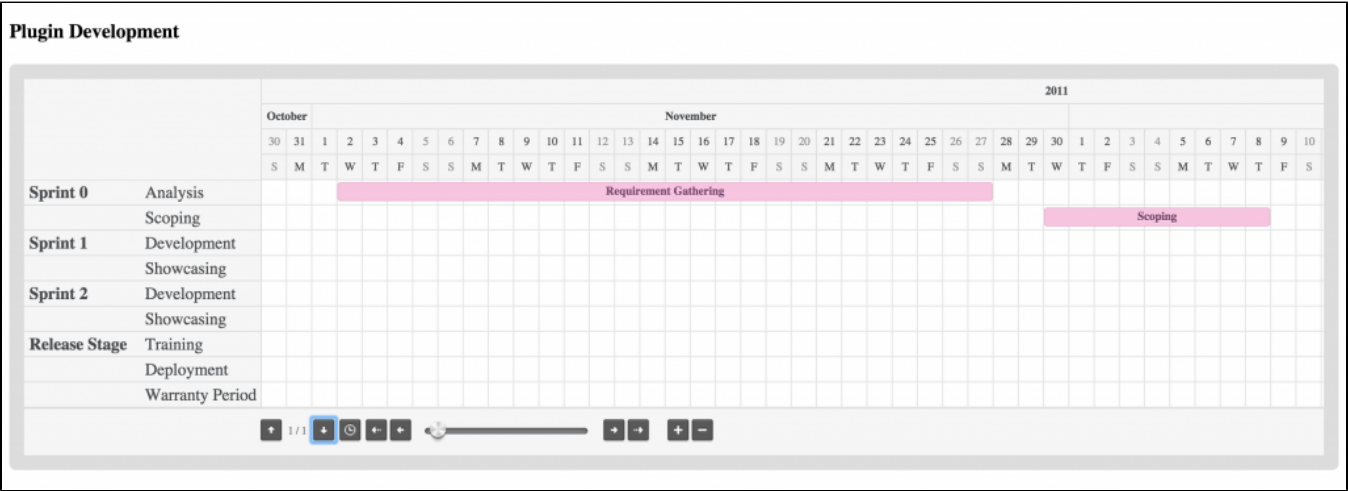
3.

Userview

- 1.
- 2.
- 3.

4.

HTMLJogetHTML static.zipHTML



5./ API

FreeMaker

6.

```
- Home
- joget
- plugins
- jw-community
-5.0.0
```

"plugins" "jw-community" Joget Workflow

"plugins" maven

```
cd joget/plugins/
~/joget/jw-community/5.0.0/wflow-plugin-archetype/create-plugin.sh org.joget.tutorial gantt_chart_menu 5.0.0
```

shellmaven

```
Define value for property 'version': 1.0-SNAPSHOT: : 5.0.0
[INFO] Using property: package = org.joget.tutorial
Confirm properties configuration:
groupId: org.joget.tutorial
artifactId: gantt_chart_menu
version: 5.0.0
package: org.joget.tutorial
Y: : y
```

"BUILD SUCCESS" "plugins" "gantt_chart_menu"

IDE maven [NetBeans](#)

7. Just code it!

a.

"org.joget.tutorial" "GanttChartMenu" **org.joget.apps.userview.model.UserviewMenu**

b.

AppPluginUtil.getMessage18nMESSAGE_PATH

Implementation of all basic abstract methods

```
package org.joget.tutorial;

import org.joget.apps.app.service.AppPluginUtil;
import org.joget.apps.userview.model.UserviewMenu;

public class GanttChartMenu extends UserviewMenu {

    private final static String MESSAGE_PATH = "messages/GanttChartMenu";

    public String getName() {
        return "Gantt Chart Menu";
    }
    public String getVersion() {
        return "5.0.0";
    }

    public String getClassName() {
        return getClass().getName();
    }

    public String getLabel() {
        //support i18n
        return AppPluginUtil.getMessage("org.joget.tutorial.GanttChartMenu.pluginLabel", getClassName(),
MESSAGE_PATH);
    }

    public String getDescription() {
        //support i18n
        return AppPluginUtil.getMessage("org.joget.tutorial.GanttChartMenu.pluginDesc", getClassName(),
MESSAGE_PATH);
    }

    public String getPropertyOptions() {
        return AppUtil.readPluginResource(getClassName(), "/properties/ganttChartMenu.json", null, true,
MESSAGE_PATH);
    }

    @Override
    public String getCategory() {
        return "Marketplace";
    }
    @Override
    public String getIcon() {
        //sorry, i am reuse the icon of other plugin here
        return "/plugin/org.joget.apps.userview.lib.HtmlPage/images/grid_icon.gif";
    }

    @Override
    public boolean isHomePageSupported() {
        return true; // Can use as first page of the userview
    }

    @Override
    public String getDecoratedMenu() {
        return null; // using default
    }

    @Override
    public String getRenderPage() {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

UigetPropertyOptions "/properties/ganttChartMenu.json""gantt_chart_menu / src / main""resources / properties""properties""ganttChartMenu.json"

"@@ message.key @@"i18n

```

[ {
  title : '@@userview.ganttchart.config@@',
  properties : [ {
    name : 'id',
    label : 'Id',
    type : 'hidden'
  },
  {
    name : 'customId',
    label : '@@userview.ganttchart.customId@@',
    type : 'textfield',
    regex_validation : '^[a-zA-Z0-9_]+$',
    validation_message : '@@userview.ganttchart.invalidId@@'
  },
  {
    name : 'label',
    label : '@@userview.ganttchart.label@@',
    type : 'textfield',
    required : 'True'
  },
  {
    name : 'title',
    label : '@@userview.ganttchart.title@@',
    type : 'textfield'
  },
  {
    name : 'binder',
    label : '@@userview.ganttchart.binder@@',
    type : 'elementselect',
    required : 'True',
    options_ajax : '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.datalist.model.
DataListBinderDefault',
    url : '[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions'
  },
  {
    label : '@@userview.ganttchart.mapping@@',
    type : 'header'
  },
  {
    name : 'category',
    label : '@@userview.ganttchart.mapping.category@@',
    type : 'textfield',
    required : 'True'
  },
  {
    name : 'task',
    label : '@@userview.ganttchart.mapping.task@@',
    type : 'textfield',
    required : 'True'
  },
  {
    name : 'activity',
    label : '@@userview.ganttchart.mapping.activity@@',
    type : 'textfield',
    required : 'True'
  },
  {
    name : 'fromDate',
    label : '@@userview.ganttchart.mapping.fromDate@@',
    type : 'textfield',
    required : 'True'
  },
  {
    name : 'toDate',
    label : '@@userview.ganttchart.mapping.toDate@@',
    type : 'textfield',
    required : 'True'
  },
  {
    name : 'dateFormat',

```

```

        label : '@@userview.ganttchart.mapping.dateFormat@',
        type : 'textfield',
        required : 'True',
        value : 'yyyy-MM-dd'
    },
    {
        name : 'taskId',
        label : '@@userview.ganttchart.mapping.taskId@',
        type : 'textfield'
    },
    {
        name : 'cssClass',
        label : '@@userview.ganttchart.mapping.cssClass@',
        type : 'textfield'
    }
  ]
},
{
  title : '@@userview.ganttchart.advanced@',
  properties : [
    {
      name : 'itemsPerPage',
      label : '@@userview.ganttchart.itemsPerPage@',
      type : 'textfield',
      required : 'True',
      value : '20'
    },
    {
      name : 'navigate',
      label : '@@userview.ganttchart.navigate@',
      type : 'selectbox',
      required : 'True',
      value : 'scroll',
      options : [{
        value : 'buttons',
        label : '@@userview.ganttchart.navigate.buttons@'
      },
      {
        value : 'scroll',
        label : '@@userview.ganttchart.navigate.scroll@'
      }
    ]
    },
    {
      name : 'scale',
      label : '@@userview.ganttchart.scale@',
      type : 'selectbox',
      required : 'True',
      value : 'days',
      options : [{
        value : 'hours',
        label : '@@userview.ganttchart.scale.hours@'
      },
      {
        value : 'days',
        label : '@@userview.ganttchart.scale.days@'
      },
      {
        value : 'weeks',
        label : '@@userview.ganttchart.scale.weeks@'
      },
      {
        value : 'months',
        label : '@@userview.ganttchart.scale.months@'
      }
    ]
    },
    {
      name : 'maxScale',
      label : '@@userview.ganttchart.maxScale@',
      type : 'selectbox',
      required : 'True',
      value : 'months',
      options : [{

```

```

        value : 'hours',
        label : '@@userview.ganttchart.scale.hours@@'
    },
    {
        value : 'days',
        label : '@@userview.ganttchart.scale.days@@'
    },
    {
        value : 'weeks',
        label : '@@userview.ganttchart.scale.weeks@@'
    },
    {
        value : 'months',
        label : '@@userview.ganttchart.scale.months@@'
    }
    ]
},
{
    name : 'minScale',
    label : '@@userview.ganttchart.minScale@@',
    type : 'selectbox',
    required : 'True',
    value : 'hours',
    options : [{
        value : 'hours',
        label : '@@userview.ganttchart.scale.hours@@'
    },
    {
        value : 'days',
        label : '@@userview.ganttchart.scale.days@@'
    },
    {
        value : 'weeks',
        label : '@@userview.ganttchart.scale.weeks@@'
    },
    {
        value : 'months',
        label : '@@userview.ganttchart.scale.months@@'
    }
    ]
},
{
    name : 'useCookie',
    label : '@@userview.ganttchart.useCookie@@',
    type : 'checkbox',
    options : [{
        value : 'true',
        label : ''
    }
    ]
},
{
    name : 'scrollToToday',
    label : '@@userview.ganttchart.scrollToToday@@',
    type : 'checkbox',
    options : [{
        value : 'true',
        label : ''
    }
    ]
},
{
    name : 'onItemClick',
    label : '@@userview.ganttchart.onItemClick@@',
    type : 'codeeditor',
    mode : 'javascript',
    value : '//console.log(data); //data obj hold all the columns value of a row'
},
{
    name : 'onAddClick',
    label : '@@userview.ganttchart.onAddClick@@',
    type : 'codeeditor',
    mode : 'javascript',
    value : '//console.log(datetime); //the DateTime in ms for the clicked Cell\n//console.log(rowId);
//the row ID of clicked Cell'
}

```



```

        from: "/Date(1320192000000)/",
        to: "/Date(1322401600000)/",
        label: "Requirement Gathering",
        customClass: "ganttRed"
    }
  ],
  {
    name: " ",
    desc: "Scoping",
    values: [
      {
        from: "/Date(1322611200000)/",
        to: "/Date(1323302400000)/",
        label: "Scoping",
        customClass: "ganttRed"
      }
    ]
  },
  {
    name: "Sprint 1",
    desc: "Development",
    values: [
      {
        from: "/Date(1323802400000)/",
        to: "/Date(1325685200000)/",
        label: "Development",
        customClass: "ganttGreen"
      }
    ]
  },
  {
    name: " ",
    desc: "Showcasing",
    values: [
      {
        from: "/Date(1325685200000)/",
        to: "/Date(1325695200000)/",
        label: "Showcasing",
        customClass: "ganttBlue"
      }
    ]
  },
  {
    name: "Sprint 2",
    desc: "Development",
    values: [
      {
        from: "/Date(1326785200000)/",
        to: "/Date(1325785200000)/",
        label: "Development",
        customClass: "ganttGreen"
      }
    ]
  },
  {
    name: " ",
    desc: "Showcasing",
    values: [
      {
        from: "/Date(1328785200000)/",
        to: "/Date(1328905200000)/",
        label: "Showcasing",
        customClass: "ganttBlue"
      }
    ]
  },
  {
    name: "Release Stage",
    desc: "Training",
    values: [
      {
        from: "/Date(1330011200000)/",
        to: "/Date(1336611200000)/",
        label: "Training",
        customClass: "ganttOrange"
      }
    ]
  },
  {
    name: " ",
    desc: "Deployment",
    values: [
      {
        from: "/Date(1336611200000)/",
        to: "/Date(1338711200000)/",
        label: "Deployment",
        customClass: "ganttOrange"
      }
    ]
  },
  {
    name: " ",
    desc: "Warranty Period",

```

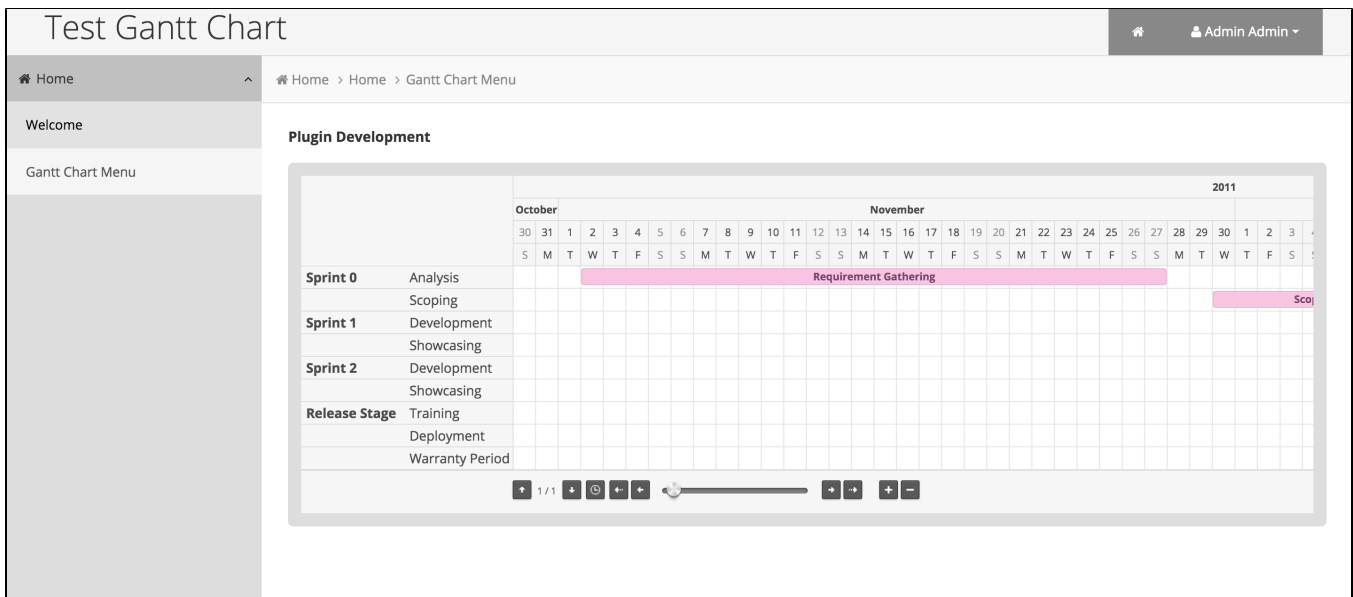


```

        values: [{
            from: "/Date(1336611200000)/",
            to: "/Date(1349711200000)/",
            label: "Warranty Period",
            customClass: "ganttOrange"
        }]
    },
    navigate: "scroll",
    maxScale: "hours",
    itemsPerPage: 10
});
});
</script>
</div>

```

cd Felix Framework f



HTMLgetRenderPagegantChart.ftl

```

@Override
public String getRenderPage() {
    Map model = new HashMap();
    model.put("request", getRequestParameters());
    model.put("element", this);

    //populate data in JSON format
    model.put("data", getData());

    PluginManager pluginManager = (PluginManager)AppUtil.getApplicationContext().getBean("pluginManager");
    String content = pluginManager.getPluginFreeMarkerTemplate(model, getClass().getName(), "/templates
/ganttChart.ftl", MESSAGE_PATH);
    return content;
}

protected String getData() {
    String json = "[]";

    try {
        DataListCollection data = null;
        String idColumn = getPropertyString("taskId");

        //get the binder
        Object binderData = getProperty("binder");
        if (binderData != null && binderData instanceof Map) {
            Map bdMap = (Map) binderData;
            if (bdMap != null && bdMap.containsKey("className") && !bdMap.get("className").toString().

```

```

isEmpty() {
    PluginManager pluginManager = (PluginManager) AppUtil.getApplicationContext().getBean
("pluginManager");
    DataListBinder binder = (DataListBinder) pluginManager.getPlugin(bdMap.get("className").
toString());

    if (binder != null) {
        Map bdProps = (Map) bdMap.get("properties");
        binder.setProperties(bdProps);

        data = binder.getData(null, bdProps, new DataListFilterQueryObject[0], null, null,
null, null);

        if (idColumn.isEmpty()) {
            idColumn = binder.getPrimaryKeyColumnName();
        }
    }
}

JSONArray dataArray = new JSONArray();
String dateFormat = getPropertyString("dateFormat");
SimpleDateFormat sdf = new SimpleDateFormat(dateFormat);

if (data != null && !data.isEmpty()) {
    String currentCategory = null;
    int cat_count = 0;
    String currentTask = null;
    int task_count = 0;
    int act_count = 0;

    JSONObject taskObj = null;
    JSONArray taskValuesArray = null;
    for (Object r : data) {
        String id = getValue(r, idColumn);
        String cat = getValue(r, getPropertyString("category"));
        String task = getValue(r, getPropertyString("task"));
        String act = getValue(r, getPropertyString("activity"));
        String formDate = getValue(r, getPropertyString("fromDate"));
        String toDate = getValue(r, getPropertyString("toDate"));
        String status = getValue(r, getPropertyString("cssClass"));

        if (currentTask == null || !currentTask.equals(task) || (currentTask.equals(task) && !cat.
equals(currentCategory))) {
            currentTask = task;
            if (taskObj != null) {
                taskObj.put("values", taskValuesArray);
                dataArray.put(taskObj);
            }
            taskObj = new JSONObject();
            taskValuesArray = new JSONArray();
            taskObj.put("desc", task);
            taskObj.put("id", id);
            task_count++;
        }

        if (currentCategory == null || !currentCategory.equals(cat)) {
            currentCategory = cat;
            taskObj.put("name", cat);
            cat_count++;
        }

        JSONObject valueObj = new JSONObject();
        JSONObject actObj = new JSONObject();
        act_count++;

        actObj.put("taskId", id);
        actObj.put("category", cat);
        actObj.put("task", task);
        actObj.put("activity", act);
        actObj.put("formDate", formDate);
        actObj.put("toDate", toDate);
    }
}

```

```

        actObj.put("status", status);

        valueObj.put("dataObj", actObj);
        valueObj.put("label", act);
        valueObj.put("from", "/" + Date("+sdf.parse(formDate).getTime()+")");
        valueObj.put("to", "/" + Date("+sdf.parse(toDate).getTime()+")");
        valueObj.put("customClass", "cat_" + cat_count + " task_" + task_count + " act_" + act_count +
"+status.replace(" ", "_"));
        taskValuesArray.put(valueObj);
    }
    if (taskObj != null) {
        taskObj.put("values", taskValuesArray);
        dataArray.put(taskObj);
    }
}

    return dataArray.toString();
} catch (Exception e) {
    LogUtil.error(GanttChartMenu.class.getName(), e, json);
}

return json;
}

protected String getValue(Object o, String name) {
    String value = "";

    try {
        Object v = LookupUtil.getBeanProperty(o, name);
        if (v != null) {
            return v.toString();
        }
    } catch (Exception e) {
        LogUtil.error(GanttChartMenu.class.getName(), e, name);
    }

    return value;
}

```

```

<link href="${request.contextPath}/plugin/org.joget.tutorial.GanttChartMenu/lib/jquery.gantt/css/style.css"
rel="stylesheet" type="text/css" />
<script src="${request.contextPath}/plugin/org.joget.tutorial.GanttChartMenu/lib/jquery.gantt/js/jquery.fn.
gantt.min.js"></script>
<div class="gantt_chart_menu_body">
  <#if element.properties.title?? ><h3>${element.properties.title!}</h3></#if>
  ${element.properties.customHeader!}
  <div class="gantt"></div>
  <script>
    $(function() {
      "use strict";
      $(".gantt").gantt({
        source: ${data!},
        months: [@@userview.ganttChart.months.label@@],
        dow: [@@userview.ganttChart.dow.label@@],
        itemsPerPage: ${element.properties.itemsPerPage!},
        navigate: "${element.properties.navigate!}",
        scale: "${element.properties.scale!}",
        maxScale: "${element.properties.maxScale!}",
        minScale: "${element.properties.minScale!}",
        waitText: "%%userview.ganttChart.waitText%%",
        onItemClick: function (data) {
          ${element.properties.onItemClick!}
        },
        onAddClick: function(datetime, rowId) {
          ${element.properties.onAddClick!}
        },
        onRender: function() {
          ${element.properties.onRender!}
        },
        useCookie: <#if element.properties.useCookie! == 'true'>true<#else>false</#if>,
        scrollToToday: <#if element.properties.scrollToToday! == 'true'>true<#else>false</#if>
      });
    });
  </script>
  ${element.properties.customFooter!}
</div>

```

c.

POM

```

<!-- Change plugin specific dependencies here -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.0</version>
</dependency>
<dependency>
  <groupId>org.json</groupId>
  <artifactId>json</artifactId>
  <version>20080701</version>
</dependency>
<dependency>
  <groupId>displaytag</groupId>
  <artifactId>displaytag</artifactId>
  <version>1.2</version>
  <exclusions>
    <exclusion>
      <artifactId>slf4j-api</artifactId>
      <groupId>org.slf4j</groupId>
    </exclusion>
    <exclusion>
      <artifactId>jcl104-over-slf4j</artifactId>
      <groupId>org.slf4j</groupId>
    </exclusion>
    <exclusion>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<!-- End change plugin specific dependencies here -->

```

d.

getLabelgetDescription118ni18n

"gantt_chart_menu / src / main"resources / messages"GanttChartMenu.properties"

```

org.joget.tutorial.GanttChartMenu.pluginLabel=Gantt Chart Menu
org.joget.tutorial.GanttChartMenu.pluginDesc=Display data in a Gantt Chart view
userview.ganttchart.config=Configure Gantt Chart Menu
userview.ganttchart.customId=Custom ID
userview.ganttchart.invalidId=Only alpha-numeric and underscore characters allowed
userview.ganttchart.label=Label
userview.ganttchart.title=Page Title
userview.ganttchart.binder=Data Binder
userview.ganttchart.mapping=Column to Data Mappings
userview.ganttchart.mapping.category=Category (column ID)
userview.ganttchart.mapping.task=Task (column ID)
userview.ganttchart.mapping.activity=Activity (column ID)
userview.ganttchart.mapping.fromDate=Activity From Date (column ID)
userview.ganttchart.mapping.toDate=Activity To Date (column ID)
userview.ganttchart.mapping.dateFormat=Date format for Activity From/To Date
userview.ganttchart.mapping.taskId=Task Id (column ID)
userview.ganttchart.mapping.cssClass=Status (column ID, use as CSS class for styling)
userview.ganttchart.advanced=Advanced
userview.ganttchart.itemsPerPage=Item pre page
userview.ganttchart.navigate=Navigator
userview.ganttchart.navigate.buttons=Buttons
userview.ganttchart.navigate.scroll=Scroll
userview.ganttchart.scale=Default Scale
userview.ganttchart.scale.hours=Hours
userview.ganttchart.scale.days=Days
userview.ganttchart.scale.weeks=Weeks
userview.ganttchart.scale.months=Months
userview.ganttchart.maxScale=Maximum Scale
userview.ganttchart.minScale=Minimum Scale
userview.ganttchart.useCookie=Use cookie for storing chart states
userview.ganttchart.scrollToToday=Auto scroll to today after rendered
userview.ganttchart.onItemClick=On Item Clicked Event (Javascript)
userview.ganttchart.onAddClick=On Empty Space Clicked Event (Javascript)
userview.ganttchart.onRender=On Rendered Event (Javascript)
userview.ganttchart.customHeader=Custom Header (HTML)
userview.ganttchart.customFooter=Custom Footer (HTML)
userview.ganttChart.months.label="January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"
userview.ganttChart.dow.label="S", "M", "T", "W", "T", "F", "S"
userview.ganttChart.waitText>Loading...

```

e. Felix

ActivatorFelix

```

public void start(BundleContext context) {
    registrationList = new ArrayList<ServiceRegistration>();
    //Register plugin here
    registrationList.add(context.registerService(GanttChartMenu.class.getName(), new GanttChartMenu(),
null));
}

```

f.

"gantt_chart_menu / target"gantt_chart_menu-5.0.0.jar"

jar jar

Filter by Type

Userview Menu

<input type="checkbox"/>	Plugin Name	Plugin Description	Plugin Version	
<input type="checkbox"/>	CRUD	CRUD element for data ma	5.0.0	
<input type="checkbox"/>	Import Menu	The Import Menu enables €	5.0.0	
<input type="checkbox"/>	Datalist Inbox	Datalist Inbox list down ass	5.0.0	
<input type="checkbox"/>	List		5.0.0	
<input type="checkbox"/>	Form	Userview Form Menu Item	5.0.0	
<input type="checkbox"/>	Gantt Chart Menu	Display data in a Gantt Cha	5.0.0	
<input type="checkbox"/>	HTML Page		5.0.0	
<input type="checkbox"/>	Inbox		5.0.0	
<input type="checkbox"/>	JasperReports	JasperReports Userview P	5.0.0	
<input type="checkbox"/>	Link		5.0.0	

10

Page 1 of 2

Displaying 1 to 10 of 16 items

""Userview.

Design Userview

Design Your Menu

Settings

Theme & Permission

Preview

Preview Userview

Save

Save Userview

Undo

Redo

Run Process

Enterprise

CRUD

Datalist Inbox

Import Menu

JasperReports

Process Status

SLA Report

SQL Chart

Universal Inbox

User Profile

Marketplace

Gantt Chart Menu

Test Gantt Chart

Click to edit

#date.EEE, d MMM yyyy#

Logout

Menu

<i class='icon-home'></i> Home

Drop menu item here

Welcome

<i class='icon-tasks'></i>Manage activity

Drop menu item here

Manage activity

Gantt Chart Menu

<Content>

Powered by Joget

Configure Gantt Chart Menu

Configure Gantt Chart Menu > Data Binder (Form Data Binder) > Advanced

Custom ID

Label *

Gantt Chart Menu

Page Title

Gantt Chart Menu

Data Binder *

Form Data Binder

Column to Data Mappings

Category (column ID) *

category

Task (column ID) *

task

Activity (column ID) *

activity

Activity From Date (column ID) *

fromDate

Activity To Date (column ID) *

toDate

Date format for Activity From/To Date *

yyyy-MM-dd

Task Id (column ID)

Status (column ID, use as CSS class for styling)

status

< Prev

Next >

OK

Cancel

====/

Select Form

Configure Gantt Chart Menu > Select Form > Advanced

Form

activity

Extra Filter Condition

< Prev

Next >

OK

Cancel

Advanced

Configure Gantt Chart Menu > Data Binder (Form Data Binder) > Advanced

Item pre page *

20

Navigator *

Scroll

Default Scale *

Days

Maximum Scale *

Months

Minimum Scale *

Months

Use cookie for storing chart states

☒

Auto scroll to today after rendered

☒

On Item Clicked Event (Javascript)

1 console.log(data); //data obj hold all the columns value of a row

On Empty Space Clicked Event (Javascript)

1 console.log(datetime); //the DateTime in ms for the clicked Cell

2 console.log(rowId); //the row ID of clicked Cell

< Prev

Next >

OK

Cancel

Custom Footer (HTML)

1 <style>

2 .fn-gantt .Done {

3 background-color: #D8EDA3;

4 }

5 .fn-gantt .Pending {

6 background-color: #FCD29A;

7 }

8 .fn-gantt .Aborted {

9 background-color: #F9C4E1;

10 }

11 </style>

"HTML" css

	Category	Task	Activity	From Date	To Date	Status	
<input type="checkbox"/>	Sprint 0	Analysis	Requirement Gathering	2015-01-05	2015-01-12	Done	Edit
<input type="checkbox"/>	Sprint 0	Scoping	Scoping	2015-01-13	2015-01-16	Done	Edit
<input type="checkbox"/>	Sprint 1	Scoping	Scoping	2015-01-19	2015-01-21	Done	Edit
<input type="checkbox"/>	Sprint 1	Development	Project Setup	2015-01-22	2015-01-23	Done	Edit
<input type="checkbox"/>	Sprint 1	Development	Module 1	2015-01-26	2015-01-28	Done	Edit
<input type="checkbox"/>	Sprint 1	Showcasing	Showcasing	2015-01-29	2015-01-30	Done	Edit
<input type="checkbox"/>	Sprint 2	Development	Module 2	2015-02-02	2015-02-10	Done	Edit
<input type="checkbox"/>	Sprint 2	Development	Module 3	2015-02-11	2015-02-12	Aborted	Edit
<input type="checkbox"/>	Sprint 2	Showcasing	Showcasing	2015-02-12	2015-02-13	Done	Edit
<input type="checkbox"/>	Sprint 3	Development	Module 3	2015-02-16	2015-02-25	Done	Edit

Admin Admin

[Home](#) > [Manage activity](#) > [Gantt Chart Menu](#)

Gantt Chart Menu

17

The Gantt chart displays the project schedule from January 1st to February 8th. The tasks and their durations are as follows:

Task	Start Date	End Date	Duration
Sprint 0 Analysis	Jan 1	Jan 1	1 day
Sprint 0 Scoping	Jan 1	Jan 1	1 day
Sprint 1 Scoping	Jan 1	Jan 1	1 day
Sprint 1 Development	Jan 1	Jan 1	1 day
Sprint 1 Showcasing	Jan 1	Jan 1	1 day
Sprint 2 Development	Jan 1	Jan 1	1 day
Sprint 2 Showcasing	Jan 1	Jan 1	1 day
Sprint 3 Development	Jan 1	Jan 1	1 day
Sprint 3 Showcasing	Jan 1	Jan 1	1 day
Sprint 4 Testing	Jan 1	Jan 1	1 day
Release Stage Training	Jan 1	Jan 1	1 day
Release Stage Deployment	Jan 1	Jan 1	1 day
Release Stage Warranty Period	Jan 1	Jan 1	1 day

[gantt_chart_menu.zip](#)

APP_testGanttChart-1-20151106194035.jwa

jar<http://marketplace.joget.org/>