VARCHART

JGantt Workshop 4, Part 2

JGantt JSF Edition
For Web Applications

Jürgen Theis

NETRONIC

JGantt JSF Edition

Overview
Demonstration
Concept
Requirements
Pros & Cons

- **Goal of the product**
  - Support of developing web applications
  - No need of Java on client side
- **State of the project**
  - We are at the design phase
  - First prototype has been developed
- **Goal of this presentation part**
  - Give an overview of the concept
  - Get feedback
  - Discuss the proposed approach

Demonstration

Overview
Demonstration
Concept
Requirements
Pros & Cons

What are we talking about?

Some samples showing

- Interactions (selection, scrolling, ...)
- Editing by standard JSF components

GanttFaces.jsf

What is JGantt JSF?

Overview
Demonstration
Concept
Requirements
Pros & Cons

- **It is a JavaServer Faces component**
- **It follows the concept of server side graphics**
- **It contains**
  - the JComponent JGantt as a kernel
  - the tag class JGanttChartTag
  - the UIComponent class UIJGanttChart
  - the renderer class JGanttChartRenderer
- **It is attributed by implementing the interface** JGIGanttConfigurator

Concept - Requesting JSF-Pages

Overview
Demonstration
Concept
Requirements
Pros & Cons

Servlet Container

JSF Servlet

New modules

What you have to do ...

Client

Web Inter/Intranet

Server

Servlet Container

1) request
JSF-page

JSF Servlet

2) load
& parse

JSP-file

Application class
(backing bean)

5) generate
HTML

JGanttChartTag
JGanttChartRenderer

UIGanttChart

HTML Code

```
<html>
  <body>
...
<input type="image" name="gantt1" style="height: 600px; width: 800px; border:3px solid" src="graphservlet?id=gantt1&amp;data=links.xml&amp;config=TestConfiguratorClass&amp;format=PNG_ACT" />
.....
```

3) create

6) request
servlet

GraphServlet

generate
image

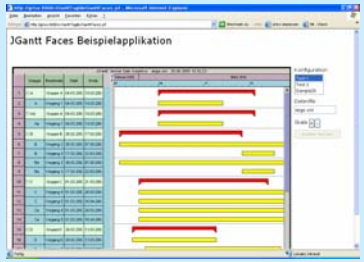
GraphGenerator

JGantt

Session State

4) attribute

TestConfiguratorClass



7) image
data

Client

Web

Server

Requirements

Overview
Demonstration
Concept
Requirements
Pros & Cons

- **On server side**
  - Any servlet container supporting Servlet 2.3 and JSP 1.2 specification (e.g. Tomcat 5)
- **On client side**
  - Any HTML browser running JavaScript
  - Nothing else!

Limitations of JGantt JSF as Opposed to JGantt

NETRONIC

Overview
Demonstration
Concept
Requirements
Pros & Cons

- **Reduced interaction capabilities**
  - no internal dialogs at runtime
  - no in-place editing
  - no drag&drop
  - no context menus
  - no tool tips while moving cursor
- **Reduced set of events**
- **No printing in the first version**

Benefits

Overview
Demonstration
Concept
Requirements
Pros & Cons

- **API same as the one of JGantt**
- **Low data transfer**
- **No Java Virtual Machine on client side**

Thank you for your attention.

Any questions?
Any proposals?

Additional Slides

JSP Page (Excerpt of GanttFaces.jsp)

```
<%@ taglib uri="http://netronic.de/jgantt/taglibJSF" prefix="jg" %>
 …

<jg:ganttChart  binding="#{ganttFaces.ganttChart}"
                id="gantt1"
                style="height: 600px; width: 800px; border:3px solid”
                config =”TestConfiguratorClass”
                data="#{ganttFaces.textFieldData.value}"/>
…..
```

back

HTML code sent back to client

```
<html>
  <body>
   …
    <input type="image"
          name="gantt1"
          style="height: 600px; width: 800px; border:3px solid"
          src="graphservlet?id=gantt1&amp;amp;name=links.xml&amp;amp;
              config=TestConfiguratorClass&amp;amp;format=PNG_ACT“
    />
…..
   </body>
</html>
```

back

TestConfiguratorClass (implements JGIGanttConfigurator)

```
public void initSession (JGantt jGantt, Map attrs)
    {
    jGantt.setNodeDates (new String[] { "start", "end" });
    jGantt.setNodeSetName ("Nodes");
    jGantt.setGroupNodeDates (new String[] { "start", "end" });

    loadAppData();
    ... }

public void initRequest (JGantt jGantt, Map attrs)
    {
   String name = (String) attrs.get ("name");

    if ( ! name.equals (oldName))
            reloadData(); // new Gantt graph

     ... }
```

back