

Tapestry5HowToCreateGenericGraphComponent

Based on the [Tapestry5HowToCreateASimpleGraphComponent](#) this is how you can create a component that can be passed a JFreeChart object to allow you to create any kind of chart

To use add this to your page:

```
<t:chart width="200" height="150" chart="lineChart"/>
```

add this to your page class:

```
public JFreeChart getLineChart() {
    XYSeries series = new XYSeries("Average Size");
    series.add(20.0, 10.0);
    series.add(40.0, 20.0);
    series.add(70.0, 50.0);
    XYDataset xyDataset = new XYSeriesCollection(series);
    final JFreeChart chart = ChartFactory.createXYLineChart
("Sample XY Chart", // Title
    "Height",        // X-Axis label
    "Weight",        // Y-Axis label
    xyDataset,        // Dataset
    PlotOrientation.HORIZONTAL,
    true,             // Show legend
    true,             // Tooltips
    true              // Urls
);

    return chart;
}
```

add this to your pom.xml under "dependencies":

```
<dependency>
    <groupId>jfree</groupId>
    <artifactId>jfreechart</artifactId>
    <version>1.0.5</version>
</dependency>
```

Source:

(put this component class into yourApp.components package)

```
package com.liftyourgame.application.components;

import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;

import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.Link;
import org.apache.tapestry5.MarkupWriter;
import org.apache.tapestry5.StreamResponse;
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.dom.Element;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.ioc.services.TypeCoercer;
import org.apache.tapestry5.services.Response;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;

public class Chart {
    @Inject
    private ComponentResources _resources;
```

```

@Inject
private TypeCoercer _coercer;

/**list(array) of paired values(label & value): [String,Number,String,Number,...]*/
@Parameter(required=true)
private JFreeChart _chart;

@Persist
private JFreeChart _context;

@Parameter(required=true)
private int _width;

@Parameter(required=true)
private int _height;

void beginRender(MarkupWriter writer)
{
    _context=_chart;
    Object[] params = { new Integer(_width),
                        new Integer(_height) };

    //generate action link
    Link link = _resources.createEventLink("chart", params);
    Element img = writer.element("img", "src", link);
    _resources.renderInformalParameters(writer);
}

void afterRender(MarkupWriter writer)
{
    writer.end();
}

public StreamResponse onChart(final int width, final int height/*, Object ...rest*/) {

    return new StreamResponse(){
        public String getContentType(){
            return "image/jpeg";
        }
        public InputStream getStream() throws IOException {
            BufferedImage image = _context.createBufferedImage(width, height);
            ByteArrayOutputStream byteArray = new ByteArrayOutputStream() ;
            ChartUtilities.writeBufferedImageAsJPEG(byteArray, image) ;
            return new ByteArrayInputStream(byteArray.toByteArray());
        }
        public void prepareResponse(Response response){}
    };
}

```