Quick Reference

Version 2.3-SNAPSHOT

Quick Reference

Chapter 1. Core Chart

1.1. Pie Chart

VisualizationController.groovy

```
class VisualizationController {
  def pieChart() {
   List myDailyActivitiesColumns = [['string', 'Task'], ['number', 'Hours per Day']]
   List myDailyActivitiesData = [['Work', 11], ['Eat', 2], ['Commute', 2], ['Watch
TV', 2], ['Sleep', 7]]
   render template: "pieChart", model: ["myDailyActivitiesColumns":
myDailyActivitiesColumns,
    "myDailyActivitiesData": myDailyActivitiesData]
  }
}
```

_pieChart.gsp

```
<gvisualization:pieCoreChart elementId="piechart" title="My Daily Activities"
width="${450}" height="${300}" columns="${myDailyActivitiesColumns}"
data="${myDailyActivitiesData}" />


<div id="piechart"></div>
```

1.2. Bar Chart

VisualizationController.groovy

```
class VisualizationController {
  def barChart() {
    List companyPerformanceColumns = [['string', 'Year'], ['number', 'Sales'],
['number', 'Expenses']]
    List companyPerformanceData = [['2004', 1000, 400], ['2005', 1170, 460], ['2006',
660, 1120], ['2007', 1030, 540]]
    render template: "barChart", model: ["companyPerformanceColumns":
companyPerformanceColumns, "companyPerformanceData": companyPerformanceData]
  }
}
```

_barChart.gsp

```
 <gvisualization:barCoreChart elementId="barchart" title="Company Performance"
width="${400}" height="${240}" vAxis="${[title: 'Year', titleColor: 'red']}"
columns="${companyPerformanceColumns}" data="${companyPerformanceData}" />


 <div id="barchart"></div>
```

1.3. Bubble Chart

VisualizationController.groovy

```
class VisualizationController {
  def bubbleChart() {
    List lifeExpectancyFertilityRateColumns =[['string', 'ID'], ['number', 'Life
Expectancy'], ['number', 'Fertility Rate'], ['string', 'Region'], ['number',
'Population']]
    List lifeExpectancyFertilityRateData = [['CAN', 80.66, 1.67, 'North America',
33739900], ['DEU', 79.84, 1.36, 'Europe', 81902307], ['DNK', 78.6, 1.84, 'Europe',
5523095], ['EGY', 72.73, 2.78, 'Middle East', 79716203], ['GBR', 80.05, 2, 'Europe',
61801570], ['IRN', 72.49, 1.7, 'Middle East', 73137148], ['IRQ', 68.09, 4.77, 'Middle
East', 31090763], ['ISR', 81.55, 2.96, 'Middle East', 7485600], ['RUS', 68.6, 1.54,
'Europe', 141850000], ['USA', 78.09, 2.05, 'North America', 307007000]]
    render template: "bubbleChart", model:
["lifeExpectancyFertilityRateColumns":lifeExpectancyFertilityRateColumns,
"lifeExpectancyFertilityRateData": lifeExpectancyFertilityRateData]
  }
}
```

_bubbleChart.gsp

```
<gvisualization:bubbleCoreChart elementId="bubblechart" title="Correlation between
life expectancy, fertility rate and population of some world countries (2010)"
hAxis="${[title: 'Life Expectancy']}" vAxis="${[title: 'Fertility Rate']}"
bubble="${[textStyle: '{fontSize: 11}']}"
columns="${lifeExpectancyFertilityRateColumns}"
data="${lifeExpectancyFertilityRateData}" />


<div id="bubblechart" style="width: 900px; height: 500px;"></div>
```

1.4. Column Chart

VisualizationController.groovy

```
class VisualizationController {
  def columnChart() {
    List companyPerformanceColumns = [['string', 'Year'], ['number', 'Sales'],
['number', 'Expenses']]
    List companyPerformanceData = [['2004', 1000, 400], ['2005', 1170, 460], ['2006',
660, 1120], ['2007', 1030, 540]]
    render template: "columnChart", model:
["companyPerformanceColumns":companyPerformanceColumns, "companyPerformanceData":
companyPerformanceData]
  }
}
```

_columnChart.gsp

```
<gvisualization:columnCoreChart elementId="columnchart" title="Company Performance"
width="${400}" height="${240}" hAxis="${[title: 'Year', titleColor: 'red']}"
columns="${companyPerformanceColumns}" data="${companyPerformanceData}" />


<div id="columnchart"></div>
```

1.5. Area Chart

VisualizationController.groovy

```
class VisualizationController {
  def areaChart() {
    List companyPerformanceColumns = [['string', 'Year'], ['number', 'Sales'],
['number', 'Expenses']]
    List companyPerformanceData = [['2004', 1000, 400], ['2005', 1170, 460], ['2006',
660, 1120], ['2007', 1030, 540]]
    render template: "areaChart", model: ["companyPerformanceColumns",
companyPerformanceColumns, "companyPerformanceData": companyPerformanceData]
  }
}
```

_areaChart.gsp

```
<gvisualization:areaCoreChart elementId="areachart" title="Company Performance"
width="${400}" height="${240}" hAxis="${[title: 'Year', titleColor: 'red']}"
columns="${companyPerformanceColumns}" data="${companyPerformanceData}" />


<div id="areachart"></div>
```

1.6. Line Chart

VisualizationController.groovy

```
class VisualizationController {
  def lineChart() {
    List companyPerformanceColumns = [['string', 'Year'], ['number', 'Sales'],
['number', 'Expenses']]
    List companyPerformanceData = [['2004', 1000, 400], ['2005', 1170, 460], ['2006',
660, 1120], ['2007', 1030, 540]]
    render template: "lineChart", model: ["companyPerformanceData":
companyPerformanceData, "companyPerformanceColumns": companyPerformanceColumns]
  }
}
```

_lineChart.gsp

```
<gvisualization:lineCoreChart elementId="linechart" width="${400}" height="${240}"
title="Company Performance" columns="${companyPerformanceColumns}"
data="${companyPerformanceData}" />

<div id="linechart"></div>
```

1.7. Scatter Chart

VisualizationController.groovy

```
class VisualizationController {
  def scatterChart() {
    List weightByAgeColumns = [['number', 'Age'], ['number', 'Weight']]
    List weightByAgeData = [[8, 12], [4, 5.5], [11, 14], [4, 5], [3, 3.5], [6.5, 7]]
    render template: "scatterChart", model: ["weightByAgeData": weightByAgeData]
  }
}
```

_scatterChart.gsp

```
<gvisualization:scatterCoreChart elementId="scatterchart" width="${400}"
height="${240}" title="Age vs. Weight comparison" hAxis="${[title: 'Age', minValue: 0,
maxValue: 15]}" vAxis="${[title: 'Weight', minValue: 0, maxValue: 15]}" legend="none"
columns="${weightByAgeColumns}" data="${weightByAgeData}" />


<div id="scatterchart"></div>
```

1.8. Stepped Area Chart

VisualizationController.groovy

```
class VisualizationController {
  def steppedAreaChart() {
    List accumulatedRatingColumns = [['string', 'Director (Year)'], ['number', 'Rotten
Tomatoes'], ['number', 'IMDB']]
    List accumulatedRatingData = [['Alfred Hitchcock (1935)', 8.4, 7.9], ['Ralph
Thomas (1959)', 6.9, 6.5], ['Don Sharp (1978)',6.5, 6.4], ['James Hawes (2008)', 4.4,
6.2]]
    render template: "steppedAreaChart", model: ["accumulatedRatingColumns":
accumulatedRatingColumns, "accumulatedRatingData":accumulatedRatingData]
  }
}
```

_steppedAreaChart.gsp

```
<gvisualization:steppedAreaCoreChart elementId="steppedareachart" width="${400}"
height="${240}" title="The decline of \'The 39 Steps\'" vAxis="${[title: 'Accumulated
Rating']}" isStacked="${true}" columns="${accumulatedRatingColumns}"
data="${accumulatedRatingData}" />


<div id="steppedareachart"></div>
```

1.9. Candlestick Chart

VisualizationController.groovy

```
class VisualizationController {
  def columnChart() {
    List countByDayColumns = [['string', 'Day'], ['number', ''], ['number', ''],
['number', ''], ['number', '']]
    List countByDayData = [['Mon', 20, 28, 38, 45], ['Tues', 31, 38, 55, 66], ['Wed',
50, 55, 77, 80], ['Thurs', 50, 77, 66, 77], ['Fri', 15, 66, 22, 68]]
    Map candlestickOptions = [hollowIsRising: true]
    render template: "candlestickChart", model: ["countByDayColumns":
countByDayColumns, "countByDayData": countByDayData, "candlestickOptions":
candlestickOptions]
  }
}
```

_candlestickChart.gsp

```
<gvisualization:candlestickCoreChart elementId="candlestickchart" legend="none"
columns="${countByDayColumns}" data="${countByDayData}"
candlestick="${candlestickOptions}" />


<div id="candlestickchart" style="width: 300px; height: 300px;"></div>
```

1.10. Combo Chart

VisualizationController.groovy

```
class VisualizationController {
  def comboChart() {
   List monthlyCoffeeProdByCountryColumns = [['string', 'Month'], ['number',
'Bolivia'], ['number', 'Ecuador'], ['number', 'Madagascar'], ['number', 'Papua
Guinea'], ['number', 'Rwanda'], ['number', 'Average']]
   List monthlyCoffeeProdByCountryData = [['2004/05', 165, 938, 522, 998, 450, 614.6],
['2005/06', 135, 1120, 599, 1268, 288, 682], ['2006/07', 157, 1167, 587, 807, 397,
623], ['2007/08', 139, 1110, 615, 968, 215, 609.4], ['2008/09', 136, 691, 629, 1026,
366, 569.6]]
   render template: "comboChart", model: ["monthlyCoffeeProdByCountryColumns":
monthlyCoffeeProdByCountryColumns, "monthlyCoffeeProdByCountryData":
monthlyCoffeeProdByCountryData]
  }
}
```

_comboChart.gsp

```
<gvisualization:comboCoreChart elementId="combochart" title="Monthly Coffee Production
by Country" vAxis="${[title: 'Cups']}" hAxis="${[title: 'Month']}" seriesType="bars"
series="${[5: [type: 'line']]}" columns="${monthlyCoffeeProdByCountryColumns}"
data="${monthlyCoffeeProdByCountryData}" />


<div id="combochart" style="width: 700px; height: 400px;"></div>
```

Chapter 2. Additional Charts

2.1. Gauge

VisualizationController.groovy

```
class VisualizationController {
  def gauge() {
   List systemPerformanceColumns = [['string', 'Label'], ['number', 'Value']]
   List systemPerformanceData = [['Memory', 80], ['CPU', 55], ['Network', 68]]
   render template: "gauge", model: ["systemPerformanceColumns":
systemPerformanceColumns, "systemPerformanceData": systemPerformanceData]
  }
}
```

_gauge.gsp

```
<gvisualization:gauge elementId="gauge" width="${400}" height="${120}" redFrom="${90}"
redTo="${100}" yellowFrom="${75}" yellowTo="${90}" minorTicks="${5}"
columns="${systemPerformanceColumns}" data="${systemPerformanceData}" />


<div id="gauge"></div>
```

2.2. Table

VisualizationController.groovy

```
class VisualizationController {
  def table() {
   List employeeColumns = [['string', 'Name'], ['string', 'Salary'], ['boolean', 'Full
Time Employee']]
   List employeeData = [['Mike', '$10,000', true], ['Jim', '$8,000', false], ['Alice',
'$12,500', true], ['Bob', '$7,000', true]]
   render template: "table", model: ["employeeColumns": employeeColumns,
"employeeData": employeeData]
  }
}
```

_table.gsp

```
<gvisualization:table elementId="table" width="${400}" height="${130}"
columns="${employeeColumns}" data="${employeeData}" select="selectHandler"
ready="readyHandler"/>


<div id="table"></div>
```

2.3. Map

VisualizationController.groovy

```
class VisualizationController {
  def map() {
   List mapColumns = [['number', 'Lat'], ['number', 'Lon'], ['string', 'Name']]
   List mapData = [[37.4232, -122.0853, 'Work'], [37.4289, -122.1697, 'University'],
[37.6153, -122.3900, 'Airport']
   render template: "map", model: ["mapColumns": mapColumns, "mapData": mapData]
  }
}
```

_map.gsp

```
<gvisualization:map elementId="map" columns="${mapColumns}" data="${mapData}" />


<div id="map" style="width: 400px; height: 300px"></div>
```

2.4. Annotated Time Line

VisualizationController.groovy

```
class VisualizationController {
  def annotatedTimeLine() {
   List pensColumns = [['date', 'Date'], ['number', 'Sold Pencils'], ['string',
'title1'], ['string', 'text1'], ['number', 'Sold Pens'], ['string', 'title2'],
['string', 'text2']]
   List pensData = [[DateUtil.createDate(2008, 1, 1), 30000, null, null, 40645, null,
null], [DateUtil.createDate(2008, 1, 2), 14045, null, null, 20374, null, null],
[DateUtil.createDate(2008, 1, 3), 55022, null, null, 50766, null, null],
[DateUtil.createDate(2008, 1, 4), 75284, null, null, 14334, 'Out of Stock','Ran out of
stock on pens at 4pm'], [DateUtil.createDate(2008, 1, 5), 41476, 'Bought Pens','Bought
200k pens', 66467, null, null], [DateUtil.createDate(2008, 1, 6), 33322, null, null,
39463, null, null]]
   render template: "annotatedTimeLine", model: ["pensColumns": pensColumns,
"pensData": pensData]
  }
}
```

_annotatedTimeLine.gsp

```
<gvisualization:annotatedTimeLine elementId="annotatedtimeline"
columns="${pensColumns}" data="${pensData}" />


<div id="annotatedtimeline" style='width: 700px; height: 240px;'></div>
```

2.5. Annotation Chart

VisualizationController.groovy

```
class VisualizationController {
  def annotationChart() {
   List pensColumns = [['date', 'Date'], ['number', 'Sold Pencils'], ['string',
'title1'], ['string', 'text1'], ['number', 'Sold Pens'], ['string', 'title2'],
['string', 'text2']]
   List pensData = [[DateUtil.createDate(2008, 1, 1), 30000, null, null, 40645, null,
null], [DateUtil.createDate(2008, 1, 2), 14045, null, null, 20374, null, null],
[DateUtil.createDate(2008, 1, 3), 55022, null, null, 50766, null, null],
[DateUtil.createDate(2008, 1, 4), 75284, null, null, 14334, 'Out of Stock','Ran out of
stock on pens at 4pm'], [DateUtil.createDate(2008, 1, 5), 41476, 'Bought Pens','Bought
200k pens', 66467, null, null], [DateUtil.createDate(2008, 1, 6), 33322, null, null,
39463, null, null]]
   render template: "annotationChart", model: ["pensColumns": pensColumns, "pensData":
pensData]
  }
}
```

_annotationChart.gsp

```
<gvisualization:annotationChart elementId="annotationchart" columns="${pensColumns}"
data="${pensData}" />


<div id="annotationchart" style='width: 700px; height: 240px;'></div>
```

2.6. Org Chart

VisualizationController.groovy

```
class VisualizationController {
  def orgChart() {
   def orgColumns = [['string', 'Name'], ['string', 'Manager'], ['string', 'ToolTip']]
   def orgData = [[new Cell(value: 'Mike', label: 'Mike<div style="color:red; font-
style:italic">President</div>'), '', 'The President'], [new Cell(value: 'Jim', label:
'Jim<div style="color:red; font-style:italic">Vice President<div>'), 'Mike', 'VP'],
['Alice', 'Mike', ''], ['Bob', 'Jim', 'Bob Sponge'], ['Carol', 'Bob', '']]
   render template: "orgChart", model: ["orgColumns": orgColumns, "orgData": orgData]
  }
}
```

_orgChart.gsp

```
<gvisualization:orgChart elementId="orgchart" allowHtml="${true}"
columns="${orgColumns}" data="${orgData}" />


<div id="orgchart"></div>
```

2.7. Intensity Map

VisualizationController.groovy

```
class VisualizationController {
  def intensityMap() {
   def popularityColumns = [['string', 'Country'], ['number', 'Popularity']]
   def popularityData = [['Germany', 200], ['United States', 300], ['Brazil', 400],
['Canada', 500], ['France', 600], ['RU', 700]]
   render template: "intensityMap", model: ["populationColumns": populationColumns,
"populationData": populationData]
  }
}
```

_intensityMap.gsp

```
<gvisualization:intensityMap elementId="intensitymap" columns="${populationColumns}"
data="${populationData}" />


<div id="intensitymap"></div>
```

2.8. Geo Map

VisualizationController.groovy

```
class VisualizationController {
  def geoMap() {
    def popularityColumns = [['string', 'Country'], ['number', 'Popularity']]
    def popularityData = [['Germany', 200], ['United States', 300], ['Brazil', 400],
['Canada', 500], ['France', 600], ['RU', 700]]
    render template: "geoMap", model: ["popularityColumns": popularityColumns,
"popularityData": popularityData]
  }
}
```

_geoMap.gsp

```
<gvisualization:geoMap elementId="geomap" columns="${popularityColumns}"
data="${popularityData}" />


<div id="geomap"></div>
```

2.9. Geo Chart

VisualizationController.groovy

```
class VisualizationController {
  def geoChart() {
    def popularityColumns = [['string', 'Country'], ['number', 'Popularity']]
def popularityData = [['Germany', 200], ['United States', 300], ['Brazil', 400],
['Canada', 500], ['France', 600], ['RU', 700]]
    render template: "geoChart", model: ["popularityColumns": popularityColumns,
"popularityData": popularityData]
  }
}
```

_geoChart.gsp

```
<gvisualization:geoChart elementId="geochart" width="${556}" height="${347}"
columns="${popularityColumns}" data="${popularityData}" />


<div id="geochart"></div>
```

2.10. Motion Chart

VisualizationController.groovy

```
class VisualizationController {
  def motionChart() {
   List fruitColumns = [['string', 'Fruit'], ['date', 'Date'], ['number', 'Sales'],
['number', 'Expenses'], ['string', 'Location']]
   List fruitData = [['Apples', DateUtil.createDate(1988, 0, 1), 1000, 300, 'East'],
['Oranges', DateUtil.createDate(1988, 0, 1), 1150, 200, 'West'], ['Bananas',
DateUtil.createDate(1988, 0, 1), 300, 250, 'West'], ['Apples',
DateUtil.createDate(1989, 6, 1), 1200, 400, 'East'], ['Oranges',
DateUtil.createDate(1989, 6, 1), 750, 150, 'West'], ['Bananas',
DateUtil.createDate(1989, 6, 1), 788, 617, 'West']]
   render template: "motionChart", model: ["fruitColumns": fruitColumns, "fruitData":
fruitData]
  }
}
```

_motionChart.gsp

```
<gvisualization:motionChart elementId="motionchart" columns="${fruitColumns}"
data="${fruitData}" />


<div id="motionchart"></div>
```

2.11. Tree Map

VisualizationController.groovy

```
class VisualizationController {
  def treeMap() {
   List marketByRegionColumns = [['string', 'Region'], ['string', 'Parent'],
['number', 'Market trade volume (size)'], ['number', 'Market increase/decrease
(color)']]
   List marketByRegionData = [['Global', null, 0, 0], ['America', 'Global', 0, 0],
['Europe', 'Global', 0, 0], ['Asia', 'Global', 0, 0], ['Australia', 'Global', 0, 0],
['Africa', 'Global', 0, 0], ['Brazil', 'America', 11, 10], ['USA', 'America', 52, 31],
['Mexico', 'America', 24, 12], ['Canada', 'America', 16, -23], ['France', 'Europe',
42, -11], ['Germany', 'Europe', 31, -2], ['Sweden', 'Europe', 22, -13], ['Italy',
'Europe', 17, 4], ['UK', 'Europe', 21, -5], ['China', 'Asia', 36, 4], ['Japan',
'Asia', 20, -12], ['India', 'Asia', 40, 63], ['Laos', 'Asia', 4, 34], ['Mongolia',
'Asia', 1, -5], ['Israel', 'Asia', 12, 24], ['Iran', 'Asia', 18, 13], ['Pakistan',
'Asia', 11, -52], ['Egypt', 'Africa', 21, 0], ['S. Africa', 'Africa', 30, 43],
['Sudan', 'Africa', 12, 2], ['Congo', 'Africa', 10, 12], ['Zair', 'Africa', 8, 10]]
   render template: "treeMap", model: ["marketByRegionColumns": marketByRegionColumns,
"marketByRegionData": marketByRegionData]
  }
}
```

_treeMap.gsp

```
<gvisualization:treeMap elementId="treemap" minColor="#f00" midColor="#ddd"
maxColor="#0d0" headerHeight="${15}" fontColor="black" showScale="${true}"
columns="${marketByRegionColumns}" data="${marketByRegionData}" />

<div id="treemap" style="width: 900px; height: 500px;"></div>
```

2.12. Timeline

VisualizationController.groovy

```
class VisualizationController {
  def timeline() {
   List timelineColumns = [['string', 'President'], ['date', 'Start'], ['date',
'End']]
   List timelineData = [['Washington',  DateUtil.createDate(1789, 3, 29),
DateUtil.createDate(1797, 2, 3)], ['Adams',  DateUtil.createDate(1797, 2, 3),
DateUtil.createDate(1801, 2, 3)], ['Jefferson',  DateUtil.createDate(1801, 2, 3),
DateUtil.createDate(1809, 2, 3)]]
   render template: "timeline", model: ["timelineColumns": timelineColumns,
"timelineData": timelineData]
  }
}
```

_timeline.gsp

```
<gvisualization:timeLine elementId="timeline" columns="${timelineColumns}"
data="${timelineData}" />


 <div id="timeline"></div>
```

2.13. Calendar Chart

VisualizationController.groovy

```
class VisualizationController {
  def calendarChart() {
   List calendarColumns = [['date', 'Date'], ['number', 'Won/Loss']]
   List calendarData = [[DateUtil.createDate(2012, 3, 13), 37032],
[DateUtil.createDate(2012, 3, 14), 38024], [DateUtil.createDate(2012, 3, 15), 38024],
[DateUtil.createDate(2012, 3, 16), 38108], [DateUtil.createDate(2012, 3, 17), 38229]]
   render template: "calendarChart", model: ["calendarColumns": calendarColumns,
"calendarData": calendarData]
  }
}
```

_calendarChart.gsp

```
<gvisualization:calendarChart elementId="calendarchart" columns="${calendarColumns}"
data="${calendarData}" />


<div id="calendarchart"></div>
```

2.14. Data Table Roles

VisualizationController.groovy

```
class VisualizationController {
  def dataTableRoles() {
   List dataTableRoleExampleColumns = [['string', 'Month'], ['number', 'Sales'],
[type: 'number', role: 'interval'], [type:'number', role:'interval'], [type:'string',
role:'annotation'], [type:'string', role:'annotationText'],
[type:'boolean',role:'certainty']]
   List dataTableRoleData = [
           ['April',1000,  900, 1100,  'A','Stolen data', true],
           ['May',  1170, 1000, 1200,  'B','Coffee spill', true],
           ['June',  660,  550,  800,  'C','Wumpus attack', true],
           ['July', 1030, null, null, null, null, false]
   ]
   render template: "dataTableRoles", model: ["dataTableRoleExampleColumns":
dataTableRoleExampleColumns, "dataTableRoleData": dataTableRoleData]
  }
}
```

_dataTableRoles.gsp

```
<gvisualization:lineCoreChart elementId="dataTableRoles" width="${400}"
height="${240}" columns="${dataTableRoleExampleColumns}" data="${dataTableRoleData}"
legend="${[position: 'top', alignment: 'center']}" />


<div id="dataTableRoles"></div>
```