

Group By Report

The **group by report** is similar to group by chart, but it displays the aggregated values as a table.

Report preview

| Issues | | | | | |
|-----------------------|------|---------|-----|--------|--------|
| Issue Type \ Priority | High | Highest | Low | Lowest | Medium |
| IssueType 1 | 5 | 5 | 5 | 2 | 10 |
| IssueType 2 | 9 | 7 | 8 | 8 | 5 |
| IssueType 3 | 7 | 10 | 5 | 10 | 12 |
| IssueType 4 | 6 | 7 | 17 | 11 | 4 |
| IssueType 5 | 7 | 11 | 12 | 5 | 12 |
| Task | 0 | 0 | 0 | 0 | 6 |

Parameters

| Parameter | | Parameter | Default Value |
|------------|----------|------------------|---------------|
| UI | Code | | |
| JQL | JQL | JQL Autocomplete | |
| Group By X | GroupByX | Group By Picker | Project |
| Group By Y | GroupByY | Group By Picker | Issue Type |
| Value | Value | Value Picker | IssueCount |

Layout Script

Used layout: [2D Table](#).

```

var div = $("#chart");

if(chartData.columns.length == 0)
{
    $(div).append("<h4>"+chartData.empty.label.text+"</h4>");
}
else
{
    if(chartData.custom.title)
        $(div).append("<h4>"+chartData.custom.title+"</h4>");
    var table = d3.select(div.parent().get(0)).append('table')
    var thead = table.append('thead')
    var tbody = table.append('tbody');

    $("#chart ~ table").addClass("table")
    thead.append('tr').selectAll('th').data(chartData.columns[1]).enter().append('th').text(function
(column) { return column; });

    if ( chartData.ytype )
    {
        $('thead > tr > th:first').text(chartData.ytype + " \\ " + chartData.xtype)
    }
    else
    {
        $('thead > tr > th:first').text(chartData.xtype)
    }
    var tr = tbody.selectAll('tr').data(chartData.groups[0]).enter().append("tr");

    var td = tr.selectAll("td").data(function(d)
    {
        var result = new Array();
        $.each(chartData.columns,function(index,value){
            if(value[0] == d){
                result.push.apply(result,value);
            }
        });
        return result;
    }).enter().append("td").text(function(d,i)
    {
        return d;
    });
}

```

Data Script

```

import java.lang.reflect.Field;
import java.math.BigDecimal;

import org.apache.lucene.document.Document;

import com.atlassian.jira.component.ComponentAccessor;
import com.atlassian.jira.issue.DocumentIssueImpl;
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.jql.parser.JqlParseException;
import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.query.Query;

import com.decadis.jira.xchart.api.ChartParam;
import com.decadis.jira.xchart.api.model.Period
import com.decadis.jira.xchart.api.util.DateUtils;

def clear = { String string -> (string == null || string.trim().length() == 0 ) ? "-" : string; }

```

```

def metaCountGroup = chartBuilder.newDataCollector();

JqlQueryParser jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser.class);
Query query = null;

try {
    query = jqlQueryParser.parseQuery(JQL);
} catch (JqlParseException e) {
    throw new IllegalArgumentException("Bad JQL: " + jql);
}

def valueExtractor = chartBuilder.getValueExtractor(Value);
if ( valueExtractor == null ) {
    throw new IllegalArgumentException("No value extractor implementation for " + Value);
}

def groupValueExtractorX = chartBuilder.getGrouper(GroupByX);
def groupValueExtractorY = chartBuilder.getGrouper(GroupByY);

Field documentField;
try
{
    documentField = DocumentIssueImpl.class.getDeclaredField("document");
    documentField.setAccessible(true);
    for ( Issue issue : chartBuilder.getFilterUtils().performSearch(query, user) )
    {
        Document doc = (Document) documentField.get(issue);
        for ( String groupX : groupValueExtractorX.getGroups((Document) documentField.get(issue)) )
        {
            for ( String groupY : groupValueExtractorY.getGroups((Document) documentField.get(issue)) )
            {
                groupY = groupValueExtractorY.getResolvedValue(groupY, issue);
                groupX = groupValueExtractorX.getResolvedValue(groupX, issue);

                groupY = clear(groupY);
                groupX = clear(groupX);

                metaCountGroup.addValue(valueExtractor.get(issue, doc), groupY, groupX);
            }
        }
    }
} catch (Exception e){
    System.out.println("e " + e);
}

metaCountGroup.fillMissingValues();

def chartData = chartBuilder.newChartData(groupValueExtractorY.getGroupName());
chartData.setXType(groupValueExtractorX.getGroupName());
chartData.setType("table");
chartData.setLabels(true);
chartData.addCustomData("title", valueExtractor.getTitle());

chartBuilder.getChartUtil().transformResult(metaCountGroup, chartData, true);

valueExtractor.formatValues(chartData);

return chartData;

```

If you still have questions, feel free to refer to our [support team](#).