

Group By Report (multiple values)

This report sums different fields grouped in another certain field and all is displayed in a table.

Report preview

sectors	widely held stock	number of stocks	market capital
Banking	1,251.84	42,365,630,000	534,610,000,000
Car Manufacturing	1,859.83	26,767,740,000	984,950,000,000
Chemical Industry	1,046.49	11,366,700,000	136,710,000,000
Clothing / Apparel / Shoes	700.6	5,588,870,000	159,247,700,000

Parameters

Parameter		Type	Default value
UI	Code		
JQL	JQL	JQL Autocomplete	
Group By X	GroupByX	Group By Picker	Project
Accumulated fields	Values	Value Picker (multiple values)	

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);
}

Iterator<String> i = Values.iterator();
Set<String> fieldsIds = new HashSet<>();
while ( i.hasNext() )
{
    String key = i.next();
    if ( fieldsIds.contains(key) ) {
        i.remove();
    }
    else {
        fieldsIds.add(key);
    }
}

def groupValueExtractorX = chartBuilder.getGrouper(GroupByX);
Map<String, String> keysToResolvedValues = new LinkedHashMap<String, String>();

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)) )
        {
            groupX = groupValueExtractorX.getResolvedValue(groupX, issue);
            groupX = clear(groupX);

            for ( String key : fieldsIds )
            {
                def valueExtractor = chartBuilder.getValueExtractor(key);
                keysToResolvedValues.put(key, valueExtractor.getTitle());
                BigDecimal yValue = valueExtractor.get(issue, doc);
                metaCountGroup.addValue(yValue, groupX, key);
            }
        }
    }
} catch (Exception e){
    System.out.println("e " + e);
}

metaCountGroup.fillMissingValues();

def chartData = chartBuilder.newChartData();
chartData.setXType(groupValueExtractorX.getGroupName());
chartData.setType("table");
chartData.setLabels(true);

chartBuilder.getChartUtil().transformResultMultiValue(metaCountGroup, chartData, true, keysToResolvedValues);

return chartData;

```