

Simple Table Report

In this example, we display the number of created issues grouped by Issue Type on the X-Axis, and by Period on Y-axis.

The report looks like this, where both the **Jql_Param** and **Period_Picker** are dynamically populated.

[approve Download Scripted Chart Bundle](#)

Chart preview

Jql_Param

project = Testing

Period_Picker

Day

Week

Month

Quarter

Half year

Year

Show chart

Issues \ Project	2015 - Q2	2015 - Q3	2015 - Q4	2016 - Q1	2016 - Q2	2016 - Q3	2016 - Q4	2017 - Q1	2017 - Q2
Issue Type 2	91	102	103	72	81	79	96	80	90
Issue Type 3	81	92	103	81	91	82	91	91	77
Issue Type 4	61	100	97	95	86	103	74	94	78
Sub-task	162	191	176	247	218	220	217	221	227
Task	59	90	96	74	92	91	98	76	74

The period is a helper [xCharts](#) class, that handles time periods and can be found in the [API Documentation](#).

but the Period can be also dynamically set, by using a parameter:

```
Period selectedPeriod = Period.fromString(Period_Picker);
```

If Period_Picker is a **TEXT** Parameter, it has to have one of the following values:

- Day
- Week
- Month
- Quarter
- Half a year
- Year

An alternative way is to use the "Time Period Picker".

Parameters

Name	Type	Default value
Jql_Param	JQL Autocomplete	
Period_Picker	Time Period Picker	Month

Layout Script

Used layout: [DefaultTable](#).

```
var div = $("#chart");
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; });
$('thead > tr > th:first').text(chartData.ytype + " \\ " + 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.math.BigDecimal;
import java.text.DateFormat;
import java.util.Calendar;

import com.atlassian.jira.component.ComponentAccessor;
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.query.Query;
import com.decadis.jira.xchart.api.model.Period;
import com.decadis.jira.xchart.api.util.DateUtils;

def metaCountGroup = chartBuilder.newDataCollector();
DateFormat dateFormat = DateUtils.SimpleDateFormat;

JqlQueryParser jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser.class);
Query query = jqlQueryParser.parseQuery(Jql_Param);
Period selectedPeriod = Period.fromString(Period_Picker);

// Attention: Security is ignored
for ( Issue issue : chartBuilder.getFilterUtils().performSearchOverrideSecurity(query) )
{
    Calendar cwCreated = dateUtils.getStartOfPeriod(issue.getCreated(), selectedPeriod);
    String cw = dateUtils.getPeriodName(cwCreated.getTime(), selectedPeriod);
    metaCountGroup.addValue(BigDecimal.ONE, issue.getIssueType().getName(), cw);
}

metaCountGroup.fillMissingValues();

def chartData = chartBuilder.newChartData("Issues");
chartData.setXType("Project");
chartData.setType("table");

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

return chartData;

```

Example of resulting JSON object (chartData)

```

{
  "IssueType 1" :
  {
    "2016 - 01" : 2,
    "2016 - 02" : 2,
    "2016 - 03" : 2,
    "2016 - 04" : 1,
    "2016 - 05" : 1,
    "2016 - 06" : 1,
    "2016 - 07" : 0,
    "2016 - 08" : 0,
    "2016 - 09" : 1,
    "2016 - 10" : 2,
    "2016 - 11" : 1,
    "2016 - 12" : 2,
    "2017 - 01" : 0
  },
  "IssueType 2" :
  {
    "2016 - 01" : 1,
    "2016 - 02" : 3,
    "2016 - 03" : 0,
    "2016 - 04" : 2,
    "2016 - 05" : 1,

```

```

        "2016 - 06" : 1,
        "2016 - 07" : 2,
        "2016 - 08" : 0,
        "2016 - 09" : 2,
        "2016 - 10" : 1,
        "2016 - 11" : 2,
        "2016 - 12" : 1,
        "2017 - 01" : 1
    },
    "IssueType 3" :
    {
        "2016 - 01" : 1,
        "2016 - 02" : 0,
        "2016 - 03" : 3,
        "2016 - 04" : 1,
        "2016 - 05" : 3,
        "2016 - 06" : 1,
        "2016 - 07" : 2,
        "2016 - 08" : 2,
        "2016 - 09" : 0,
        "2016 - 10" : 2,
        "2016 - 11" : 1,
        "2016 - 12" : 1,
        "2017 - 01" : 0
    },
    "IssueType 4" :
    {
        "2016 - 01" : 1,
        "2016 - 02" : 3,
        "2016 - 03" : 2,
        "2016 - 04" : 0,
        "2016 - 05" : 2,
        "2016 - 06" : 3,
        "2016 - 07" : 4,
        "2016 - 08" : 3,
        "2016 - 09" : 3,
        "2016 - 10" : 0,
        "2016 - 11" : 0,
        "2016 - 12" : 2,
        "2017 - 01" : 0
    },
    "IssueType 5" :
    {
        "2016 - 01" : 1,
        "2016 - 02" : 0,
        "2016 - 03" : 1,
        "2016 - 04" : 5,
        "2016 - 05" : 1,
        "2016 - 06" : 2,
        "2016 - 07" : 1,
        "2016 - 08" : 3,
        "2016 - 09" : 2,
        "2016 - 10" : 4,
        "2016 - 11" : 4,
        "2016 - 12" : 3,
        "2017 - 01" : 1
    }
}

```



Related examples

Title

[Using Jira Software specific classes and Pickers in Scripts](#)

[Story status category grouped by Epic and custom value](#)

[Simple Timeseries Chart](#)

[Simple Table Report](#)

[Simple Scripting Example](#)

[Report - Lucene Group By](#)

[Open issues with average](#)

[Issues in specific status \(Period\)](#)

[Group ordering in scripted charts](#)

[Gantt Diagram](#)

[Customers in a Google Map](#)

[Created vs. resolved with trend](#)

[Comments count by user in JQL result](#)

[Block Search](#)

[2Y Axes Chart](#)

[Simple External Database Chart](#)

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