

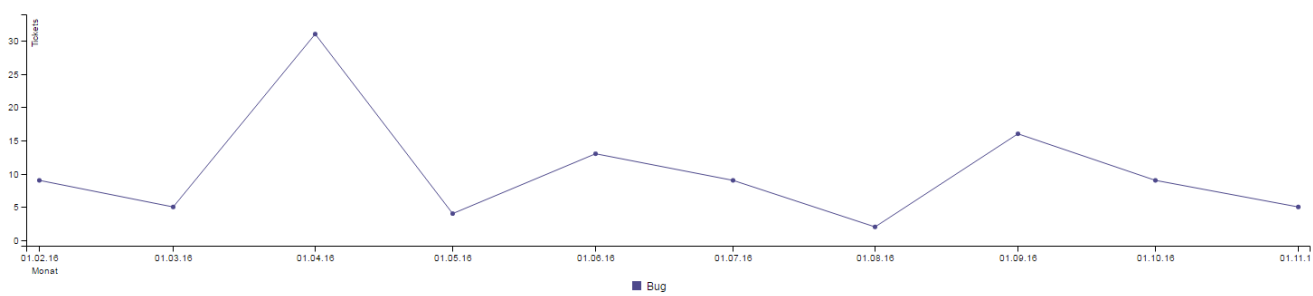
Simple Timeseries Chart

This example illustrates a simple use case: showing the number of certain issue types over a time period.

The resulting chart, when having a JQL Parameter that selects only Bug issue types, is the following:

[approve](#) [Download Scripted Chart Bundle](#)

Chart preview



Parameters

Name	Type
JQL	JQL autocomplete

Layout Script

used layout: [Default Timeseries](#).

c3.js Example

```
c3.generate({
  data: chartData,
  axis: {
    x: {
      type: 'timeseries',
      label: {
        text: 'Months',
        position: 'outer-left'
      },
      tick: {
        format: '%d.%m.%y',
        culling: {
          max: 25
        },
        fit: true,
        multiline: false
      }
    },
    y: {
      label: 'Tickets'
    }
  }
})

// chartData - this variable contains the aggregated data for display and is generated by the script
```

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.JqlParseException;
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;
import com.decadis.jira.xchart.api.model.ChartData;

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

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

try
{
    query = jqlQueryParser.parseQuery(JQL); //JQL is a parameter of type Jql Autocomplete Picker
} catch (JqlParseException e)
{
    throw new IllegalArgumentException("Bad JQL: " + query);
}

Period selectedPeriod = Period.MONTH;

for ( Issue issue : chartBuilder.getFilterUtils().performSearchOverrideSecurity(query) )
{
    Calendar cwCreated = dateUtils.getStartOfPeriod(issue.getCreated(), selectedPeriod);
    String cw = dateFormat.format(cwCreated.getTime());
    metaCountGroup.addValue(BigDecimal.ONE, issue.getIssueType().getName(), cw);
}

ChartData chartData = chartBuilder.newChartData("Issues");
chartData.setxFormat("%Y.%m.%d");
chartData.setType("line");

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

return chartData;

```

Example of resulting JSON object (chartData)

```

{
  "colors": {
    "Bug": "#4e498c"
  },
  "xs": {
    "Bug": "Bug-x"
  },
  "empty": {
    "label": {
      "text": "No data."
    }
  },
  "custom": {
  },
  "columns": [
    [
      "Bug", 9, 5, 31, 4, 13, 9, 2, 16, 9, 5
    ],
    [
      "Bug-x", "2016.02.01", "2016.03.01", "2016.04.01", "2016.05.01",
      "2016.06.01", "2016.07.01", "2016.08.01", "2016.09.01", "2016.10.01", "2016.11.01"
    ]
  ],
  "groups": [
    [
      "Bug"
    ]
  ],
  "hide": [
  ],
  "type": "line",
  "xFormat": "%Y.%m.%d",
  "order": null,
  "labels": null,
  "ytype": "Issues",
  "xtype": null
}

```



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.