Issues in specific status (Period)

This chart will display the number of issues that were in a specific status in a specific period, selected with a custom JQL and a group picker.

approve Download Scripted Chart Bundle

Chart preview

Time	Status: In Progress
2017 - 11	16
2017 - 12	16
2018 - 01	17
2018 - 02	15
2018 - 03	16
2018 - 04	16
2018 - 05	16
2018 - 06	3

Parameters

Name	Type	Default
Jql_Param	JQL autocomplete	
Period_Param	time Period Picker	Month
Periods_Param	Text	
Group_Param	Group Picker (single group)	
Satus_Param	Text	

Layout Script

JavaScript Template

```
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("#chart").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 )
    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;
    });
}


updateFrameHeight()
```

Data Script

Groovy Script

```
// This script has five parameters

// Jql_Param - to choose the issues (can use the JQL Picker)
// Period_Param - to choose the time period ("day", "week", "month", "quarter", "halfyear" or "year")
// Periods_Param - the number of past periods to show
// Group_Param - the name of the group to compute the average (issues/groupmembers)
// Status_Param - the ID of the status, which an issue was in a period to count

import java.math.BigDecimal;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;

import org.apache.log4j.Logger;

import com.atlassian.jira.component.ComponentAccessor;
import com.atlassian.jira.config.StatusManager;
```

```
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.issue.changehistory.ChangeHistoryManager;
import com.atlassian.jira.issue.history.ChangeItemBean;
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.util.DateUtils;
import com.decadis.jira.xchart.api.model.Period;



Logger logger = Logger.getLogger("Issues in specific status in specific period - xChart");

def dateUtil = chartBuilder.getDateUtils();
final Date now = new Date();

// evaluate the period parameter, should be "day", "week", "month", "quarter", "halfyear" or "year"
Period period = Period.fromString(Period_Param);
def countGroup = chartBuilder.newDataCollector();


// Helper class to check if one period of time overlaps another
public class Timespan {
    final static Date now = new Date();

    // Initializer for fix timespans, e.g. last month => period = MONTH, offset = -1
    public Timespan(DateUtils dateUtil, Period period, int offset) {
        start = dateUtil.getStartOfPeriodN(now, period, offset).getTime();
        end = dateUtil.getStartOfPeriodN(now, period, offset + 1).getTime();
    }

    public Timespan(Date start, Date end) {
        this.start = start;
        this.end = end;
    }

    public Date start;
    public Date end;

    // check if timespan overlaps
    public boolean overlap(Timespan other) {
        return start.compareTo(other.end) <= 0 && other.start.compareTo(end) <= 0;
    }
}

// number of periods
final int nrIntervals = Integer.valueOf(Periods_Param);

Timespan[] intervals = new Timespan[nrIntervals];

def dateFormat = dateUtil.getDateformat("yyyy - MM");

def status = ComponentAccessor.getComponent(StatusManager.class).getStatus(Status_Param);
def statusName = status == null ? "none" : status.getNameTranslation();


// initialize the count group
for ( int i = 0; i < intervals.length; ++i ) {
    intervals[i] = new Timespan(dateUtil,period, i - intervals.length + 1);
    countGroup.addValue(BigDecimal.ZERO, dateFormat.format(intervals[i].start), "Status: " + statusName);
}


// JQL param to choose issues
String jql = Jql_Param

logger.debug("JQL=" + jql);

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

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

// for evaluation of issue history
ChangeHistoryManager changeHistoryManager = ComponentAccessor.getChangeHistoryManager();

// to get the average. issue in progress count divided by the number of group members
def users = (double)ComponentAccessor.getGroupManager().getUsersInGroupCount(Group_Param);
def v = BigDecimal.valueOf(users != 0.0 ? 1.0/users : 0.0);

logger.info("users in group " + Group_Param + " = " + users)

// helper array to count issues only once in a period
boolean[] values = new boolean[intervals.length];

// reset the helper array
for(int i = 0; i<values.length; ++i) {
    values[i] = false;
}

// iterate over the issues
for ( Issue issue : chartBuilder.getFilterUtils().performSearch(query, user) ) {

    Date since = null;
    Date until = now;

    // iterate over the issue change history of the status field
    for ( ChangeItemBean item : changeHistoryManager.getChangeItemsForField(issue, "status") ) {

        // when the status changes from the SOI to another we have to look which periods machtes...
        if ( item.getFrom().equals(Status_Param) && !item.getTo().equals(Status_Param) ) {

            // special case if the SOI is the initial status of an issue
            if ( since == null ) {
                since = issue.getCreated();
            }

            until = item.getCreated();

            // search overlapping periods
            for ( int i = 0; i<values.length; ++i ) {
                if ( intervals[i].overlap(new Timespan(since, until)) ) {
                    values[i] = true;
                }
            }
            // issue is no longer in SOI so reset since timestamp
            since = null;
        }
        // when issue changes from another state to SOI, the period begins
        if ( item.getTo().equals(Status_Param) && !item.getFrom().equals(Status_Param) ) {
            since = item.getCreated();
        }
    }

    // special case if issue is still in SOI
    if ( since != null || issue.getStatusId().equals(Status_Param) ) {
        logger.debug("status = " + issue.getStatusId())
        if ( since == null ) {
            since = issue.getCreated();
        }
        for ( int i = 0; i<values.length; ++i ) {
            if ( intervals[i].overlap(new Timespan(since, now)) ) {
                values[i] = true;
            }
        }
    }
```

```
    // add the pre-computed value to the periods in which the issue was in SOI
    for(int i = 0; i<values.length; ++i) {
        if(values[i]) {
            countGroup.addValue(v, dateUtil.getPeriodName(intervals[i].start,period), "Status: " +
statusName);
            values[i]=false
        }
    }
}

def chartData = chartBuilder.newChartData("Time");

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

return chartData
```

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.