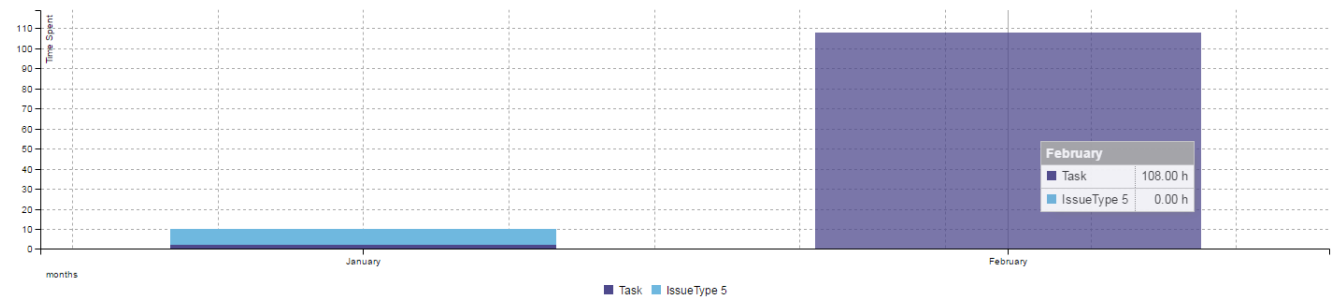


Worklog chart (Grouped by issue attribute)

The **worklog chart** shows the accumulated time spent, grouped by chosen fields.

Chart preview



Parameters

Parameter		Type	Default value
UI	Code		
JQL	JQL	JQL Autocomplete	
Chart Type	ChartType	Chart Type Picker	Bar
Group By X	GroupByX	Group By Picker	Project
Group By	GroupBy	Group By Picker	Issue Type

Layout Script

Used layout: [Easy Group By](#).

```

function formatTooltipAsHours(value, ratio, id, index)
{
    return value.toFixed(2) + ' h';
}

function formatTooltipAsHoursWithDays(value, ratio, id, index)
{
    var hours = parseInt(value);
    var days = parseInt(hours / 24);
    value = value - days * 24;
    if (days > 0)
    {
        return days + 'd, ' + value.toFixed(2) + ' h';
    }
    return value.toFixed(2) + ' h';
}

var c3arg = {
    onrendered: updateFrameHeight,
    data: chartData,
    axis: {
        x: {
            type: 'category', // this is needed to load string x value
            label: {
                text: chartData.custom.xLabel,
                position: 'outer-left'
            }
        },
        y: {
            label: chartData.ytype
        }
    }
};

if (chartData.custom && chartData.custom.tooltip)
{
    var tooltipFunction = eval(chartData.custom.tooltip);
    c3arg.tooltip = {
        format: {
            value: tooltipFunction
        }
    };
}

c3.generate(c3arg);

```

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.Issue;
import com.atlassian.jira.issue.DocumentIssueImpl;
import com.atlassian.jira.issue.worklog.Worklog;
import com.atlassian.jira.issue.worklog.WorklogManager;
import com.atlassian.jira.jql.parser.JqlParseException;
import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.query.Query;
import com.atlassian.jira.util.I18nHelper;

import com.decadis.jira.xchart.api.ChartParam;

```

```

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

def il8n = ComponentAccessor.getJiraAuthenticationContext().getI18nHelper();
def wlm = ComponentAccessor.getWorklogManager();
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 groupValueExtractorX = chartBuilder.getGrouper(GroupByX);
def groupValueExtractor = chartBuilder.getGrouper(GroupBy);

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 ( Worklog wl : wlm.getByIssue(issue) )
        {
            BigDecimal value = BigDecimal.valueOf(wl.getTimeSpent()).divide(BigDecimal.valueOf(3600), 4,
            BigDecimal.ROUND_HALF_UP);
            for ( String group : groupValueExtractor.getGroups(doc) )
            {
                for ( String groupx : groupValueExtractorX.getGroups(doc) )
                {
                    String grpName = groupValueExtractorX.getResolvedValue(groupx, issue);
                    metaCountGroup.addValue(value, group, grpName);
                }
            }
        }
    }
} catch (Exception e){}

metaCountGroup.fillMissingValues();

def chartData = chartBuilder.newChartData(il8n.getText("common.concepts.time.spent"));

for ( String grpKey : metaCountGroup.keySet() ) {
    chartData.addGroupName(grpKey, groupValueExtractor.getResolvedValue(grpKey, null));
}

chartData.setXType(groupValueExtractorX.getGroupName());
chartData.setType(ChartType);
chartData.addCustomData("tooltip", "formatTooltipAsHours");

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

return chartData;

```

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