

# Logging PageEvents to Statsd (Datadog)

A possible way to Log User- and Page-Access to statsd is via the Event system - using [Adaptavist's Scriptrunner for Confluence](https://scriptrunner.adaptavist.com/latest/confluence/ConfluenceEventHandlers.html#_collecting_stats), se [https://scriptrunner.adaptavist.com/latest/confluence/ConfluenceEventHandlers.html#\\_collecting\\_stats](https://scriptrunner.adaptavist.com/latest/confluence/ConfluenceEventHandlers.html#_collecting_stats)

Read [Access Logging in Confluence](#) for good reasons to log via the Event system.



To make this work, the jar file from <https://github.com/datadog/java-dogstatsd-client#java-dogstatsd-client> must be placed under confluence /WEB-INF/lib and Confluence must be restarted. This will load the jar file into the Tomcat.

The script is executed by an Event Handler in Confluence:

- Custom event handler**  
Run your own groovy scripts in response to events

Note

An optional note, used only for your reference.

Events

PageViewEvent ×

Select the event(s) this code will handle

Script file

post.groovy

Path to the script file accessible on the server

Inline script

1

Enter your script here

?

→

↶

Update

Cancel

this executes this script for every PageViewEvent:

## post.groovy

```
import com.atlassian.confluence.spaces.SpaceManager
import com.atlassian.sal.api.component.ComponentLocator
import com.atlassian.confluence.event.events.content.page.*
import groovy.transform.Field
import com.timgroup.statsd.StatsDClient;
import com.timgroup.statsd.NonBlockingStatsDClient;

def spaceManager = ComponentLocator.getComponent(SpaceManager)
def pageManager = ComponentLocator.getComponent(PageManager)

String userName="Anonymous"
def currentUser = AuthenticatedUserThreadLocal.get()
if (currentUser)
{
    userName=(String)currentUser.name
}

//System.out.println("Start...")

def event = event as PageEvent

def spaceKey = event.page.spaceKey
def pageId = event.page.id as String

//System.out.println("Posting.....")

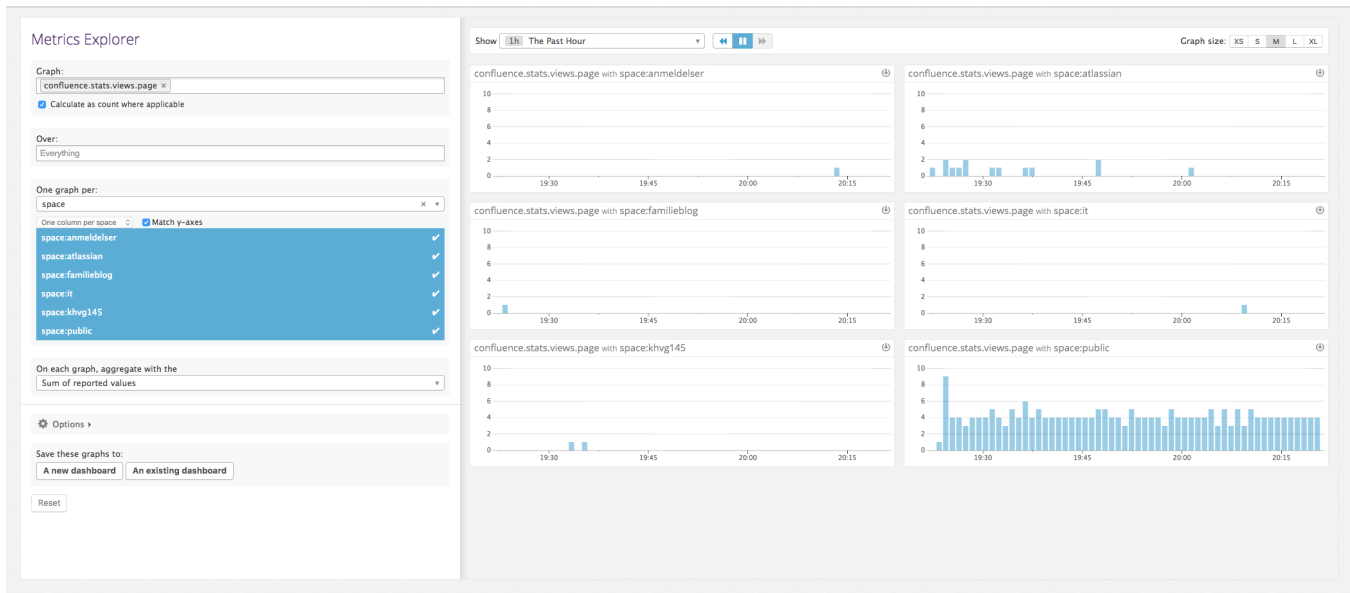
String[] tags = ["user:${userName}", "space:${spaceKey}", "user:${pageId}"]

StatsDClient statsdpage = new NonBlockingStatsDClient("confluence.stats.views","localhost",8125,tags);
statsdpage.incrementCounter("page");

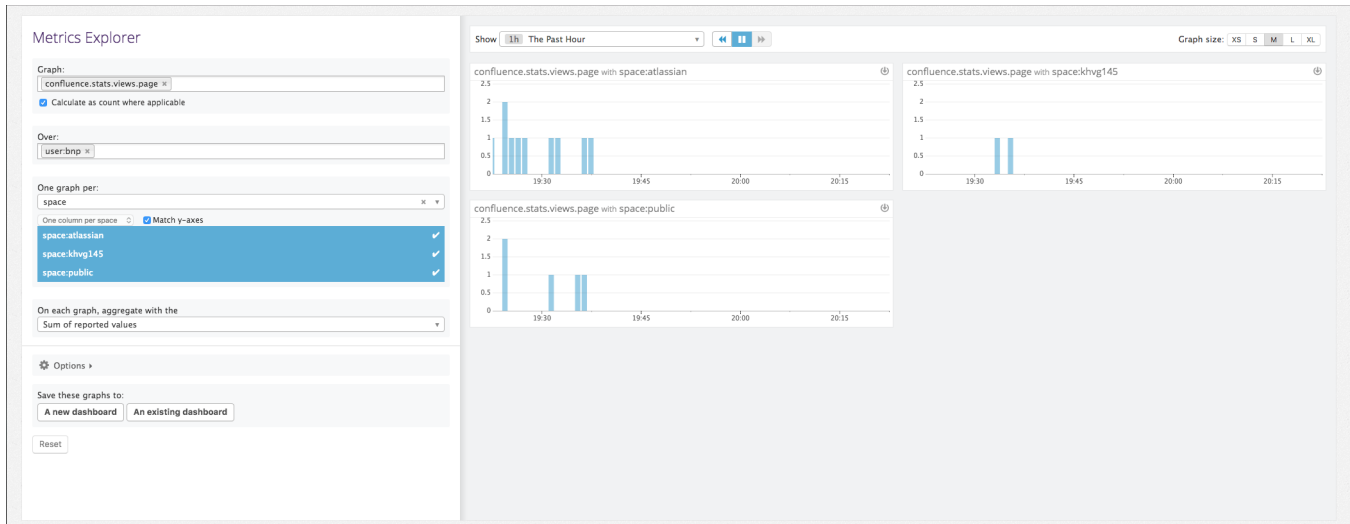
try
{
    //System.out.println("Closing socket");
    statsdpage.stop();
}
catch(Exception ex) {
    //System.out.println("Catching the exception");
}

//System.out.println("End.....")
```

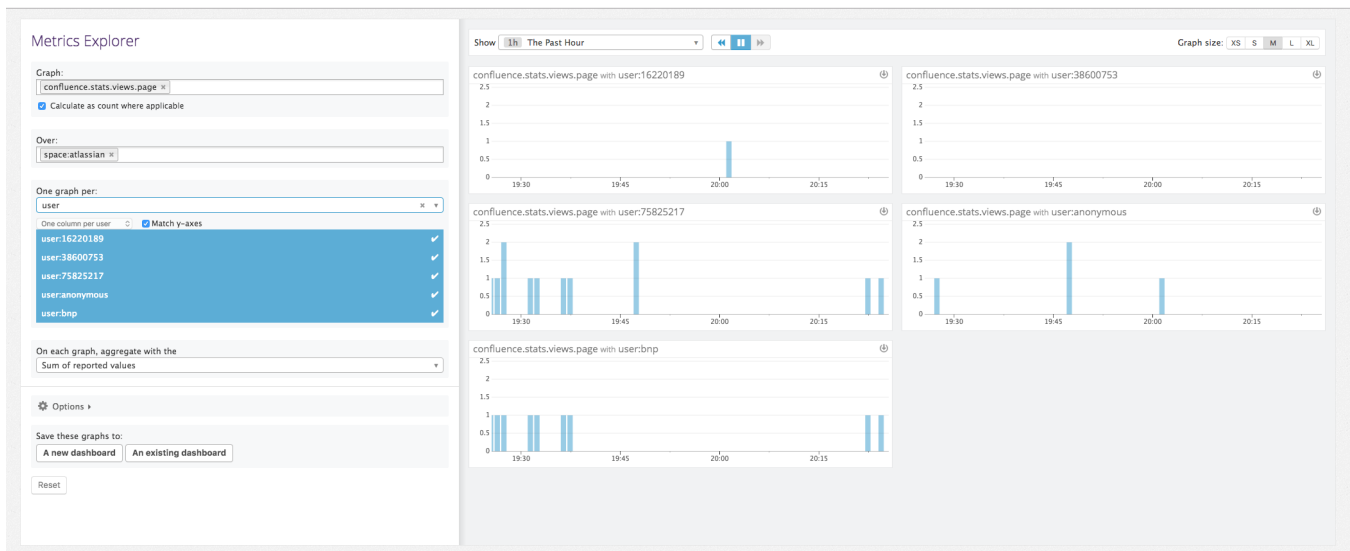
Viewing the data in DataDog - here its all Page Views pr. Space:



Or tracking a single User pr. Space:



Finally - vice versa - Monitoring a single Space for PageViews pr. User:





For some reason, this kills the Confluence Tomcat with "java.io.IOException: Too many open files", so my best guess is that the script is not releasing some resources. Raising limits has not helped...

```
12-Mar-2017 10:39:30.087 SEVERE [http-nio-8090-Acceptor-0] org.apache.tomcat.util.net.
NioEndpoint$Acceptor.run Socket accept failed
java.io.IOException: Too many open files
    at sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method)
    at sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:422)
    at sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:250)
    at org.apache.tomcat.util.net.NioEndpoint$Acceptor.run(NioEndpoint.java:682)
    at java.lang.Thread.run(Thread.java:745)
```