Logging PageEvents to Elasticsearch

A possible way to Log User- and Page-Access in Confluence is via the Event system - using Adaptavist's Scriptrunner for Confluence.

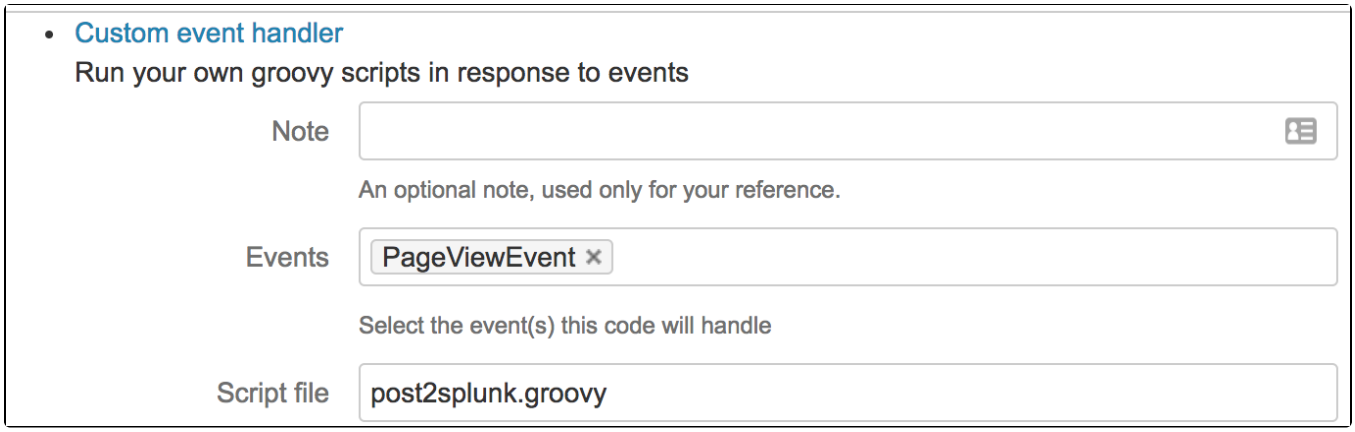
This ways has Pros and Cons - read Access Logging in Confluence. On Pro is that the POST to Splunk in in the backend; so we dont need to open for the receiving system in the Firewall

> My site is mainly external as a website, with only one internal user, myself "bnp". In that situation, the PageViewEvent is not so interesting as if this was an internal system with multiple users.
>
> Currently, I have found no way to correlate bot/spider/monitoring hits from the real PageViews.
>
> Also, PageViewEvents only occur when a page is rendered and this gives back HTTP Code "200 OK" to the client. See Different Loggings for different logging compares.

We do POST a json like this to Elasticsearch at URL http://elkserver1:9200/webaccess/pageevent/

This will create an index named "webacecss" and give out data the type "pageevent"

```
{
    "timestamp": 1426279439,
    "event-type": "PageView",
    "space-key": "IT",
    "confluence-page-title": "Atlassian Home",
    "confluence-page-id": "199002",
    "username": "bnp"
}
```



this executes this script for every PageViewEvent:

```
import com.atlassian.confluence.user.AuthenticatedUserThreadLocal
import com.atlassian.confluence.user.*;
import java.net.URL;
import java.net.URLEncoder;
import java.net.MalformedURLException;
import java.io.UnsupportedEncodingException;
import com.atlassian.confluence.pages.Page
import com.atlassian.confluence.pages.PageManager
import com.atlassian.confluence.spaces.Space
import com.atlassian.confluence.spaces.SpaceManager
import com.atlassian.sal.api.component.ComponentLocator
import com.atlassian.confluence.event.events.content.page.*

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

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

```
}

def event = event as PageEvent
String eventType=(String)event.toString()
eventType=eventType.replaceAll("com.atlassian.confluence.event.events.content.page.","")
eventType=eventType.substring(0, eventType.indexOf('@'))
eventType=eventType.replaceAll("Event","")

// keys to create unique nodes for counters
// https://docs.atlassian.com/confluence/5.9.7/com/atlassian/confluence/pages/Page.html

String spaceKey = event.page.getSpace().getKey()
String pageId = event.page.getIdAsString()
String pageName = event.page.getTitle()

def requestMethod = "GET";
def URLParam = []
def baseURL = "http://elkserver1:9200/webaccess/pageevent/"

def url = new java.net.URL(baseURL);
URLConnection connection = url.openConnection();
connection.setRequestMethod(requestMethod);
connection.doOutput = true
connection.setUseCaches(false);
connection.setRequestProperty("Content-Type", "application/json;charset=UTF-8");

def dateTime = new Date()

String jSon= "{"
jSon = jSon + "\"timestamp\":\"" + dateTime.toString() + "\","
jSon = jSon + "\"event-type\":\"" + eventType  + "\","
jSon = jSon + "\"space-key\":\"" + spaceKey + "\","
jSon = jSon + "\"confluence-page-title\":\"" + pageName + "\","
jSon = jSon + "\"confluence-page-id\":\"" + pageId + "\","
jSon = jSon + "\"username\":\"" + userName + "\""
jSon = jSon + "}"

def writer = new OutputStreamWriter(connection.outputStream)
writer.write(jSon)
writer.flush()
writer.close()
connection.connect();
try
{
  connection.getContent()
}
catch (all)
{
}
String Status=connection.getResponseCode()
String Message=connection.getResponseMessage()
```

Currently I can search the data in Elasticsearch, due to a problem with the timestamp and mapping. It seems the Timestamp is not searchable/aggregatable ... a Mapping issue