ScriptRunner per Confluence – Diamo una occhiata

Estendiamo le funzionalità di Confluence

In questo post andremo a dare una prima occhiata a questo nuovo Addon, che estende le funzionalità di Confluence. Abbiamo già avuto modo di conoscere questo addon per JIRA, dove abbiamo mostrato le potenzialità.

Vediamo subito che cosa offre

Questo addon è una piacevole sorpresa di AtlasCamp. Non sono riuscito a partecipare, ma spero di essere ben presente al prossimo :-D.

Abbiamo la possibilità di poter aggiungere degli script personalizzati, esattamente come descritto per JIRA, per aggiungere azioni o macro che non sono presenti.

Possiamo quindi aggiungere delle funzioni personalizzate o sfruttare la completa libreria di script, che la Adaptavist mette a disposizione, per poter creare i nostri script

Conclusioni

Abbiamo fatto la conoscenza di un addon molto interessante. Nei prossimi post andremo a testare e ben provare per saggiarne le caratteristiche.

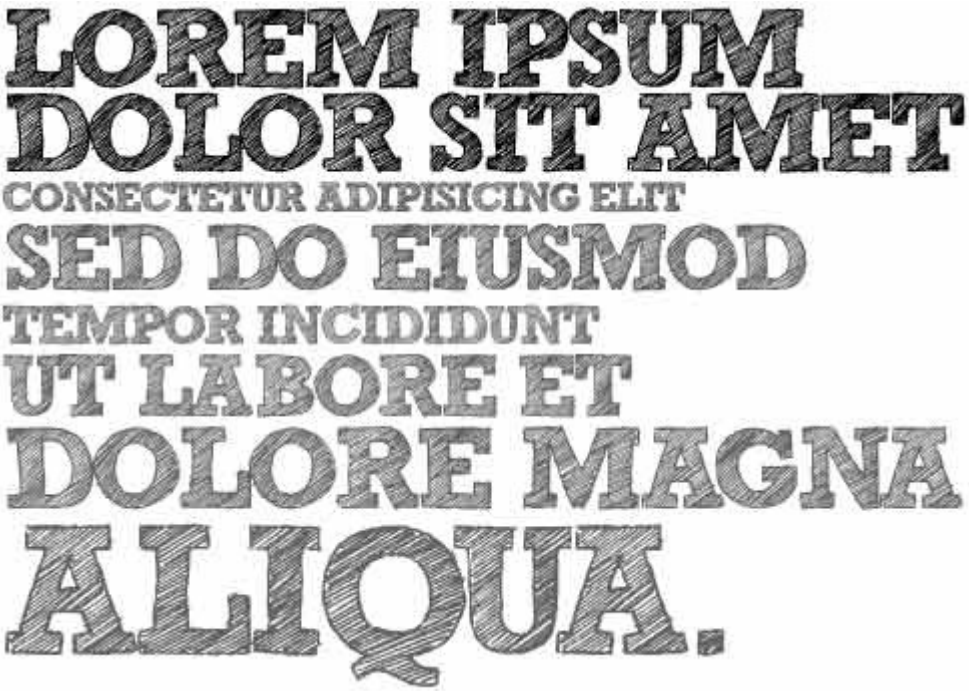
Reference

Maggiori informazioni sono presenti qui.

User Macro – Lorem Ipsum: Facciamo un nostro esempio.

Un altro esempio

In questo post, proseguiamo la serie di articoli dedicati alla creazione di User Macro. Vedremo un esempio di come creare una semplice macro che sostituisca la macro ufficiale Lorem Ipsum.



Andiamo sul pratico

Iniziamo con il definire la nostra macro di prova. Come prima cosa andiamo a definire che cosa deve fare la nostra macro . Non è cosa da poco.

Vogliamo sostituire la macro standard Lorem Ipsum di Confluence. Vogliamo implementarne una nostra che restituisca un nostro risultato personalizzato. Nel nostro caso abbiamo bisogno di ripetere una scritta solo fino a 5 volte massimo.

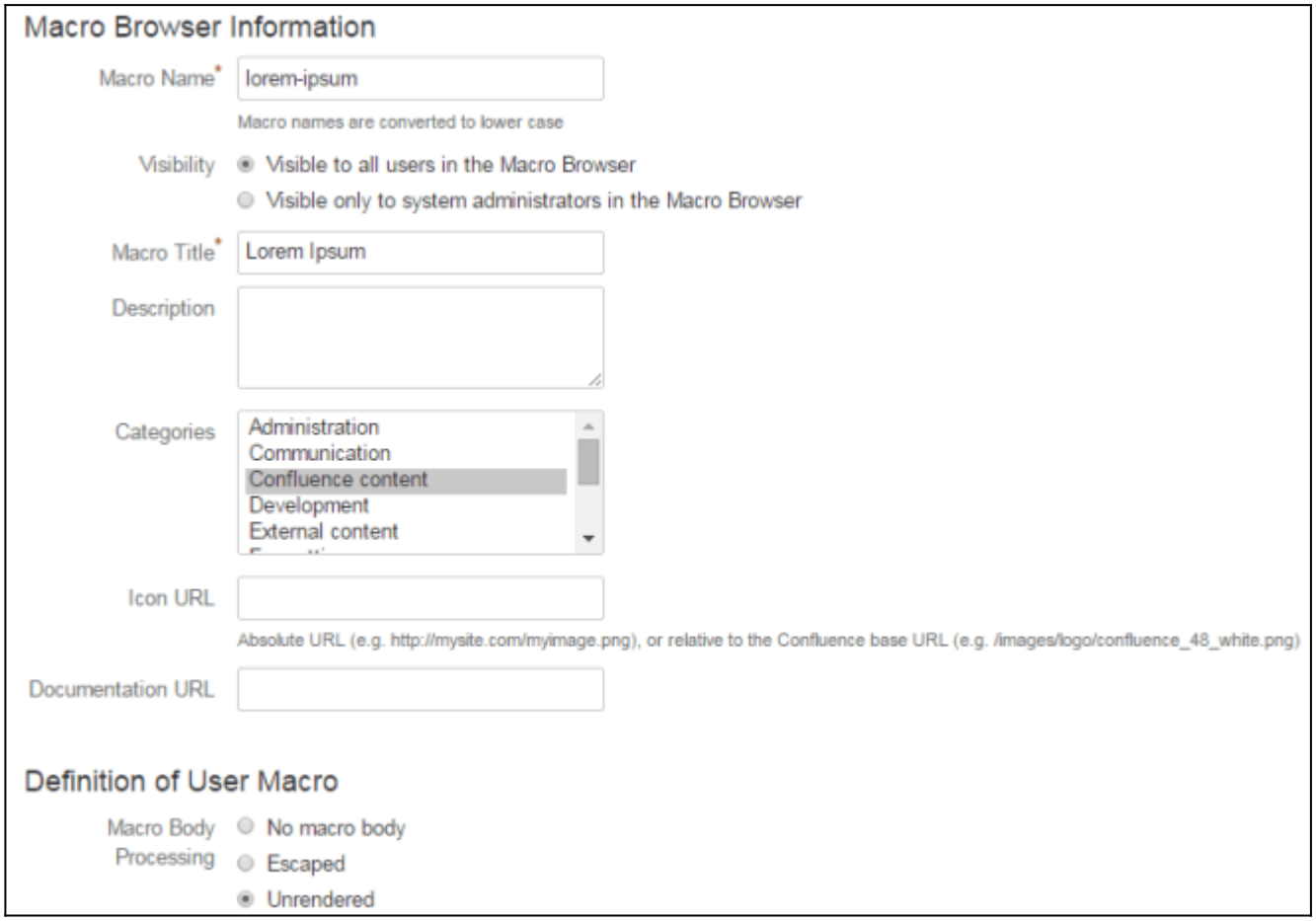
Il nostro obbiettivo è quello di avere delle sezioni si Lorem

Ipsum molto ridotte come testo.

Il nostro script di macro

Definiamo il nostro script come segue:

Come parametri principali, fate riferimento alla seguente immagine:



Il codice della macro è il seguente

```
## Macro title: Lorem Ipsum
## Macro has a body: N
## Body processing: Genera un testo a misura del Lorem Ipsum
## Output: Testo indicato
##
## Developed by: ArtigianoDelSoftware di Fabio Genovese
```

```
## Date created: 05/09/2015
## Installed by: -

## Parametri
##        @param        numpar:title=Number        of
paragraph|type=enum|enumValues=1,2,3,4,5|required=true|desc=N
umber of paragraph

## Corpo della macro
#set ( $Testo = "lorem ipsum, quia dolor sit, amet,
consectetur, adipisci velit, sed quia non numquam eius" )
#set ( $Testodef = "")
#set ( $limit = $paramnumpar )

#if ( $paramnumpar == 1 )
 #set ( $Testodef = "$Testo" )
#elseif ( $paramnumpar == 2 )
 #set ( $Testodef = "$Testo$Testo" )
#elseif ( $paramnumpar == 3 )
 #set ( $Testodef = "$Testo$Testo$Testo" )
#elseif ( $paramnumpar == 4 )
 #set ( $Testodef = "$Testo$Testo$Testo$Testo" )
#else
 #set ( $Testodef = "$Testo$Testo$Testo$Testo$Testo" )
#end

## Visualizzazione
<html>
<body>
$Testodef
</body>
<html>
```

Identifichiamo le seguenti sezioni:

- ***Macro title*** – Si tratta di un commento da inserire in cima alla macro
- ***Parametri*** – dichiarazione dei parametri
- ***Corpo della macro*** – Codice della macro
- ***Visualizzazione*** – Parte del corpo della macro che visualizza i risultati

Macro Title

Si tratta del mero commento alla procedura. Lo consiglio sempre. Inseritelo. Non occupa alcuno spazio ed è utile quando riprendete in mano la macro per farne manutenzione

```
## Parametri
##        @param        numpar:title=Number        of paragraph|type=enum|enumValues=1,2,3,4,5|required=true|desc=Number of paragraph
```

Parametri

Dichiarazione dei parametri che richiede la macro. Parte molto importante. Nel nostro caso abbiamo inserito una selezione automatica in cui l'utente può selezionare un valore da 1 a 5. Un semplice enumeratore.

```
## Parametri
##        @param        numpar:title=Number        of paragraph|type=enum|enumValues=1,2,3,4,5|required=true|desc=Number of paragraph
```

Corpo della macro

Di seguito il codice, scritto in sintassi Velocity, che realizza il tutto.

```
## Corpo della macro
#set ( $Testo = "lorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius" )
#set ( $Testodef = "")
#set ( $limit = $paramnumpar )

 #if ( $paramnumpar == 1 )
  #set ( $Testodef = "$Testo" )
 #elseif ( $paramnumpar == 2 )
  #set ( $Testodef = "$Testo$Testo" )
```

```
#elseif ( $paramnumpar == 3 )
 #set ( $Testodef = "$Testo$Testo$Testo" )
#elseif ( $paramnumpar == 4 )
 #set ( $Testodef = "$Testo$Testo$Testo$Testo" )
#else
 #set ( $Testodef = "$Testo$Testo$Testo$Testo$Testo" )
#end
```

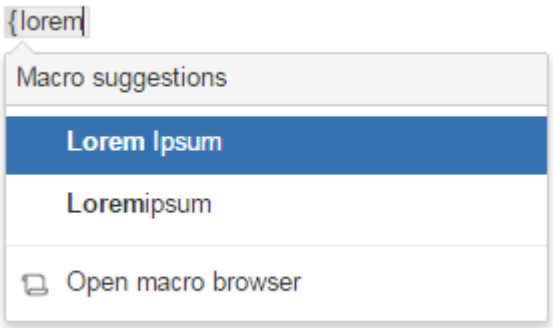
Visualizzazione

La parte che si occupa della visualizzazione è molto semplice e non richiede alcun commento.

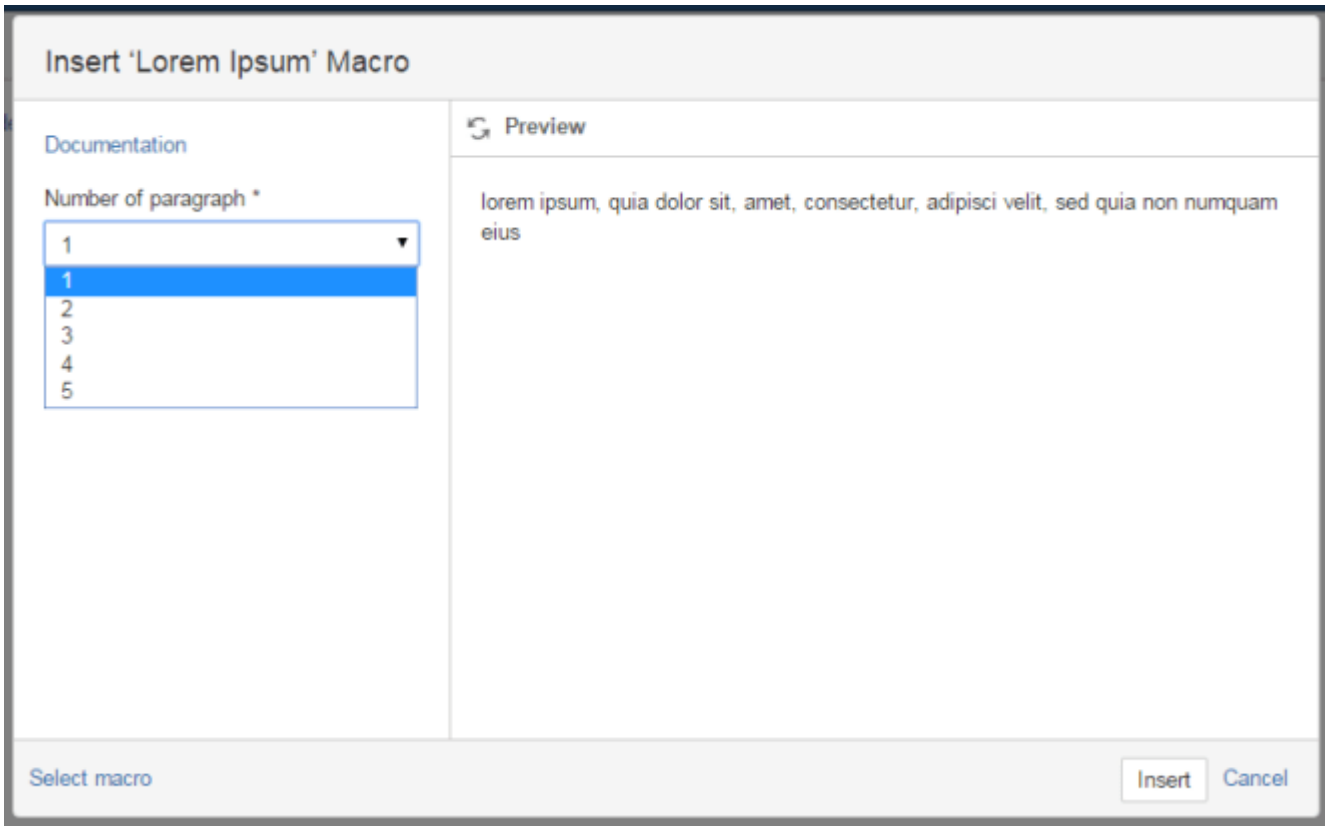
```
## Visualizzazione
<html>
<body>
$Testodef
</body>
<html>
```

Risultati?

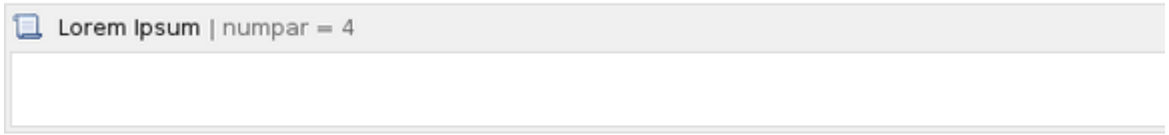
I risultati sono i seguenti. Alla chiamata vediamo che Confluence mostra anche la nostra Macro (la prima).



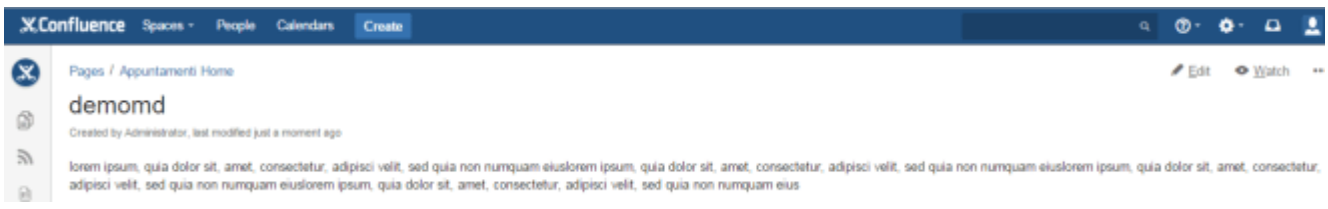
Una volta selezionata la macro, viene attivata l'autocomposizione



Una volta impostato la macro viene visualizzata sulla pagina



Il risultato finale è il seguente:



Conclusioni

Abbiamo visto come si può realizzare una semplice macro. Nei prossimi post andremo ad esaminare altri esempi.

User Macro – Un semplice esempio di utilizzo

User Macro

In questo post andremo ad consultare un articolo presente in questo blog, che ritengo molto interessante. Lo esamineremo e cercheremo di approfondire l'argomento, in quanto lo ritengo un ottimo esempio didattico.

Andiamo in dettaglio

L'articolo del blog spiega come realizzare una alternativa alla macro ***Space Detail Macro***, usando le macro per ottenere un risultato migliore

In particolare il risultato che offre con la sua macro è il seguente:

L'ho scelto perché si tratta di una macro molto semplice e, dal punto di vista didattico, la macro risulta realizzata molto bene.

Vediamo il codice

La configurazione preliminare della macro è molto semplice. La seguente immagine la riassume:

Il codice è molto semplice. Lo riporto di seguito nella sua interessa e successivamente lo andremo ad esaminare pezzo per pezzo, in modo da capire come viene eseguita ogni singola operazione.

```
## Macro title: Space Meta Data
## Macro has a body: Y or N (N)
## Body processing: Selected body processing option
## Output: Selected output option
##
## Developed by: Andrew Frayling
## Date created: 03/05/2012
## Installed by: <your name>
```

```
## Macro to display information such as number of pages,
number of blog posts, attachment size, etc. about a Space.

## @noparams

## Get space details
#set ( $spaceName = $space.getName() )
#set ( $spaceKey = $space.getKey() )
#set ( $spaceHome = $space.getHomePage() )
#set ( $spaceCreator = $space.getCreatorName() )
#set ( $spaceCreationDate = $space.getCreationDate() )
#set ( $spaceDescription = $space.getDescription() )
#set ( $pageCount = $spaceManager.findPageTotal($space) )
#set ( $blogCount =
$spaceManager.getNumberOfBlogPosts($space) )

## Get all pages in the current Space
#set ( $allPagesInSpace = $pageManager.getPages($space, true)
)

## Reset total attachment file size
#set ( $totalAttachmentFileSizeForSpace = 0 )

## Reset total number of attachments
#set ( $totalAttachmentCount = 0 )

## Loop through all pages in the current Space
#foreach ($page in $allPagesInSpace)
  ## reset the attachment count for each page
  #set ( $pageAttachmentCount = 0 )
  ## reset the attachment file size total for each page
  #set ( $totalFileSizePerPage = 0 )
  ## get the attachments for each page
  #set ( $allAttachments = $page.getAttachments() )
  ## Loop through each attachment
  #foreach ($attachment in $allAttachments)
    ## Increment the attachment count for the page
    #set ( $pageAttachmentCount = $pageAttachmentCount + 1 )
    ## Sum the size of the attachments on the page
     #set ( $totalFileSizePerPage = $totalFileSizePerPage +
$attachment.getFileSize() )
```

```
  #end
  ## End looping through attachments
  ## Increment total attachment count for the current Space
   #set ( $totalAttachmentCount = $totalAttachmentCount +
$pageAttachmentCount )
  ## Sum the total size of attachments for the current Space
     #set   (   $totalAttachmentFileSizeForSpace   =
$totalAttachmentFileSizeForSpace + $totalFileSizePerPage )
#end
## End looping through pages

## Convert attachment size to MBs
#set ( $attachmentSizeMb = ($totalAttachmentFileSizeForSpace
/ 1024.00) / 1024.00 )

## Display Space Details
<table class="confluenceTable">
  <tbody>
    <tr>
      <th class="confluenceTh">Name</th>
      <td class="confluenceTd">$spaceName</td>
    </tr>
    <tr>
      <th class="confluenceTh">Key</th>
      <td class="confluenceTd">$spaceKey</td>
    </tr>
    <tr>
      <th class="confluenceTh">Description</th>
                                                    <td
class="confluenceTd">$spaceDescription.getBodyAsString()</td>
    </tr>
    <tr>
      <th class="confluenceTh">Home Page</th>
       <td class="confluenceTd">#contentLink2($spaceHome true
false)</td>
    </tr>
    <tr>
      <th class="confluenceTh">Created By</th>
       <td class="confluenceTd">#usernameLink($spaceCreator)
($action.dateFormatter.formatDateTime($spaceCreationDate))</t
d>
```

```
    </tr>
    <tr>
      <th class="confluenceTh">Number of Pages</th>
      <td class="confluenceTd">$pageCount</td>
    </tr>
    <tr>
      <th class="confluenceTh">Number of Blog Posts</th>
      <td class="confluenceTd">$blogCount</td>
    </tr>
    <tr>
      <th class="confluenceTh">Number of Attachments</th>
         <td class="confluenceTd">$totalAttachmentCount
(including all versions)</td>
    </tr>
    <tr>
      <th class="confluenceTh">Total Size of Attachments</th>
      <td class="confluenceTd">$attachmentSizeMb MB</td>
    </tr>
  </tbody>
</table>
```

Distinguiamo le seguenti sezioni nella macro:

- *Testata*
- *Lettura parametri dello space*
- *Lettura della dimensione dei vari allegati*
- *Visualizzazione dati ottenuti*

Andiamo ad esaminarle in dettaglio.

```
## Macro title: Space Meta Data
## Macro has a body: Y or N (N)
## Body processing: Selected body processing option
## Output: Selected output option
##
## Developed by: Andrew Frayling
## Date created: 03/05/2012
## Installed by: <your name>
## Macro to display information such as number of pages,
number of blog posts, attachment size, etc. about a Space. ##
@noparams
```

L’Intestazione della macro è quello che avevo già descritto nel mio primo post. In questo caso, l’autore ha realizzato una intestazione molto ben fatta, chiara, semplice e di rapido effetto.

Spiega subito le caratteristiche della macro in poche righe.

```
## Get space details
#set ( $spaceName = $space.getName() )
#set ( $spaceKey = $space.getKey() )
#set ( $spaceHome = $space.getHomePage() )
#set ( $spaceCreator = $space.getCreatorName() )
#set ( $spaceCreationDate = $space.getCreationDate() )
#set ( $spaceDescription = $space.getDescription() )
#set ( $pageCount = $spaceManager.findPageTotal($space) )
#set ( $blogCount = $spaceManager.getNumberOfBlogPosts($space) )

## Get all pages in the current Space
#set ( $allPagesInSpace = $pageManager.getPages($space, true) )

## Reset total attachment file size
#set ( $totalAttachmentFileSizeForSpace = 0 )
```

In questa sezione, invece, va a leggere tutte le informazioni dello Space, impostando tutte le variabili con le informazioni che si vogliono visualizzare nella macro. Notiamo le variabili, definite con la sintassi che è stata già spiegata nei precedenti post.

```
## Reset total number of attachments
#set ( $totalAttachmentCount = 0 )

## Loop through all pages in the current Space
#foreach ($page in $allPagesInSpace)
  ## reset the attachment count for each page
  #set ( $pageAttachmentCount = 0 )
```

```
  ## reset the attachment file size total for each page
  #set ( $totalFileSizePerPage = 0 )
  ## get the attachments for each page
  #set ( $allAttachments = $page.getAttachments() )
  ## Loop through each attachment
  #foreach ($attachment in $allAttachments)
    ## Increment the attachment count for the page
    #set ( $pageAttachmentCount = $pageAttachmentCount + 1 )
    ## Sum the size of the attachments on the page
     #set ( $totalFileSizePerPage = $totalFileSizePerPage +
$attachment.getFileSize() )
  #end
  ## End looping through attachments
  ## Increment total attachment count for the current Space
   #set ( $totalAttachmentCount = $totalAttachmentCount +
$pageAttachmentCount )
  ## Sum the total size of attachments for the current Space
     #set   (   $totalAttachmentFileSizeForSpace   =
$totalAttachmentFileSizeForSpace + $totalFileSizePerPage )
#end
## End looping through pages

## Convert attachment size to MBs
#set ( $attachmentSizeMb = ($totalAttachmentFileSizeForSpace
/ 1024.00) / 1024.00 )
```

In questa sezione del codice, viene eseguito un esempio di ciclo sulle varie pagine, per determinare la dimensione dei vari allegati alle pagine dello Space. Questo è sicuramente un ottimo esempio che mostra come sono reperite le informazioni usando il codice delle macro.

In aggiunta vediamo come eseguire il reperimento dei dati, quali le pagine presenti dentro uno space, e degli allegati. La sintassi non è affatto difficile e consente di poter lavorare agevolmente.

```
## Display Space Details
<table class="confluenceTable">
  <tbody>
```

```
    <tr>
      <th class="confluenceTh">Name</th>
      <td class="confluenceTd">$spaceName</td>
    </tr>
    <tr>
      <th class="confluenceTh">Key</th>
      <td class="confluenceTd">$spaceKey</td>
    </tr>
    <tr>
      <th class="confluenceTh">Description</th>
                                                    <td
class="confluenceTd">$spaceDescription.getBodyAsString()</td>
    </tr>
    <tr>
      <th class="confluenceTh">Home Page</th>
       <td class="confluenceTd">#contentLink2($spaceHome true
false)</td>
    </tr>
    <tr>
      <th class="confluenceTh">Created By</th>
       <td class="confluenceTd">#usernameLink($spaceCreator)
($action.dateFormatter.formatDateTime($spaceCreationDate))</t
d>
    </tr>
    <tr>
      <th class="confluenceTh">Number of Pages</th>
      <td class="confluenceTd">$pageCount</td>
    </tr>
    <tr>
      <th class="confluenceTh">Number of Blog Posts</th>
      <td class="confluenceTd">$blogCount</td>
    </tr>
    <tr>
      <th class="confluenceTh">Number of Attachments</th>
         <td class="confluenceTd">$totalAttachmentCount
(including all versions)</td>
    </tr>
    <tr>
      <th class="confluenceTh">Total Size of Attachments</th>
      <td class="confluenceTd">$attachmentSizeMb MB</td>
    </tr>
```

```
    </tbody>
  </table>
```

Questa è la parte che si occupa della visualizzazione. Notiamo che si tratta di HTML, che sfrutta le classi e le proprietà di Confluence. Vediamo che tutti i risultati delle precedenti elaborazioni sono inserite in questa tabella, in maniera molto rapida.

Conclusioni

Abbiamo analizzato un esempio di macro, molto semplice ma nello stesso tempo, fondamentale. Con esso è possibile comprendere e capire meglio come si può realizzare una macro, come possiamo sfruttare i vari punti del linguaggio per realizzare le funzionalità di cui si abbisogna senza, in questo caso, dover installare nulla :-). Scusate se è poco

Ringraziamenti

Un ringraziamento particolare va all'autore del post blog. L'articolo è ben fatto e, anche se in inglese, è molto chiaro. I complimenti all'autore sono d'obbligo.

Le User macro – Queste sconosciute

Le User Macro

In questo post andremo ad affrontare un argomento un pò complesso: ***Le User Macro.*** Si tratta di un primo post in cui andremo ad introdurre l'argomento e che approfondiremo con ulteriori post.

Che cosa sono le User Macro?

Come prima cosa cerchiamo di dare una prima definizione delle User Macro :-).

Si tratta di azioni custom, che l'utente può definire all'interno di Confluence, per definire delle azioni mirate e

senza dover arrivare a sviluppare degli addon. Si tratta comunque di azioni limitate a specifici compiti, quali applicare delle formattazioni, etc.

Per poterle realizzare è necessario disporre di un minimo di conoscenza di sviluppo software, in quanto è necessario utilizzare delle istruzioni vere e proprie. Ma non spaventiamoci. Non è difficilissimo svilupparle.

Perchè usarle?

Il motivo è molto semplice. Disporre di un sistema atto a realizzare azioni custom, senza aver la necessità di dover creare un addon. Disporre di una semplice interfaccia, dove poter impostare un codice che esegue una semplice azione, di cui abbisognamo, è sicuramente utile. Non sfruttarlo non è proprio bello :-).

Come le definiamo?

La definizione di una User Macro viene eseguita nella sezione di amministrazione. Di conseguenza, occorre che l'utenza disponga dei privilegi per poter accedere a tale sezione.

Accedendo dal COG Menù, cui oramai siamo abituati

alla sezione ***User Macro*** della General ***Configuration***, possiamo iniziare a definire la nostra Macro.

Conclusioni

In questo primo articolo abbiamo presentato questo argomento. Nei prossimi post andremo a vedere come definire alcuni esempio di User Macro.