

FishEye – 2

Approfondiamo Fisheye

In questo post andremo a vedere il funzionamento di Fisheye, in modo da gestire un semplice repository di codice sorgente, proseguendo quanto già discusso nel precedente [post](#).



Di che cosa abbiamo bisogno?

Sostanzialmente, abbiamo bisogno delle seguenti componenti:

- Repository SVN
- FishEye
- JIRA



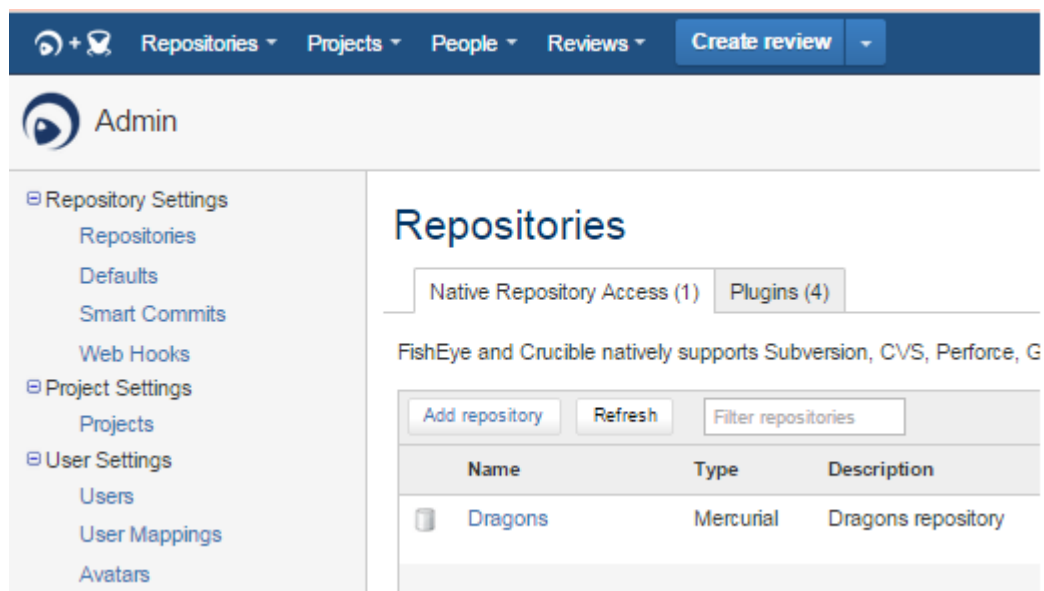
Iniziamo

Supponiamo che il sistema sia stato già configurato in modo da avere a disposizione **JIRA** come repository degli utenti di **FishEye**.

La prima operazione da eseguire, una volta che **FishEye** è stato

installato, è di eseguire le operazioni di Login come utente amministratore e di configurare un accesso ad un repository. Una volta eseguito l'accesso:

- Tramite **COG** menù, accedere alla sezione di amministrazione. Quindi selezionare, da **Repository Settings, Repositories**:

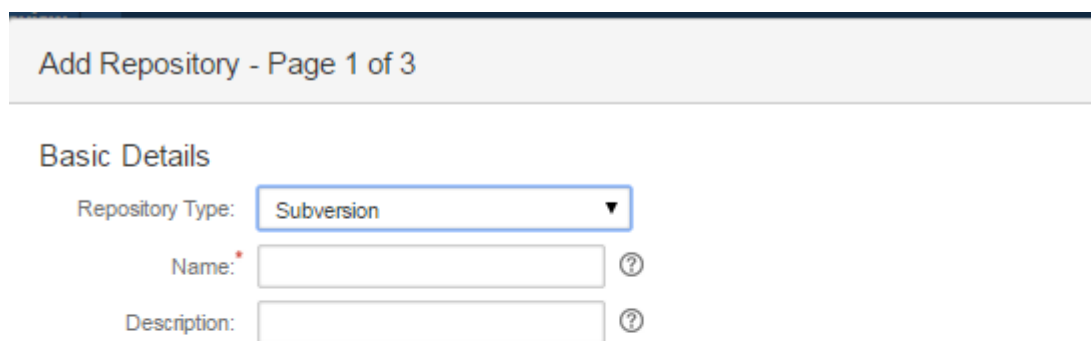


The screenshot shows the FishEye Admin interface. At the top is a navigation bar with links for Repositories, Projects, People, Reviews, and a 'Create review' button. Below this is the 'Admin' header. On the left is a sidebar menu with categories: Repository Settings (containing Repositories, Defaults, Smart Commits, Web Hooks), Project Settings (containing Projects), and User Settings (containing Users, User Mappings, Avatars). The main content area is titled 'Repositories' and includes tabs for 'Native Repository Access (1)' and 'Plugins (4)'. A text line states 'FishEye and Crucible natively supports Subversion, CVS, Perforce, G'. Below this is a table with columns 'Name', 'Type', and 'Description'. The table contains one entry: 'Dragons' (Type: Mercurial, Description: Dragons repository). Above the table are buttons for 'Add repository', 'Refresh', and a 'Filter repositories' input field.

Name	Type	Description
Dragons	Mercurial	Dragons repository

- Selezionare quindi **Add repository**

A questo punto, seguire il wizard, in modo da configurare l'accesso di FishEye al repository. Vediamolo nel dettaglio:



The screenshot shows the 'Add Repository - Page 1 of 3' wizard. Under the 'Basic Details' section, there are three fields: 'Repository Type:' with a dropdown menu set to 'Subversion', 'Name:' with a text input field and a red asterisk indicating it is required, and 'Description:' with a text input field. Each input field has a help icon (question mark in a circle) to its right.

Specificare la tipologia di repository. Fisheye consente di poter selezionare tra più tipologie (Es. Mercurial, CVS, Subversion, etc).

Add Repository - Page 2 of 3

SVN Connection Details

SVN URL: *

?

Path:

?

Username:

?

Password:

?

Passare quindi a fornire i parametri di connessione. Questa sezione del wizard varia a seconda della tipologia di repository. Viene quindi data la possibilità di eseguire il test di connessione, prima di andare al passo successivo.

Add Repository - Page 3 of 3

Final Settings

Store Diff Info: ☒ ?

Enable Repository ☒ ?

After Adding:

Checkout URL

Users can see a URL for checking out the repository's code.

☒ Use the system default settings for showing checkout urls

Show URL ☒

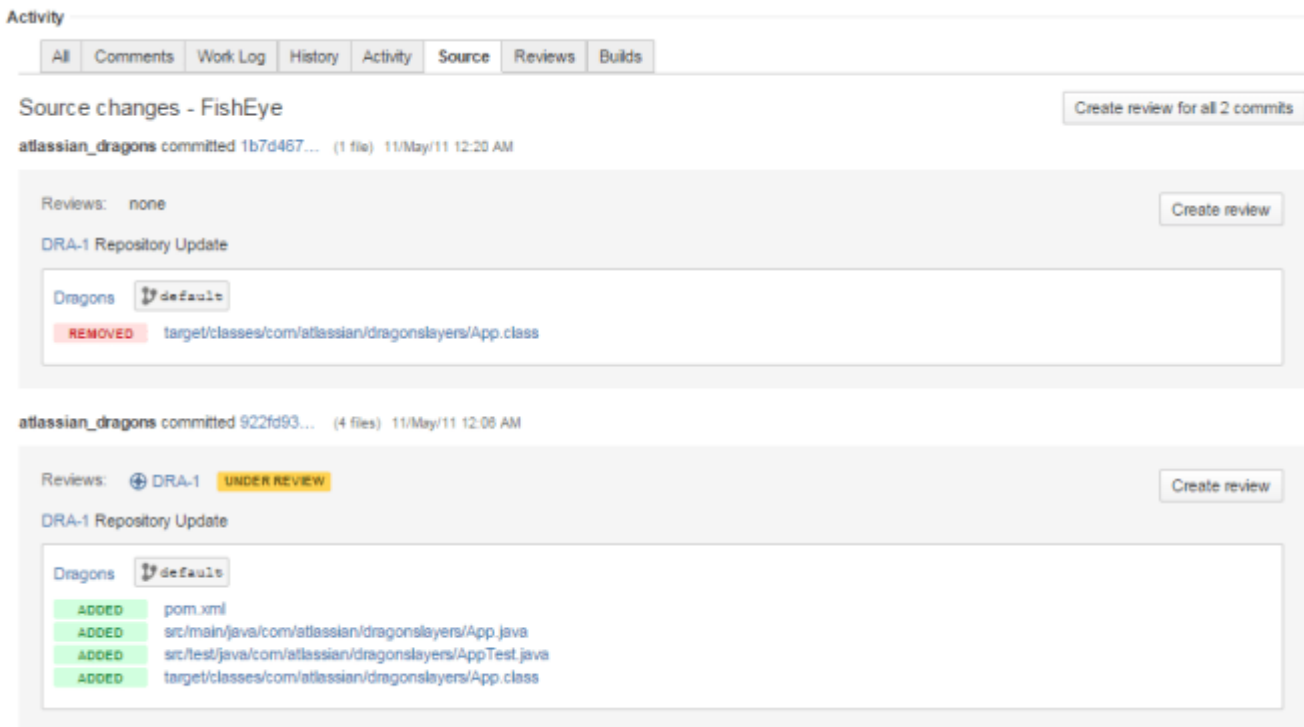
URL

Vengono quindi richieste le ultime configurazioni. Confermate anche queste ultime, si è configurato il Repository. Si può procedere alla fase di sincronizzazione. Al termine, si dispone di tutte le informazioni necessarie memorizzate sul DB di FishEye. Questa operazione potrebbe richiedere del tempo e, di conseguenza, occorre attendere che sia terminata prima di eseguire altre operazioni.

Questo è il risultato:



Come si può osservare, direttamente da FishEye andiamo a spulciare il codice. Ma questo è solo uno degli aspetti. Una volta collegato a JIRA, possiamo anche a referenziare direttamente il codice sorgente e le variazioni dello stesso da JIRA. Semplicemente, andando a vedere la sezione **source**, possiamo andare a leggere quali interventi sono stati eseguiti



il tutto semplicemente andando a inserire, nel testo delle note a fronte del Commit, la chiave della issue JIRA. Infatti, usando questa chiave, Fisheye riesce a eseguire il link con la issue JIRA e attiva il TAB **source**.

Conclusioni

Abbiamo visto, in questo post, che cosa consente di fare FishEye nel dettaglio. Nei prossimi post approfondiremo l'argomento con la Code Review (Crucible) e con altre operazioni che sono possibili con Fisheye.

Piccoli suggerimenti

Consigli vari ed assortiti

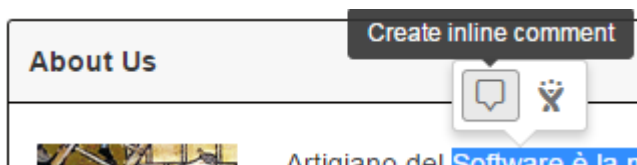
In questo post diamo dei piccoli consigli, sfruttando alcune piccole funzionalità presenti su Confluence e JIRA.



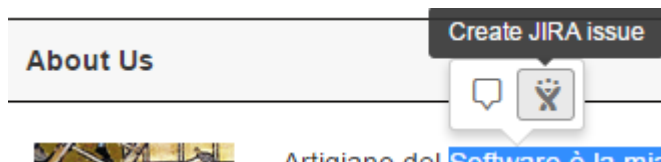
Creare commenti/Issue Jira da testo selezionato aka PINPOINT

Possiamo creare un commento selezionando un testo della pagina corrente, in modo molto semplice. Iniziamo selezionando il testo che ci serve. Vedremo che appare un piccolo gadget sopra il testo, con due opzioni:

- **Create inline comment** – attraverso il quale creiamo dei commenti in cui il testo selezionato viene evidenziato



- **Create Jira Issue** – attraverso il quale creiamo una Issue Jira partendo dal commento selezionato.



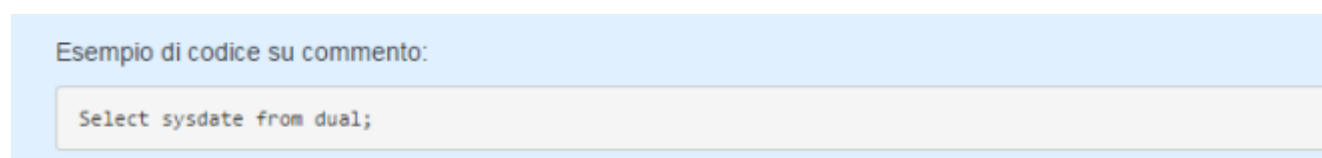
Queste autocomposizioni possono tornare molto utili per velocizzare diverse operazioni.

Indicare il codice nei commenti JIRA

In alcune occasioni può risultare utile inserire del codice nei commenti di JIRA. Un piccolo consiglio può risultare utile, al fine di rendere il commento adeguatamente formattato. Se si inserisce il commento come mostrato di seguito:

```
{code}
.....<codice>....
{code}
```

il codice sarà visualizzato su di un riquadro e formattato adeguatamente. La seguente immagine chiarisce il risultato:



Se si vuole inserire un ***syntax highlighting*** al codice scritto, allora occorre impostare il linguaggio nel tag ***code***, come mostrato in figura:

```
{code:sql}
```

.....<codice>.....
{code}

Esempio di codice formattato

```
select sysdate from dual;
```

Elenchi puntati su commenti di JIRA

Un altro esempio di come indicare degli elenchi puntati, su commenti di JIRA, è riportato di seguito:

- * esempio1
- * esempio2

- esempio1
- esempio2

Il risultato del seguente esempio è il seguente:

Esempio di elenco puntato:

- esempio1
- esempio2

Altro esempio

- esempio1
- esempio2

In questo caso, il carattere * ed il carattere – vengono rappresentati come due diversi tipi di pallino per gli elenchi puntati.

E' possibile impostare diversi livelli di elenchi puntati, utilizzando i seguenti codici:

```
# esempio
# controesempio
#* interno1
#* interno2
#* interno3
# altro esempio
```

Questo è il risultato:

1. esempio
2. controesempio
 - ◊ interno1
 - ◊ interno2
 - ◊ interno3
3. altro esempio

Invertendo l'ordine:

```
* esempio
* controesempio
*# interno1
*# interno2
*# interno3
* altro esempio
```

Il risultato è il seguente:

- esempio
- controesempio
 1. interno1
 2. interno2
 3. interno3
- altro esempio

Conclusioni

In questo post abbiamo visto dei piccoli consigli, molto semplici, ma che possono aiutare nel proprio lavoro di tutti i giorni :-). Nei prossimi post daremo altri piccoli consigli per formattare o usare tante piccole funzionalità di Confluence e JIRA e aiutare l'uso quotidiano ☐

Gallerie di immagini

Gallerie di immagini

Vedremo, in questo post, un semplice esempio di come realizzare una semplice galleria di immagine.

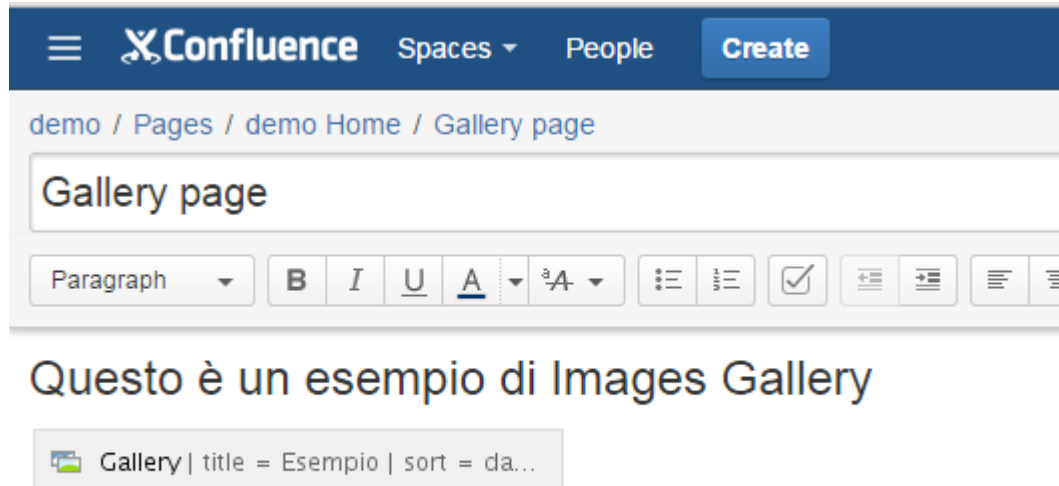


Di cosa abbiamo bisogno?

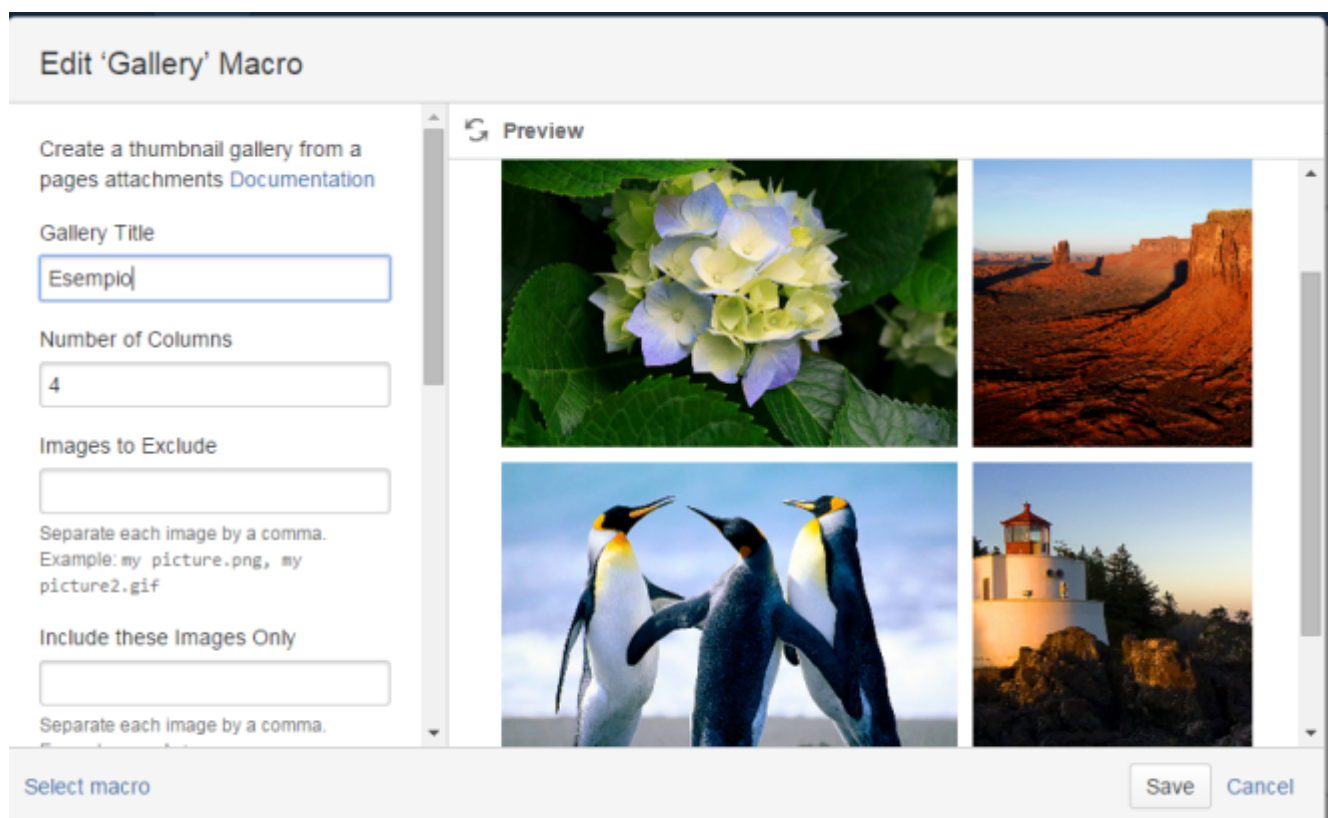
Quello di cui necessitiamo è:

- allegati a pagina
- Gallery Macro

Semplice no? Andiamo a vedere cosa inserire nella pagina. La seguente immagine mostra che cosa viene inserito nella pagina:



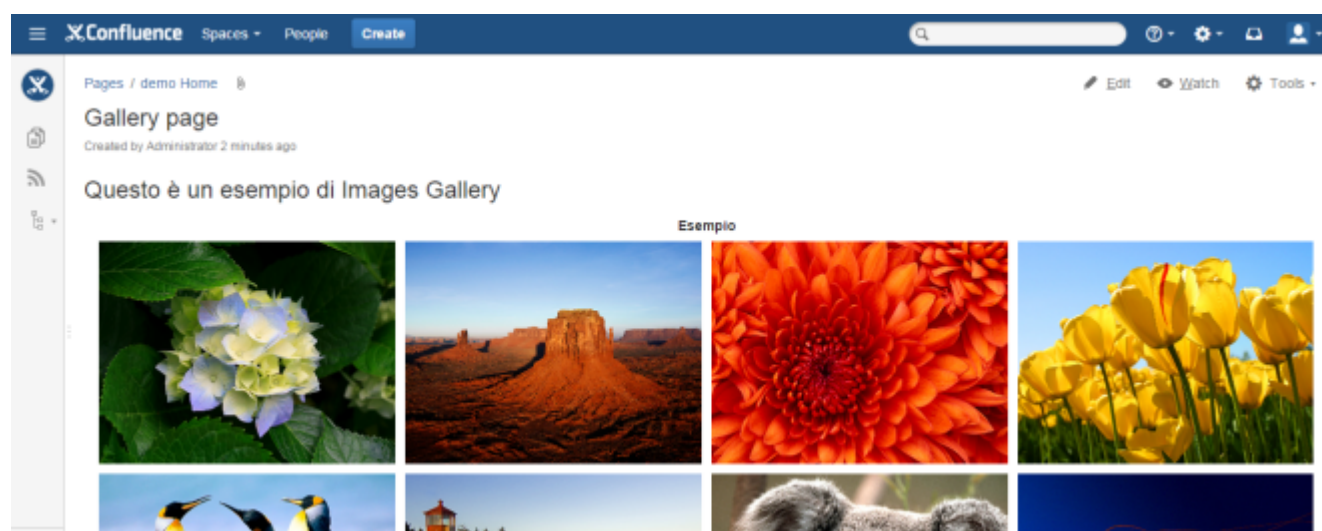
Se andiamo a visualizzare le proprietà della macro, abbiamo alcune sorprese:



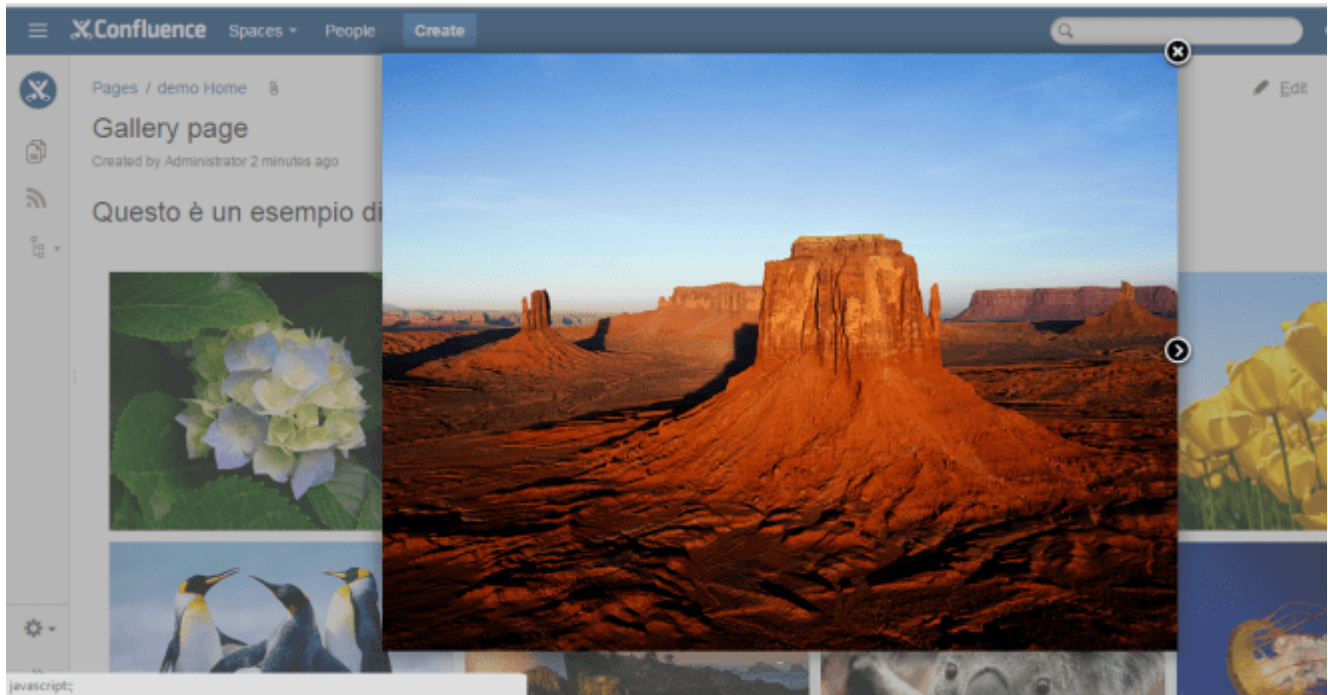
Come possiamo vedere possiamo impostare le seguenti caratteristiche:

- Titolo della Gallery
- Numero di colonne con cui si vuole impostare la gallery
- Quali immagini da escludere
- Quali immagini includere
- Quali etichette, associate alle immagini, considerare
- Quali etichette, associate alle immagini, escludere
- Da quale pagina andare a selezionare le immagini allegate.

Risultato? lo vediamo nella seguente immagine:



Se andiamo a selezionare una immagine, questo è il risultato:



Possiamo allegare le immagini su di una pagina specifica, ma possiamo usare una seconda pagina per visualizzare la gallery

□

Conclusioni

Abbiamo visto una potenzialità di Confluence, attraverso la quale poter realizzare una gallery di immagini in maniera semplice e veloce.

Reference

Fare riferimento a questo [link](#) per la manualistica di Confluence.

Gestione del tempo automatico con JIRA Cloud

Un piccolo automatismo

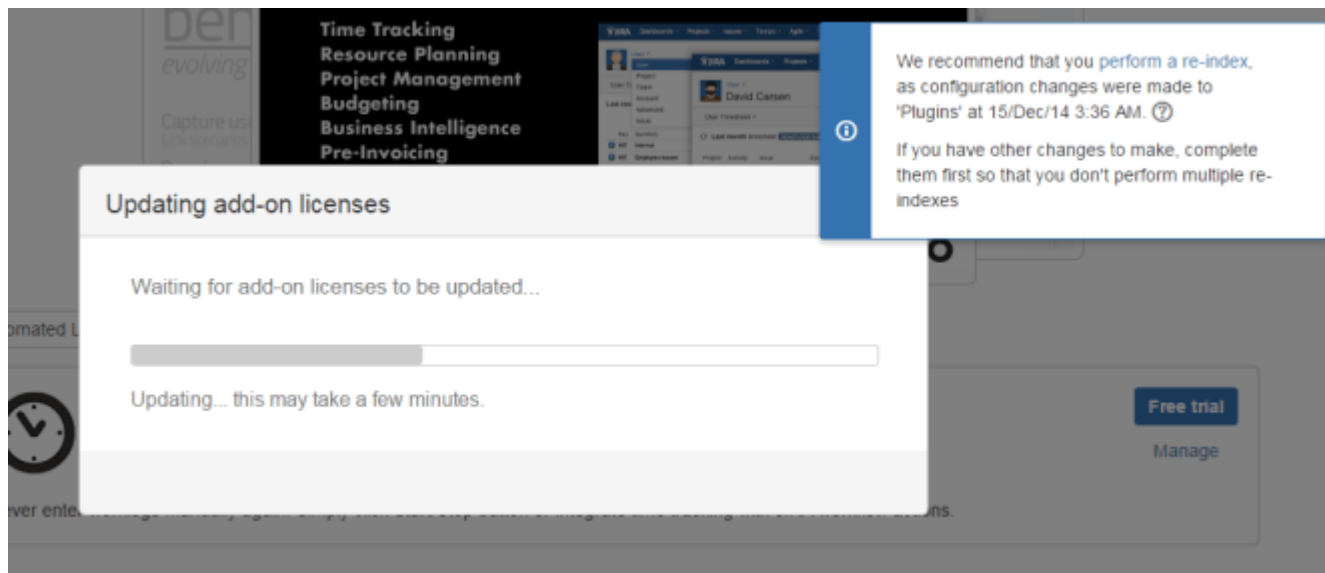
Riprendiamo questo addon per il servizio Cloud, di cui avevamo accennato in questo [post](#). Si tratta del [Automated Log Work for JIRA](#), un semplice addon che consente di automatizzare il conteggio delle ore lavorate. Questo aiuta notevolmente nelle operazioni di tracciatura delle ore lavorate.

In questo post andremo ad analizzarne il funzionamento.



First Look

L'installazione non ha richiesto tantissimo tempo ed è totalmente automatizzata. Si tratta di seguire tutte le istruzioni. In questo esempio ho installato una free trial su di una istanza Cloud (ex [onDemand](#)).



NB: Ricordarsi, al termine della installazione, di eseguire una re-index, come ricorda il nuovo sistema di messaggistica popup, mostrato in alto a destra della figura.

Terminata l'installazione, iniziamo subito ad usare l'addon. Creiamo un task di prova ed andiamo ad esaminare che cosa possiamo fare:

The screenshot shows the 'Create Issue' form. At the top right is a 'Configure Fields' button with a gear icon. The form fields are: 'Project*' with a dropdown menu showing 'Corsi Atlassian/Oracle (CA)'; 'Issue Type*' with a dropdown menu showing 'Bug' and a help icon; 'Summary*' with a text input field containing 'Demo'; 'Priority' with a dropdown menu showing 'Major' and a help icon; 'Component/s' with a dropdown menu and a hint 'Start typing to get a list of possible matches or press down to select.'; 'Affects Version/s' with a dropdown menu and the same hint; and 'Description' with a text input field containing 'Demo'.

Una volta creato, se andiamo nel dettaglio del task,

Corsi Atlassian/Oracle / CA-2

Demo

Edit Comment Assign More Start time Stop Progress Workflow Admin

Details

Type:	☒ Bug	Status:	IN PROGRESS (View Workflow)
Priority:	↑ Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Component/s:	None		
Labels:	None		

notiamo la presenza di due tasti che consentono di far ...
contare in automatico il tempo di risoluzione di un task :-).

Selezionando **Start Time**, attiviamo il conteggio

Issues Service Desk Agile Create

✓ **Timer started**
Started work timer for issue CA-2

Start time Stop Progress Workflow Admin

Selezionando **Stop Time**, si ferma il conteggio

Issues Service Desk Agile Create

✓ **Timer Stopped**
Logged 5m for an issue CA-2

Start time Stop Progress Workflow Admin

Start issue timer

Il risultato è visibile sul tab **Work Log**:

Activity

[All](#)[Comments](#)[Work Log](#)[History](#)[Activity](#)

 [Fabio Genovese \[Administrator\]](#) logged work - 19/Dec/14 11:59 PM

Time Spent:

5 minutes

AutomatedLogWork Connect auto log

Questo è ottenibile principalmente dalla versione **Cloud**. Se invece passiamo alla versione server, sono disponibili anche le seguenti opzioni:

Possibilità di configurare l'attivazione dell'addon, in modo che sia automatico o manuale, come mostrato in figura:



Se attivata la modalità automatica, quando si utilizzano le BOARD, l'attivazione/disattivazione del time tracking viene eseguita al passaggio di stato.



Conclusioni

Abbiamo visto, in questo post, l'utile addon che consente di semplificare il lavoro di uno sviluppatore, e non, in modo da poter tracciare autonomamente il proprio lavoro, senza dover ogni volta tenere traccia di quello che sta eseguendo. l'Addon è a pagamento, sia per le versioni Cloud che per le versioni server. Lo consiglio soprattutto per i team di sviluppo formati da molti elementi.

**Arrivederci Atlassian
onDemand. Benvenuto Atlassian
Cloud**

Atlassian Cloud

La Atlassian, nel [sul blog](#), ha informato che sarà reso operativo il servizio di **Atlassian Cloud**. Con questo servizio metteranno a disposizione tutti i prodotti della Atlassian nel cloud.



Ma il servizio onDemand?

Non sparisce. Semplicemente cambia nome da **onDemand** a **Cloud**. Rendono i servizi migliori e più performanti.



Tutti coloro che dispongono del servizio **onDemand** vedranno semplicemente sparire l'etichetta **onDemand** a favore dell'etichetta **Cloud**, ma continueranno ad avere gli stessi servizi, con una maggiore reattività.

Maggiori raggiagli sui servizi offerti (che sono praticamente gli stessi di **onDemand**) alla seguente [pagina di documentazione](#).

Conclusioni

Prepariamoci a delle belle sorprese. □