

JIRA Service desk 2.5 – First look

Nuove migliorie

In questo post andremo a visionare le ultime novità introdotte dalla Atlassian, su JIRA Service Desk per la versione 2.5. Continuiamo quanto riportato nei post:

- [JIRA Service Desk 2.0](#)
- [JIRA Service Desk 2.2](#)
- [JIRA Service Desk 2.3](#)
- [JIRA Service Desk 2.4](#)



Ultime novità

Le ultime novità riguardano i seguenti aspetti:

- **Reporting:** Migliorato ed aggiunte nuovi reports, per

meglio monitorare le attività degli agenti.



- **Portal:** Migliorata ulteriormente la possibilità di personalizzare il **Portal**



- **Queue Update:** Migliorato il comportamento delle code e migliorato l'update



Conclusioni

Si tratta di piccole migliorie, ma sono sicuro che introducono un grande miglioramento. Nei prossimi post andremo ad esaminarle nel dettaglio, continuando il percorso iniziato con i seguenti post:

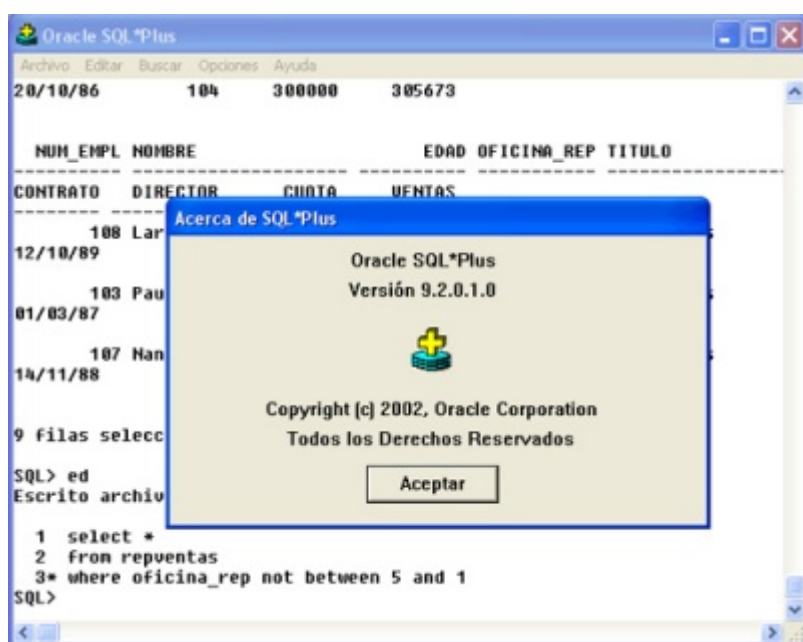
- [Esempio di uso – Parte 1](#)
- [Esempio di uso – Parte 2](#)
- [Esempio di uso – Parte 3](#)



Esportazione dati via SQLPLUS

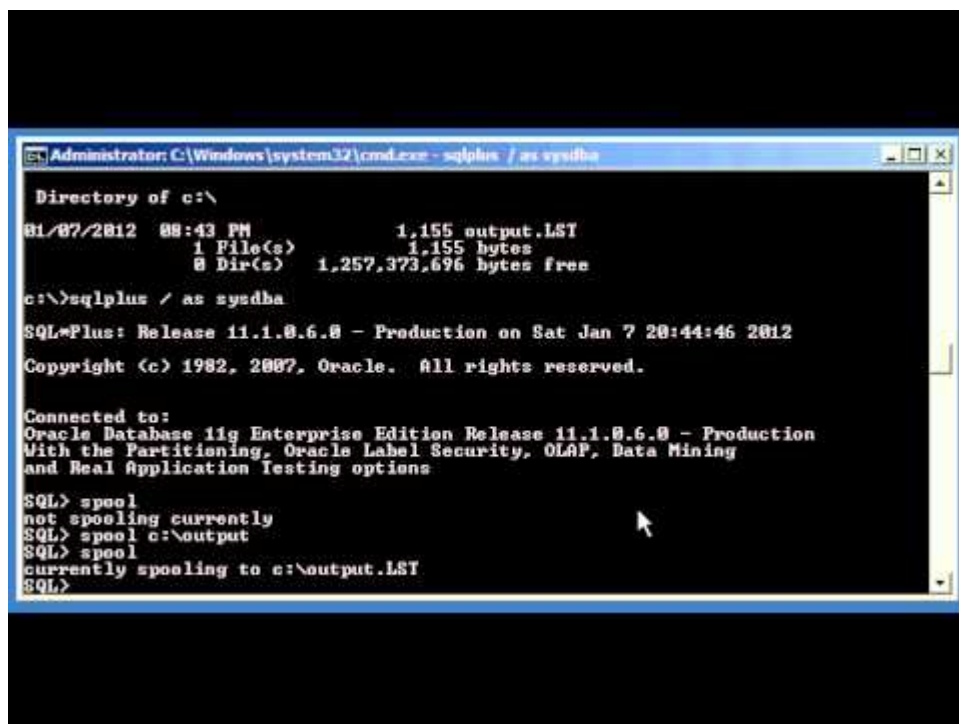
Un suggerimento

In questo post, andremo a vedere un piccolo **trucchetto** che ho usato per poter esportare dei dati, soprattutto quando si tratta di esportare dati di una certa mole.



Andiamo nel dettaglio

L'idea è di fare uso di **SQLPLUS**, che si presta bene a questo genere di lavori, sfruttando le sue capacità. In particolare ci appoggiamo alla possibilità di poter inviare su file il risultato della esecuzione della query. Stiamo parlando del comando: **SPOOL**;



```
Administrator: C:\Windows\system32\cmd.exe - sqlplus / as sysdba

Directory of c:\
01/07/2012  08:43 PM                1,155 output.LST
               1 File(s)                1,155 bytes
               0 Dir(s)  1,257,373,696 bytes free

c:\>sqlplus / as sysdba

SQL*Plus: Release 11.1.0.6.0 - Production on Sat Jan 7 20:44:46 2012
Copyright (c) 1982, 2007, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining
and Real Application Testing options

SQL> spool
not spooling currently
SQL> spool c:\output
SQL> spool
currently spooling to c:\output.LST
SQL>
```

Andiamo nel dettaglio e vediamo come fare.

Una volta definita la query, con tutti i dati che sono necessari. Quindi componiamo la query come segue:

```
Select <campo1> || <campo2> || ..... || <campon> from .....
```

oppure come

```
Select <campo1> || <delimitatore> || <campo2> ||
<delimitatore> || ..... || <campon> from .....
```

a seconda che si voglia avere una estrazione dove i campi sono a lunghezza fissa (primo caso) o con delimitatore (secondo caso).

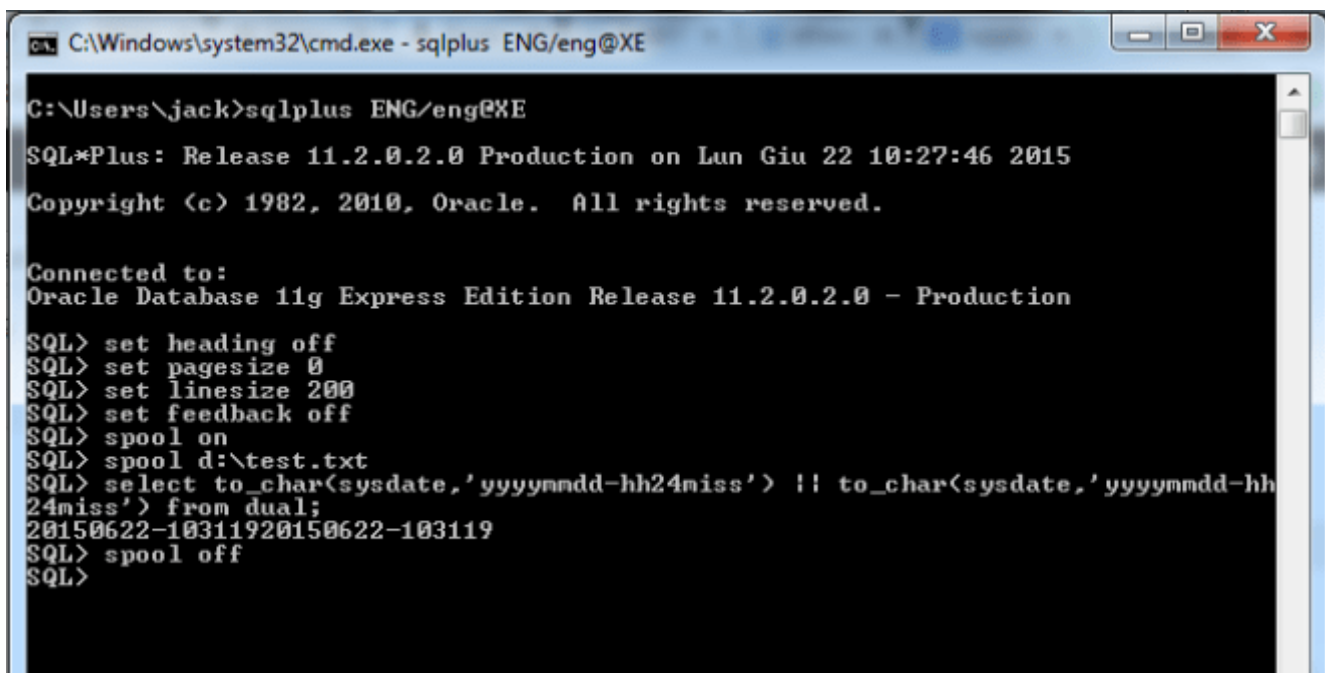
A questo punto, vediamo come procedere per avere l'estrazione vera e propria.

Attiviamo SQLPLUS, da riga di comando:

sqlplus <utente>/<password>@<dbsid>

Quindi forniamo la sequenza di comandi:

```
set heading off
set pagesize 0
set linesize 200
set feedback off
Spool on
Spool <file_destinazione>.txt
<scriviamo la Query, mettendo il ';' finale >
Spool off <da lanciare al termine della esecuzione della
query>
```



```
C:\Windows\system32\cmd.exe - sqlplus ENG/eng@XE

C:\Users\jack>sqlplus ENG/eng@XE
SQL*Plus: Release 11.2.0.2.0 Production on Lun Giu 22 10:27:46 2015
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production

SQL> set heading off
SQL> set pagesize 0
SQL> set linesize 200
SQL> set feedback off
SQL> spool on
SQL> spool d:\test.txt
SQL> select to_char(sysdate, 'yyyymmdd-hh24miss') || to_char(sysdate, 'yyyymmdd-hh
24miss') from dual;
20150622-10311920150622-103119
SQL> spool off
SQL>
```

La precedente figura da una idea per capire come procedere. Ovviamente mi sono limitato a eseguire una query di esempio per mostrare il funzionamento.

Conclusioni

Abbiamo un sistema molto semplice per eseguire delle estrazioni dati molto semplici. Il metodo è automatizzabile, creando degli script ad hoc, per semplificarci ulteriormente la vita :-).

User Macro – Primo esempio

Hello World

In questo post, continuiamo ad esaminare le User Macro. In particolare andremo a vedere quale iter seguire per creare la prima User Macro: **HELLO WORLD** ☐

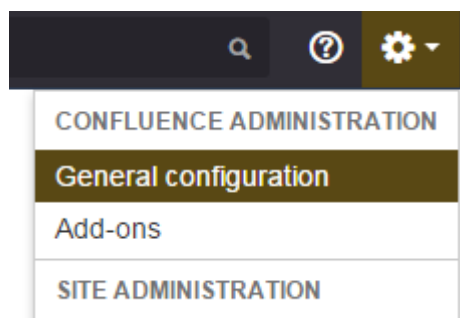


Premessa

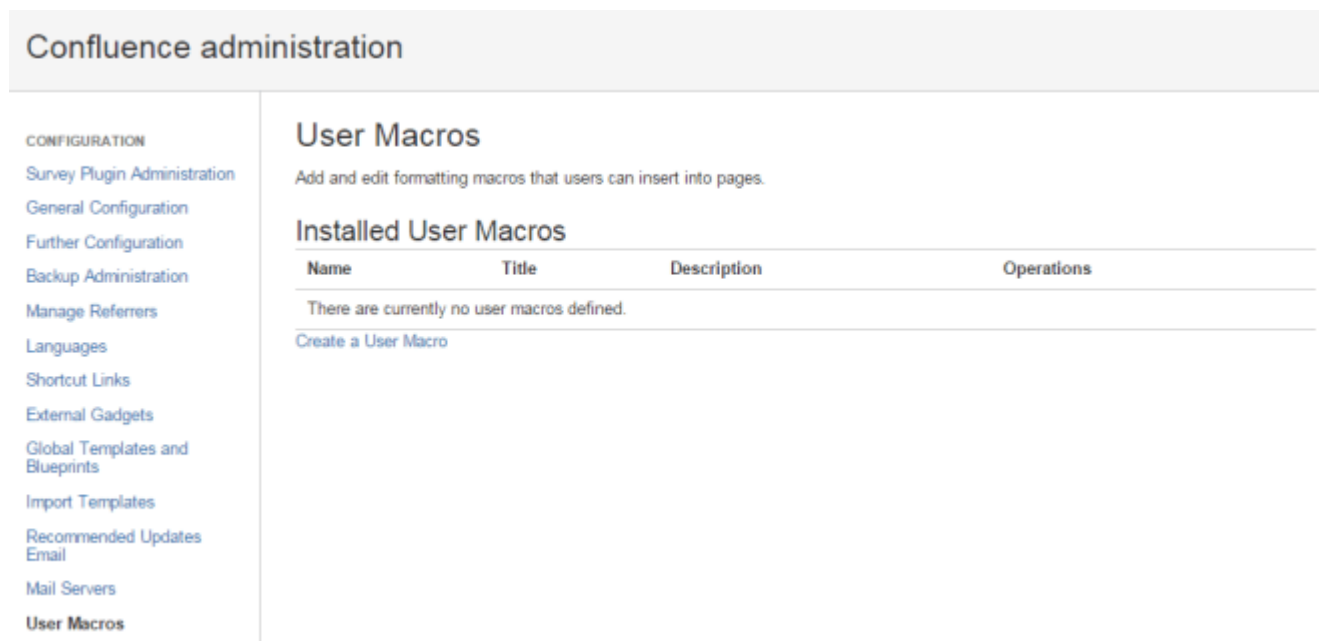
Le **User Macro** sono disponibili solo per le installazioni server. Sulle installazioni Cloud, non abbiamo alcuni Addon, causa motivi di sicurezza. In altri post, andremo ad esaminare più in dettaglio le differenze tra le due installazioni.

Procediamo

Dato che si tratta di una funzionalità esistente nello standard, non abbiamo bisogno di preparazione o di installare alcunché. Come già descritto nel precedente [post](#), andiamo a posizionarci nella sezione dedicata alla generazione delle user macro.



Quindi, sotto la sezione delle **Configuration**, andiamo a selezionare le **User Macro**.



Procediamo quindi a generare la nostra User Macro, selezionando **Create a user macro** si procede con la creazione vera e propria

Create User Macro

Macro Browser Information

Macro Name*
Macro names are converted to lower case

Visibility ☒ Visible to all users in the Macro Browser
☐ Visible only to system administrators in the Macro Browser

Macro Title*

Description

Categories

Formatting

Media

Navigation

Reporting

Visuals & images

Icon URL
Absolute URL (e.g. <http://mysite.com/myimage.png>), or relative to the Confluence

Documentation URL

Si impostano i primi parametri, quali:

- **Nome:** assegniamo il nome alla macro.
- **Visibilità:** indichiamo chi può vedere/utlizzare la macro
- **Titolo:** Sarà il testo che sarà visualizzato nella lista delle macro
- **Descrizione:** Sarà la descrizione che sarà posizionata nella lista delle varie macro.
- **Categorie** cui assegnare la macro. Le macro sono normalmente raggruppate in categorie. Si tratta solo di definire dove la macro sarà visualizzata
- **Icona** da visualizzare nell'elenco delle macro. Si tratta di una URL in cui sarà disponibile l'icona da usare.
- **Documentazione.** URL di riferimento della documentazione

associata.

- **Codice vero e proprio della macro.** Si tratta del codice della macro. Qui abbiamo l'azione definita.

Definition of User Macro

Macro Body ☒ No macro body

Processing ☐ Escaped

☐ Unrendered

You should use this option for bodies that are processed within the template before being output. Ensure that HTML is ultimately output by the template.

☐ Rendered

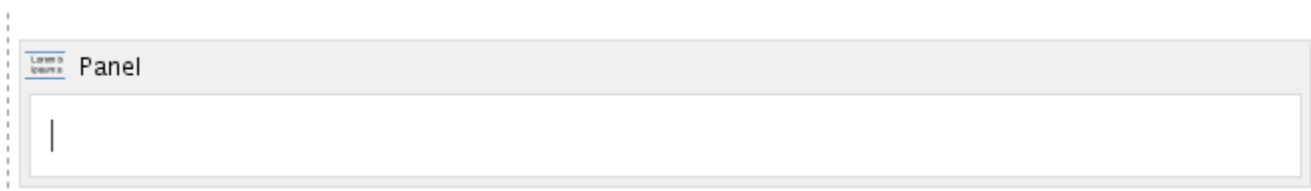
The body will be rendered so most HTML entered will be passed to the template unmodified but Confluence specific mark up such as macro definitions will be rendered.

Template *

```
## Macro title: My Macro
## Macro has a body: Y or N
## Body processing: Selected body processing option
## Output: Selected output option
##
## Developed by: My Name
## Date created: dd/mm/yyyy
## Installed by: My Name

## @noparams
Hello World!
$body
```

In questo caso abbiamo inserito il codice del nostro **Hello World**. Ricordiamoci di settare il parametro **Macro Body Processing** come **Rendered**. Questo parametro è importante in quanto indica se la macro presenta un body da trattare. Chiariamo meglio: Per body si intendono quelle macro in cui è possibile scrivere un testo da poter poi processare. Come si vede nella figura successiva:



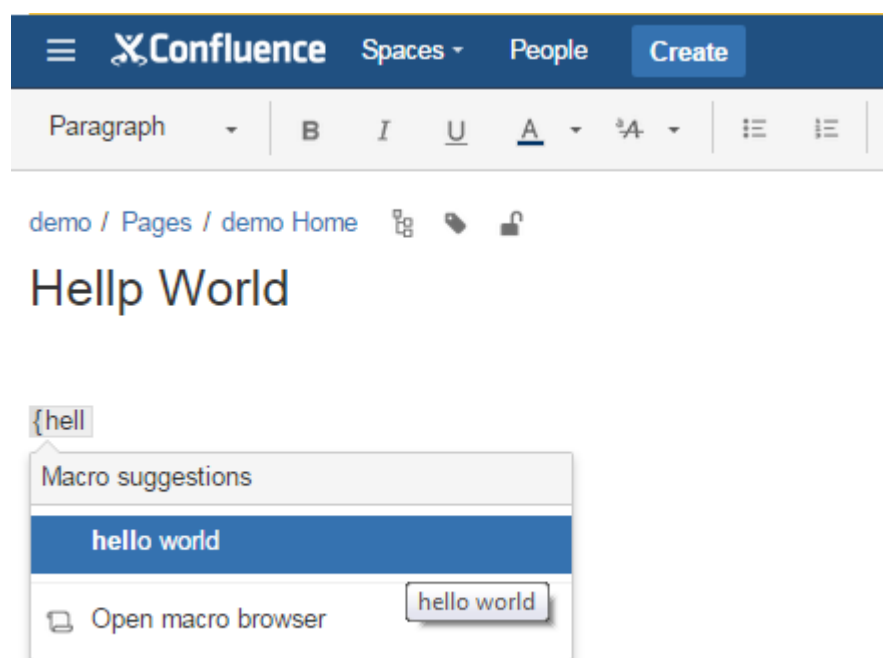
Abbiamo la visualizzazione della macro **Panel**, usata per generare un Pannello dove raggruppare informazioni. Il **Body** è la sezione dove viene posizionato il cursore e dove possiamo riportare del testo.

Chiarito questo primo concetto, vediamo le possibili

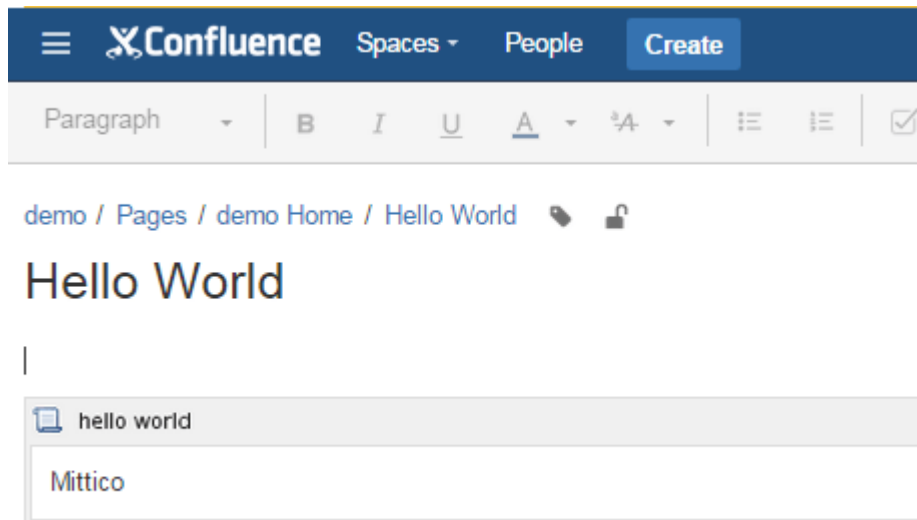
alternative del parametro precedentemente indicato:

- **No macro body** – La macro non presenta tale body.
- **Escaped** – da usare se si vogliono visualizzare i markup HTML. In questo modo, saranno aggiunti dei codici escape per la visualizzazione dei tag
- **Unrendered** – Il codice HTML sarà processato all'interno del codice vero e proprio.
- **Rendered** – HTML presente nel body sarà processato e l'output sarà visualizzato. Ad esempio: `<b>Hello World</b>` sarà visualizzato come: **Hello World** □

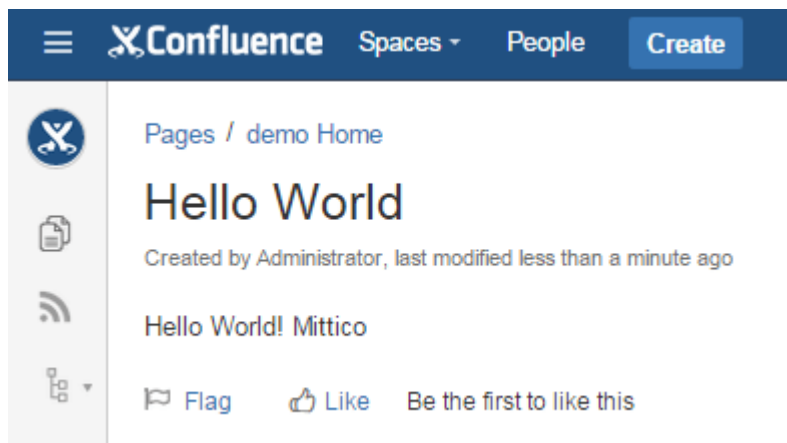
Una volta salvata, la macro è disponibile. Creiamo una nuova pagina, come mostrato in figura:



Selezioniamo la macro, come una qualsiasi macro disponibile. Quindi scriviamo un qualsiasi testo all'interno



Quindi salviamo la pagina. Il risultato è il seguente:



Conclusioni

Abbiamo visto come si può creare una semplice User Macro in Confluence.

<https://confluence.atlassian.com/display/DOC/Writing+User+Macros>

<https://developer.atlassian.com/confdev/tutorials/macro-tutorials-for-confluence/creating-a-new-confluence-macro>

<https://developer.atlassian.com/confdev/tutorials/macro-tutorials-for-confluence>

Kanoah – Test addon per JIRA – 2

Prova su strada – Parte 2

In questo post, concludiamo la prova su strada degli addons rilasciati dalla Kenoah, presentato nel post. La prima parte della prova su strada è visibile nel seguente post.



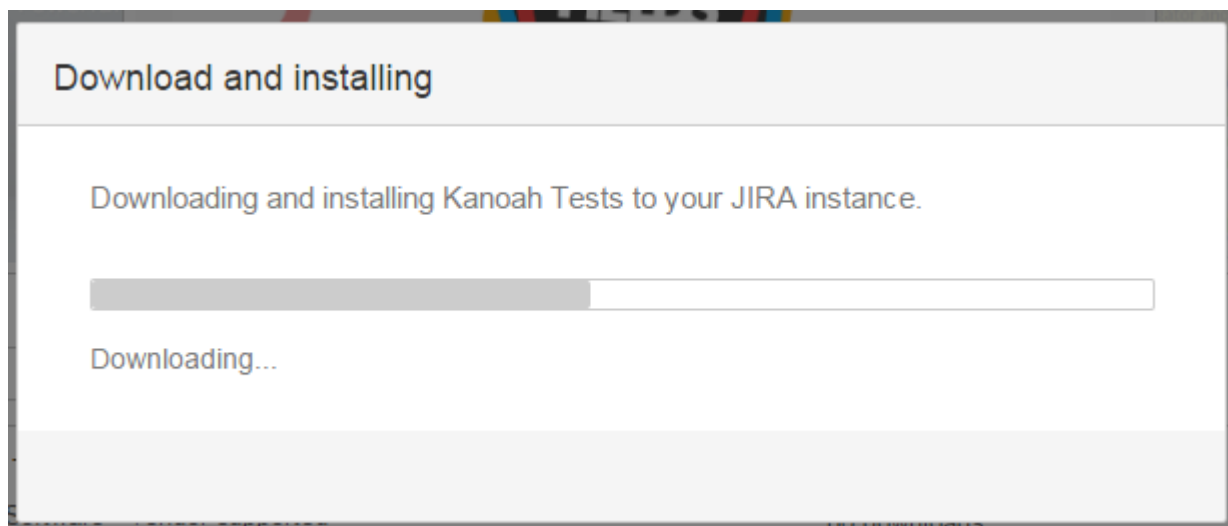
Test cases – Definizioni preliminari

Prima di iniziare la prova su strada, partiamo dalle definizioni preliminari e facciamoci aiutare da Wikipedia. Prima di continuare, fare riferimento a:

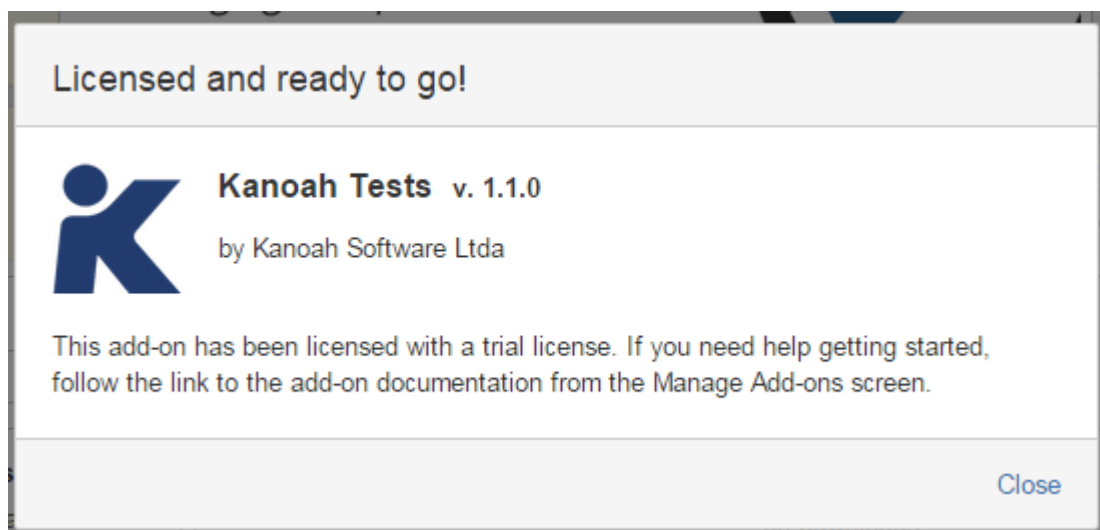
- [Test Case](#) o caso di test. Si tratta delle condizioni per verificare se il programma risponde correttamente
- [Test Plan](#) o piano dei test che devono essere eseguiti per verificare che le funzionalità, che compone un software, rispondono correttamente

Test test test test

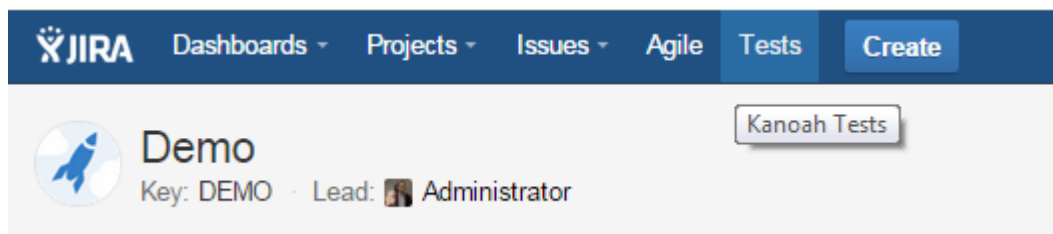
Iniziamo con l'installazione dell'addon



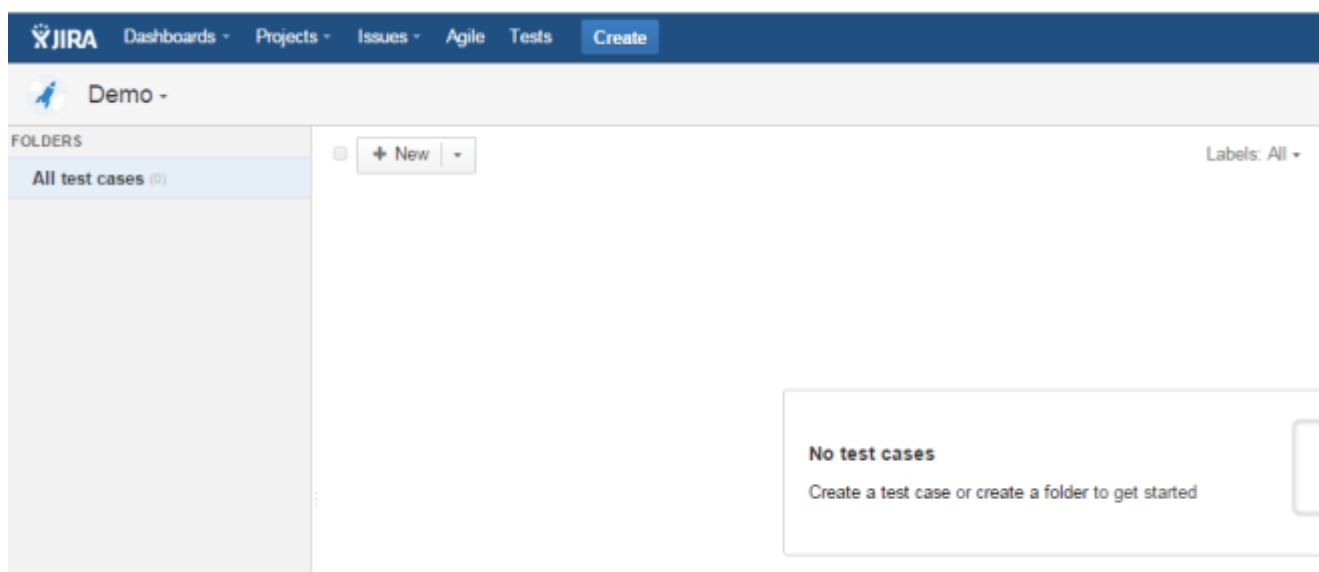
Dopo aver eseguito l'installazione, l'addon è pronto all'uso ☐



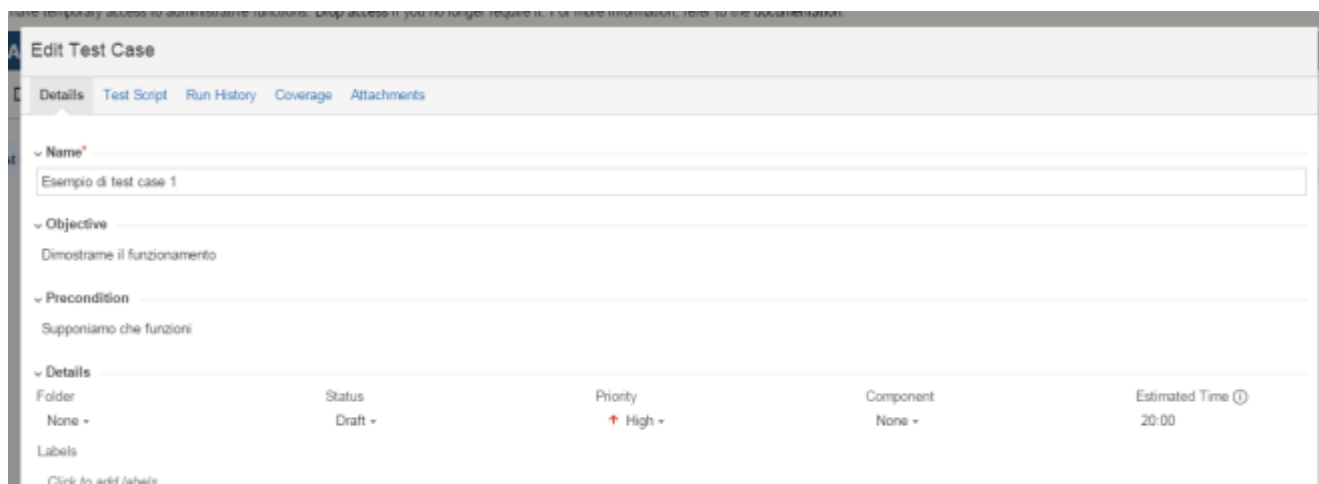
Possiamo usare subito l'addon. Tra i vari menù, abbiamo a disposizione una nuova voce: **Tests**.



Se selezioniamo questa nuova voce di menù, passiamo alla definizione dei vari Test Case



Procediamo con la definizione di un Test Case di prova :-).
Selezioniamo **NEW**

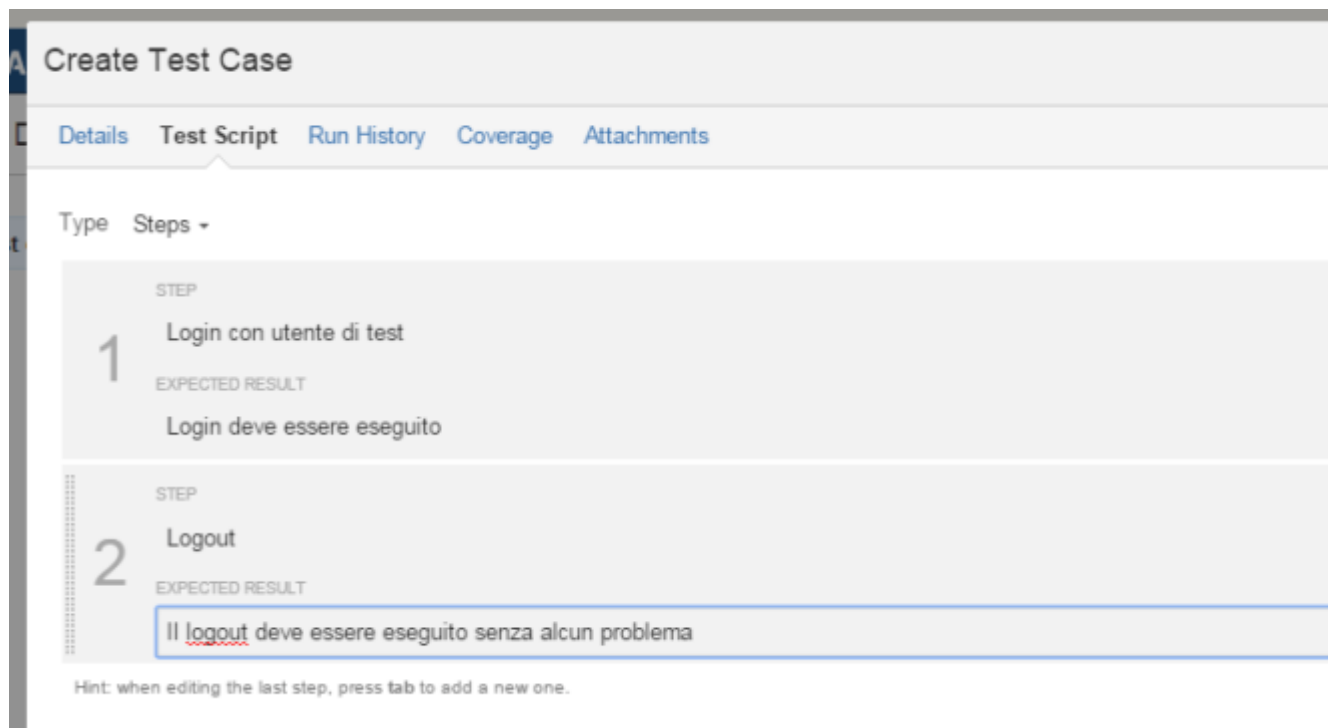


Possiamo definire le seguenti sezioni:

- **Details**, intesi come i dettagli generali del test

- **Test Script**, dove definiamo il dettaglio dei vari passi che devono essere eseguiti
- **Run History**, dove vediamo le volte in cui il test è stato eseguito
- **Coverage**, dove abbiamo l'elenco delle issues che sono coperte da questo test
- **Attachments**, dove indichiamo tutti gli attachments che sono utili

Nel definire i Test Script, abbiamo la possibilità di poter definire quali passi devono essere eseguiti per eseguire il caso di test.



The screenshot shows the 'Create Test Case' interface with tabs for 'Details', 'Test Script', 'Run History', 'Coverage', and 'Attachments'. The 'Test Script' tab is active, displaying a list of steps. The first step is 'Login con utente di test' with the expected result 'Login deve essere eseguito'. The second step is 'Logout' with the expected result 'Il logout deve essere eseguito senza alcun problema'. A hint at the bottom states: 'Hint: when editing the last step, press tab to add a new one.'

Type	Steps
STEP	1 Login con utente di test EXPECTED RESULT Login deve essere eseguito
STEP	2 Logout EXPECTED RESULT Il <u>logout</u> deve essere eseguito senza alcun problema

Hint: when editing the last step, press tab to add a new one.

ovviamente si tratta di un esempio :-). Ma immaginate quali informazioni potete inserire. Possiamo dettagliare il test in tutti i passi che sono necessari per ricostruire la situazione. Questo significa che possiamo avere il dettaglio completo di quello che si vuole fare □

A questo punto, dopo aver definito tutti i **Test Case**, possiamo costruire tutti i **Test Plans**. Vediamo come:

Create Test Plan

Details Test Cases Run History Attachments

✓ Name*

Esempio di Test Plan

✓ Objective

Regression per il

✓ Details

Folder

None -

Come per i **Test Case**, andiamo a definire prima il Dettaglio generale.

Create Test Plan

Details Test Cases Run History Attachments

No test cases selected

Search test cases and drop them here

SEARCH TEST CASES

Folder

All (1)

Statuses: All

Components: All

Labels: All

Priorities: All

Search...

✓ Esempio di test case 1

Quindi, andiamo a selezionare i Test Case che vogliamo siano eseguiti. Un drag & Drop ed il gioco è fatto.

Edit Test Plan

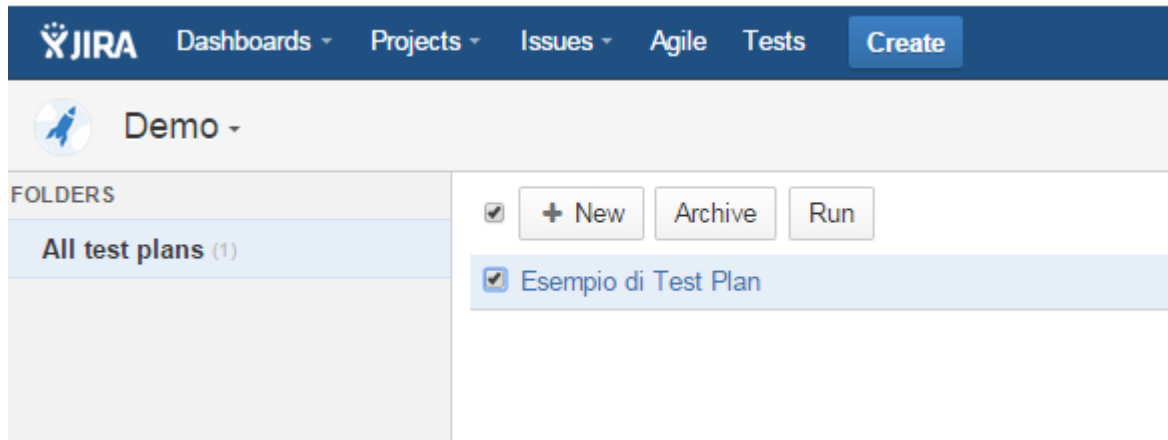
Details Test Cases Run History Attachments

↑ Esempio di test case 1

APPROVED 20h 0m

Salviamo ed il gioco è Fatto

Quindi?? andiamo ad eseguire il nostro Test Plan interessato



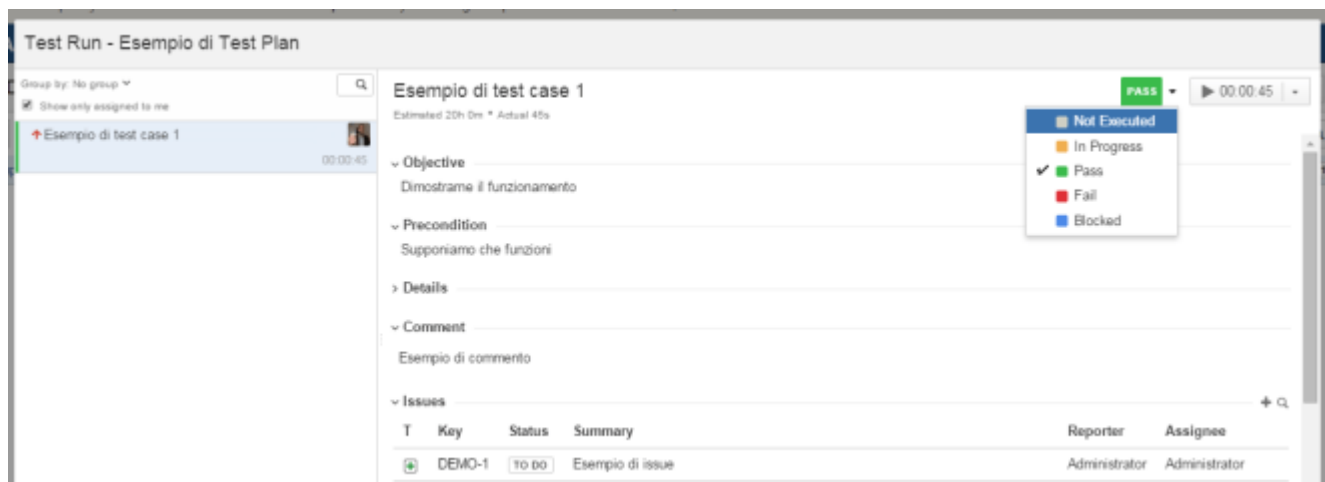
Selezioniamo il tasto **Run** per procedere con il test vero e proprio.

The screenshot shows the 'Create Test Run' form. It has a title bar 'Create Test Run'. The form contains several fields: 'Name' with a text input containing 'Esempio di Test Plan'; 'Description' with a large text area; 'Version' and 'Iteration' with dropdown menus both set to 'None'; 'Environments' with a large text area; 'Planned start date' and 'Planned end date' with date input fields.

Creando il Test Run, andiamo a indicare che caratteristiche deve possedere, indicando anche quando deve essere eseguita e conclusa.

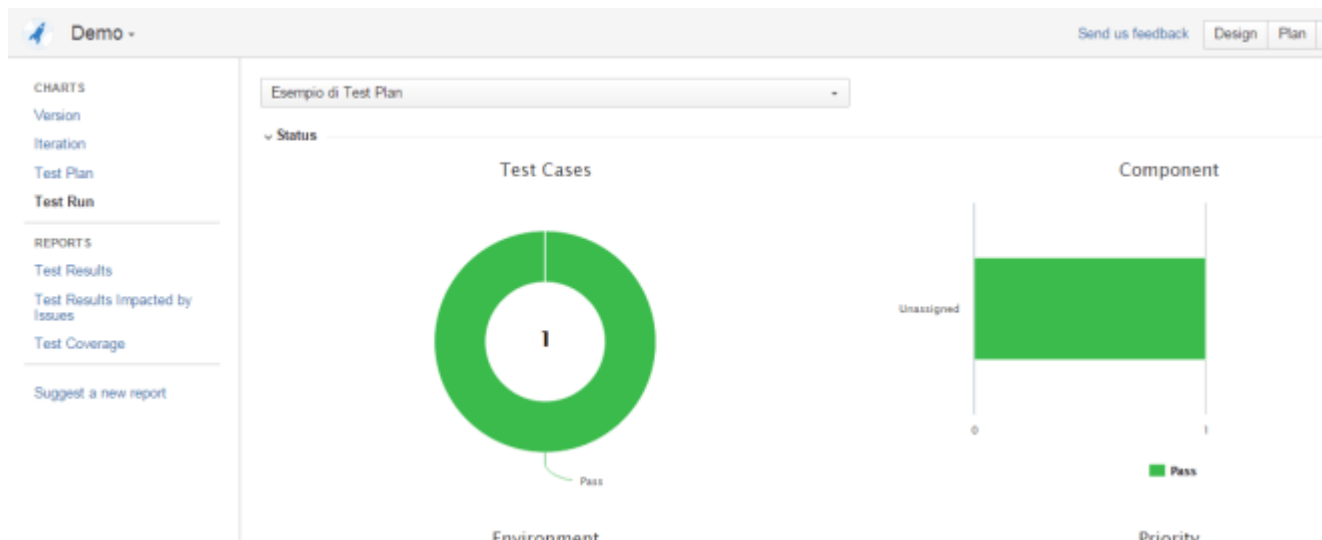


Quindi, una volta indicate queste informazioni, passiamo al RUN vero e proprio. Abbiamo a disposizione un Timer, che indica quanto tempo ha richiesto il test vero e proprio.



Se Il test è passato, settiamo lo stato in **Pass**. Altrimenti indichiamo lo stato richiesto.

Al termine, possiamo vedere i risultati nella reportistica, che l'addon mette a disposizione.



Come si vede, abbiamo tutte le informazioni del caso.

Conclusioni

Abbiamo un addon che consente di poter pianificare e gestire in maniera semplice e ben fatta tutti i test di cui abbiamo bisogno. In questo modo, possiamo disporre di una libreria di test, rendendo la vita del Team di test ... molto più piacevole e sicura.

Kanoah – Test addon per JIRA

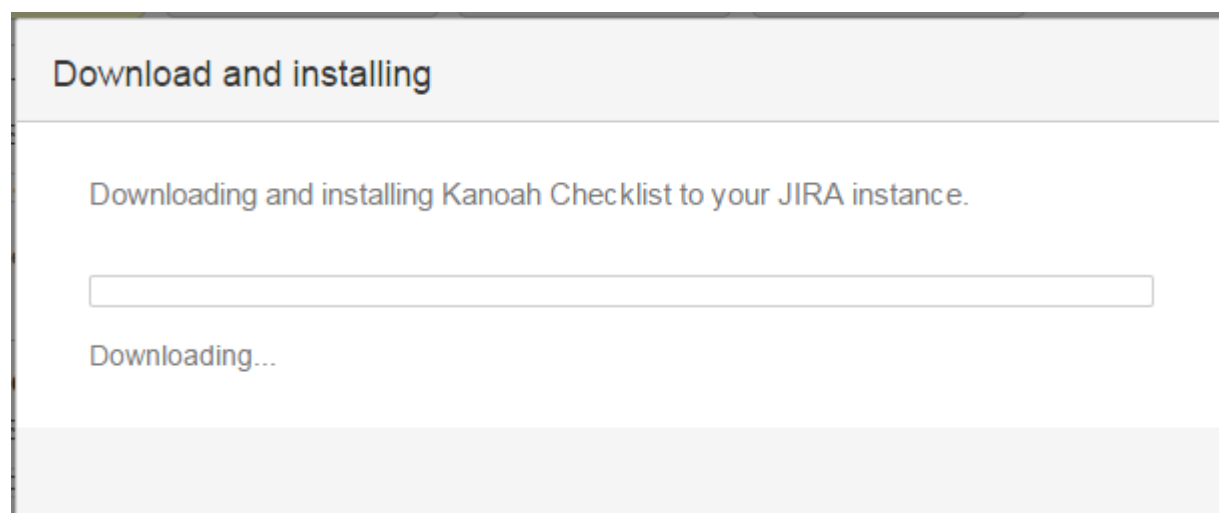
Prova su strada – Parte 1

In questo post andiamo ad esaminare la prova su strada degli addons, presentati nel precedente [post](#).

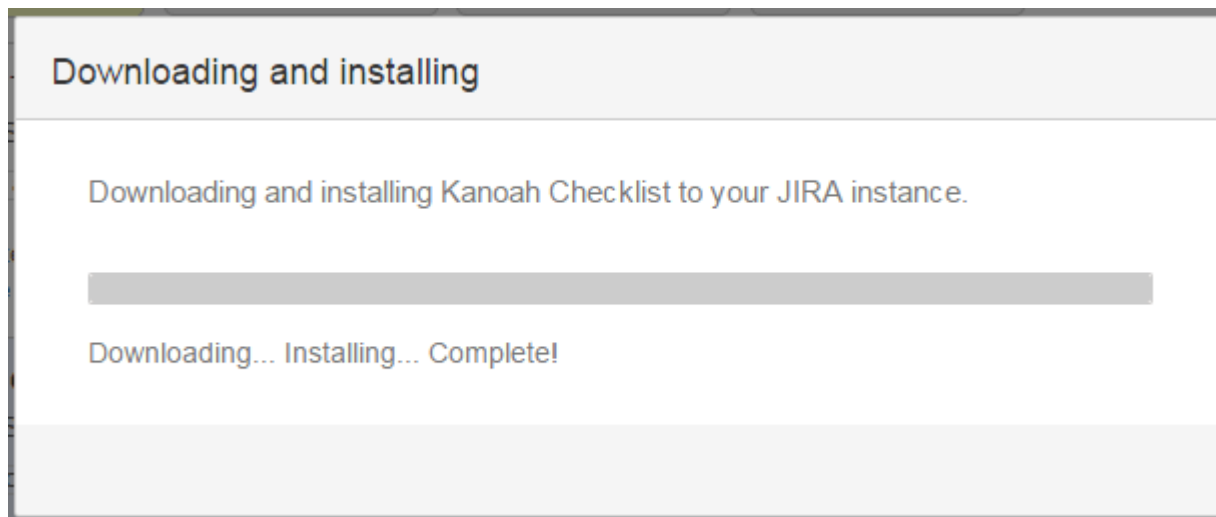


Checklist

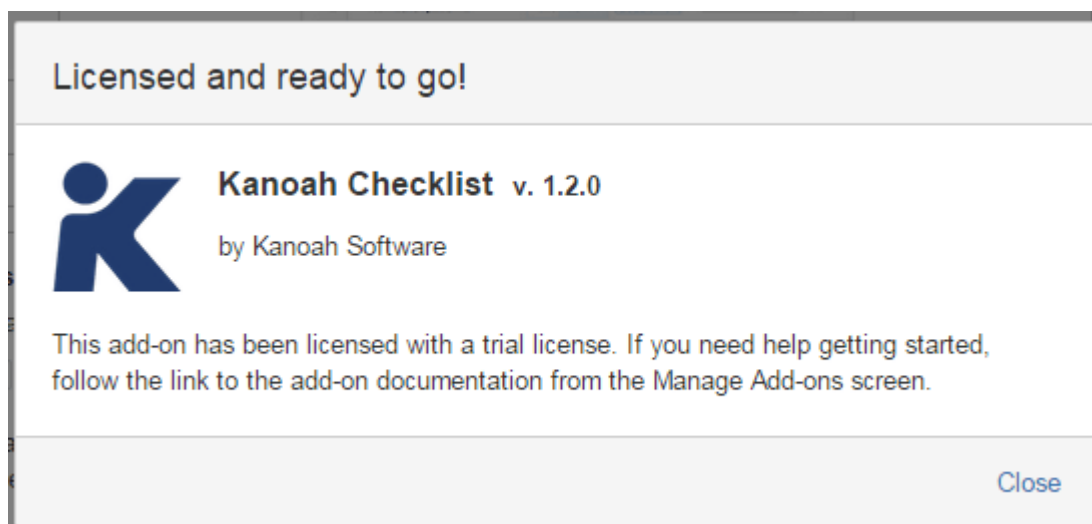
Partiamo dall'addon della checklist. Installiamo l'addon, come sempre:



L'addon viene scaricato e installato sulla nostra istanza server di JIRA.



Completata l'installazione, si passa a richiedere la licenza. Nel nostro caso, ho richiesto la licenza Trial.



Terminata l'operazione, si può usare l'addon. □

Il primo passo è quello di creare un nuovo Campo Custom, come mostrato nelle seguenti immagini.

Add Field

Name*

Lista di controllo (Create new fi...

Create **Cancel**

Possiamo creare direttamente dalla form di gestione della Issue oppure dalle form di amministrazione dei campi.

Select a Field Type

⚠ This custom field will also be visible in 3 other projects.

All
Standard
Advanced

☐ Unchecked Item
☒ Checked Item

Checklist
A checklist style custom field.

Group Picker (multiple groups)
Choose multiple user groups using a popup picker window.

Group Picker (single group)
Choose a user group using a popup picker window.

Hidden Job Switch
Hidden switch programmatically set whether or not to create a Perforce job.

Job Checkbox

Find More Custom Fields **Previous** **Next** **Cancel**

Come tipo del campo, selezioniamo **Checklist**, che consente di poter generare quello che ci serve.

The screenshot shows a dialog box titled "Configure 'Checklist' Field". It contains two input fields: "Name" with the value "Lista di controllo" and "Description" with the value "Esempio di campo checklist". At the bottom right, there are three buttons: "Previous", "Create" (highlighted in blue), and "Cancel".

Concludiamo la configurazione, indicando una descrizione. A questo punto possiamo definire le voci che compongono la checklist

The screenshot shows a dialog box titled "Edit Lista di controllo for DEMO-1". It features a text input field, a blue link labeled "Add item", and the text "Esempio di campo checklist". At the bottom right, there are two buttons: "Submit" (highlighted in blue) and "Cancel". The background is a blurred view of a software interface with sections like "People", "Assignee:", "Reporter:", "Notes:", "Watchers:", "Created:", "Updated:", "Time Tracking", and "Estimated:".

oppure possiamo entrare nel dettaglio dell'editing della issue e procedere con la generazione della checklist

Edit Issue : DEMO-1

Configure Fields

Summary*

Esempio di issue

Lista di controllo

☐ Passo01
☐ Passo02
☐ Passo03

☒ ☐

Esempio di campo checklist

Issue Type*

+ New Feature

Possiamo, ad esempio, stabilire i passi per eseguire il test e verificare che l'anomalia sia effettivamente risolto. Il risultato è il seguente:

Demo / DEMO-1

Esempio di issue

Edit

Comment

Assign

More

Start Progress

Done

Admin

Details

Type:

+ New Feature

Status:

TO DO (View Workflow)

Priority:

Major

Resolution:

Unresolved

Labels:

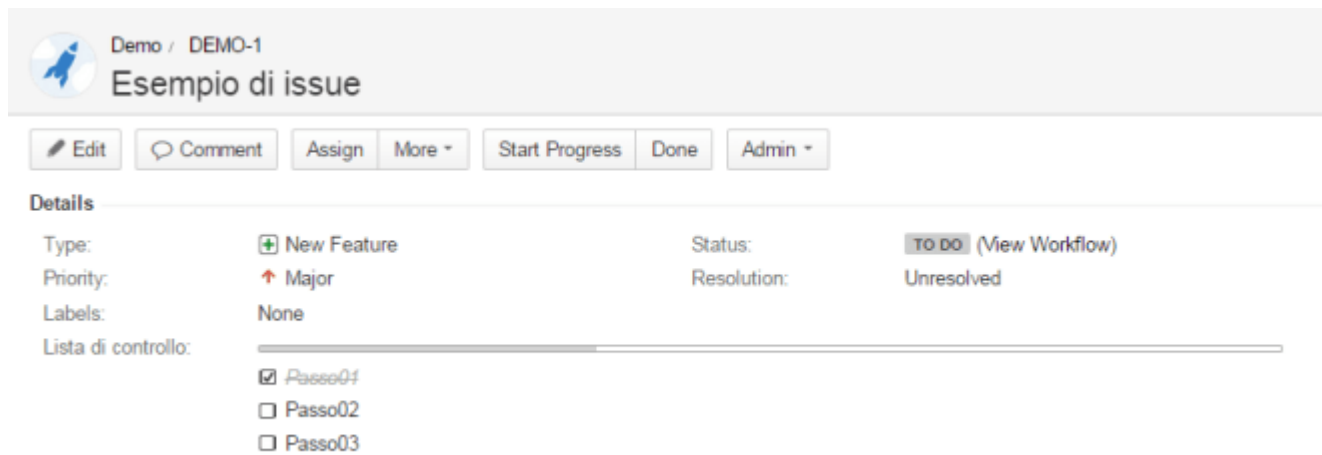
None

Lista di controllo:

☐ Passo01
☐ Passo02
☐ Passo03

Click to edit

Quando si verifica che il passo è rispettato, su può selezionare il passo stesso



The screenshot shows a Jira issue page for 'Esempio di issue' under the 'Demo / DEMO-1' project. The issue is a 'New Feature' with 'Major' priority and 'None' labels. The status is 'TO DO' and the resolution is 'Unresolved'. A progress bar is shown with three steps: 'Passo01' (checked), 'Passo02' (unchecked), and 'Passo03' (unchecked). The progress bar is partially filled, indicating the current progress.

Demo / DEMO-1
Esempio di issue

Edit Comment Assign More + Start Progress Done Admin +

Details

Type: + New Feature Status: **TO DO** (View Workflow)
Priority: ↑ Major Resolution: Unresolved
Labels: None
Lista di controllo:

☒ Passo01
☐ Passo02
☐ Passo03

e come si vede, la barra di avanzamento, si modifica.

Risultato?

Questo addon mette a disposizione uno strumento visuale che, in maniera molto semplice, fornisce una indicazione dell'avanzamento della verifica anomalia. In questo modo, sia chi esegue il test, sia chi esegue la correzione, dispone di una indicazione per capire fino a che punto si è arrivato nella verifica o nello sviluppo.

Le potenzialità sono tantissime.

Conclusioni

Abbiamo un addon che introduce una piccola feature, ma mette a disposizione una funzionalità visuale non indifferente. Nel prossimo post, andremo a vedere il secondo addon della Kenoah.

<https://marketplace.atlassian.com/plugins/com.kanoah.test-mana>

ger

<https://marketplace.atlassian.com/plugins/com.kanoah.jira-checklist>

Differenze Cloud-Server

Cloud o Server?

In questo primo post, di una serie dedicati all'argomento, andremo ad esaminare quali sono le differenze principali tra le due release dei prodotti. Obbiettivo è quello di fornire tutte le informazioni necessarie per operare la scelta migliore, in base alle esigenze.



Andiamo nel dettaglio

Credo che il primo punto da chiarire sia dare una prima spiegazione del perché di queste differenze. Innanzitutto, si

tratta di situazioni differenti. Le versioni cloud (ex [onDemand](#)) sono gestite centralmente dal personale Atlassian e, di conseguenza, alcune funzionalità sono controllate direttamente da loro. Fornirle in mano ai clienti potrebbe essere controproducente in quanto potrebbe arrivare ad impattare altre installazioni.

Quali prodotti sono disponibili in cloud?

Prima di vedere le differenze, vediamo quali prodotti Atlassian sono [disponibili](#).

- ***Confluence***
- ***JIRA***
- ***JIRA Service Desk***
- ***JIRA Agile***
- ***JIRA Portfolio***
- ***Hipchat***
- ***Bamboo***

Adesso vediamo il dettaglio delle varie caratteristiche.

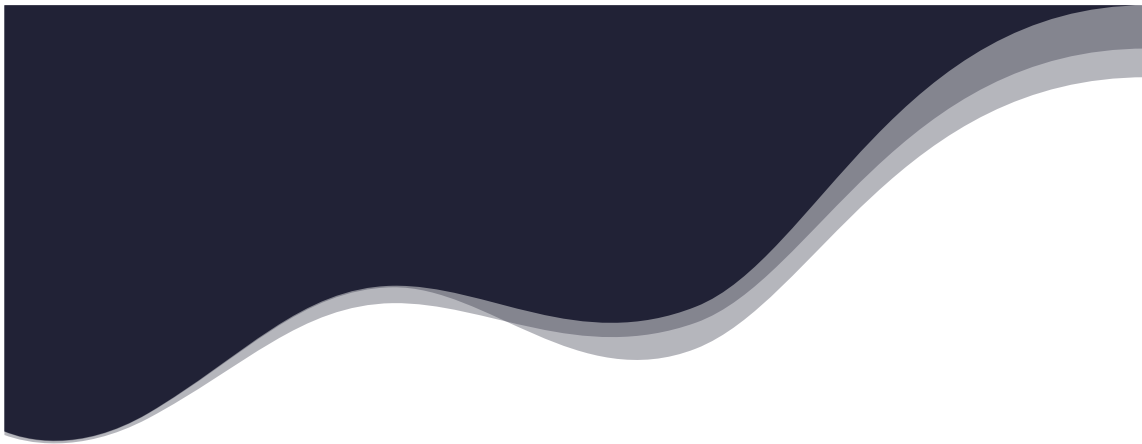
Il ruolo di ***System Administrator*** viene ricoperto dal personale Atlassian. Di conseguenza, tutte le funzionalità che ruotano a tale ruolo, non sono disponibili.



I Backup schedulati non sono possibili, ma è comunque possibile eseguire una esportazione dati, al fine di riportarla su installazioni Server. I backup giornalieri NON sono consentiti in quanto il personale Atlassian li gestisce a livello centrale.

User Macro, HTML Macro e tutto ciò che impatta HTML (come macro standard) è disabilitato. Non esiste altra possibilità di poter abilitare tali addon su Cloud. Tuttavia, sono disponibili degli addon di terze parti che consentono di [includere pagine web di siti terzi](#).

Mail Account e Mail server sono configurati e gestiti a livello centralizzato. Questo per gestire il tutto a livello centrale. Di conseguenza non è possibile customizzare questa parte.



Le versioni cloud utilizzano dei propri layout, ma non risulta possibile inserire delle variazioni ai CSS per la versione Cloud nè personalizzare il layout della versione cloud.



Non tutti gli addon sono disponibili per la versione Cloud. Il motivo è da ricercarsi nel fatto che, essendo la versione cloud centralizzata, la Atlassian ha modificato il come

impostare gli addon per la soluzione cloud. Di conseguenza, anche lo sviluppo deve avvenire in maniera differente. Il seguente [link](#) aiuta a capire quali addon sono, al momento, disponibili. Come ho già avuto modo di evidenziare in post pubblicati, il numero di addon per la versione cloud è notevolmente aumentato nell'ultimo anno. Per la tecnologia da usare, fare riferimento ad [Atlassian Connect](#). dedicheremo post a questo argomento.

Conclusioni

In questo post abbiamo dato una prima scorsa alle differenze tra le due installazioni, dove abbiamo visto quali operazioni sono possibili per una o per l'altra installazione. Questo aiuta sicuramente nella scelta del prodotto che si vuole e nel budget a disposizione. Nei prossimi post andremo a dettagliare le differenze, evidenziandole e commentandole.

Riferimenti

Pagina della documentazione Confluence disponibile [qui](#).

Gestiamo un progetto con i prodotti Atlassian – 3

Introduciamo Confluence

In questo post andremo ad introdurre Confluence nella gestione dei progetti, come elemento di supporto per la gestione dei progetti.



Partiamo subito

Il primo punto da chiarire è quello di definire quale è l'obiettivo di Confluence, nella gestione di un progetto.

Confluence deve essere il contenitore dove posizionare TUTTE le informazioni del progetto. Ovvero:

- Analisi funzionali e tecniche
- Documentazioni tecniche di riferimento
- Verbali di riunioni, sia con il cliente che interne
- **Piani di Battaglia**, ovvero il dettaglio di come sviluppare
- Piani di rilascio delle varie versioni, spiegati in dettaglio.
- Documentazione
- CV del personale
- Template di documenti
- tutto ciò che si ritiene utile per lo svolgimento del progetto ☐

Riassumendo, Confluence deve contenere tutte le informazioni del progetto, ovviamente opportunamente organizzate. ☐ Se si inserisce alla rinfusa, non serve a nulla. Il tutto deve essere organizzato e fruibile da tutto il team di sviluppo.



Organizzazione

La prima cosa che consiglio è di predisporre uno Space per progetto. In questo modo si hanno due vantaggi subito:

- Tutte le informazioni sono presenti in un unico punto
- La gestione delle [permission](#) è molto semplice e veloce

e sono due vantaggi non indifferenti :-D.

Attuando questa organizzazione, si può sfruttare anche un [addon](#) molto importante, che consente di poter generare dei template di Space. Ho già descritto le potenzialità di questo addon su altri [post](#). Le potenzialità sono enormi. Abbiamo la possibilità di creare un template di space, organizzarlo in base alle esigenze richieste o avere anche più template in base alla tipologia di progetti



Possiamo anche impostare i template per le varie pagine che comporranno lo Space del progetto. In questo modo, se dobbiamo redarre il verbale della riunione con il cliente, ci semplifichiamo notevolmente il lavoro.

Apply Page Layouts
Give your templates structure. 2 column, 3 column – the choice is yours

Intuitive Variables
Take the work out of creating content with variables that are easy for anyone to use

Add Labels
Gently enforce labeling of Confluence pages by adding labels to your templates

Insert Task Lists
Encourage users to create tasks to ensure work actually gets done

@Mention Users
Confluence will automatically notify mentioned users when a page is created from a template so you don't have to

Occorre sempre tenere a mente che Confluence deve essere uno strumento di aiuto nella conduzione del progetto, non un peso aggiuntivo che ci limita i movimenti e che non seguiamo perchè lo riteniamo un problema.

Quindi sfruttiamo tutte le funzionalità per far si che sia un valido supporto. ☐

Possiamo anche collegarci dei Database, in modo da visualizzare dei risultati di elaborazioni o delle statistiche o, molto più semplicemente, estrarre dei dati dal gestionale per poterli visualizzare e controllare. In questo modo, Confluence diventa il punto unico di gestione del progetto.

Colleghiamo Confluence e JIRA



Confluence e JIRA si collegano e si parlano senza alcun problema. Per far ciò occorre procedere con una procedura particolare, che si chiama **Application Link**, che mette in comunicazione i due sistemi. In questo modo, possiamo vedere le issue JIRA anche in Confluence :-), come già mostrato nel [post](#) precedentemente pubblicato.

TAG / Pages / TAG Home

TAG-1 - Lenovo Ideapad U510

Informazioni

Nome	Lenovo Ideapad U510
Descrizione breve	Notebook
Processore	Intel CORE i5
RAM	8Gb
Scheda Video	NVIDIA GEFORCE
Porte USB	3 di cui 1 USB3
Descrizione Lunga	-
Immagini	

ASSET su JIRA

 TAG-1 - Lenovo Ideapad U510 ASSEGNATO

Interventi

Key	Summary	T	Created	Updated	Due	Assi
No issues found Refresh						

Drivers

Elenco dei driver per Sistema Operativo

[Windows 7](#)

Cronistoria ed Annotazioni

Data	Annotazione
01.10.2013	Acquistato presso XXXXXXXXXX
Attualmente	In esercizio. Nessuna segnalazione ricevuta

Dalla figura sopra riportata, si vede che è possibile collegare i due sistemi. Per maggiori dettagli si rimanda a questa [pagina](#).

In questo modo, possiamo creare delle pagine ad hoc, dedicate alla documentazione di dettaglio di una release di bug fix. Possiamo avere un dettaglio maggiore o di livello funzionale, per i vari bug. In questo modo, su JIRA abbiamo il dettaglio tecnico (campo di database non scritto e corretto), mentre su Confluence possiamo avere il dettaglio funzionale (provincia non riportata nella anagrafica del cliente). Giusto per fare un esempio ☐

Roadmap aka Piano di battaglia

La macro [Roadmap](#) è sicuramente utile per la gestione di un progetto. La macro offre la possibilità di poter rappresentare un vero e proprio piano di battaglia per gestire il progetto:

- rilasci
- avanzamento lavori
- componenti
- variazioni
- etc etc etc ☐



Come già mostrato nei precedenti post, e dalla immagine sopra riportata, si può vedere che, grazie ad un colpo d'occhio, abbiamo il piano di battaglia, le azioni che saranno intraprese, gli sviluppi predisposti, e tutto ciò che serve.

Questo è di sicuro utile per gestire il tutto ed avere sempre sott'occhio la situazione.

Conclusione

Abbiamo visto in questo post come Confluence diventa, nell'ambito dello sviluppo di progetti, un punto centrale e nevralgico. Abbiamo un unico punto in cui reperire e concentrare le informazioni, senza dover impazzire a cercare il tutto. Verbalì, documenti di analisi, documenti del cliente

trovano perfettamente posto in Confluence e, grazie alle sue funzionalità, si rivela il più importante dei collaboratori di un progetto.

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 abbiamo bisogno, 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.

Graphviz – Prova su strada

Prova su strada

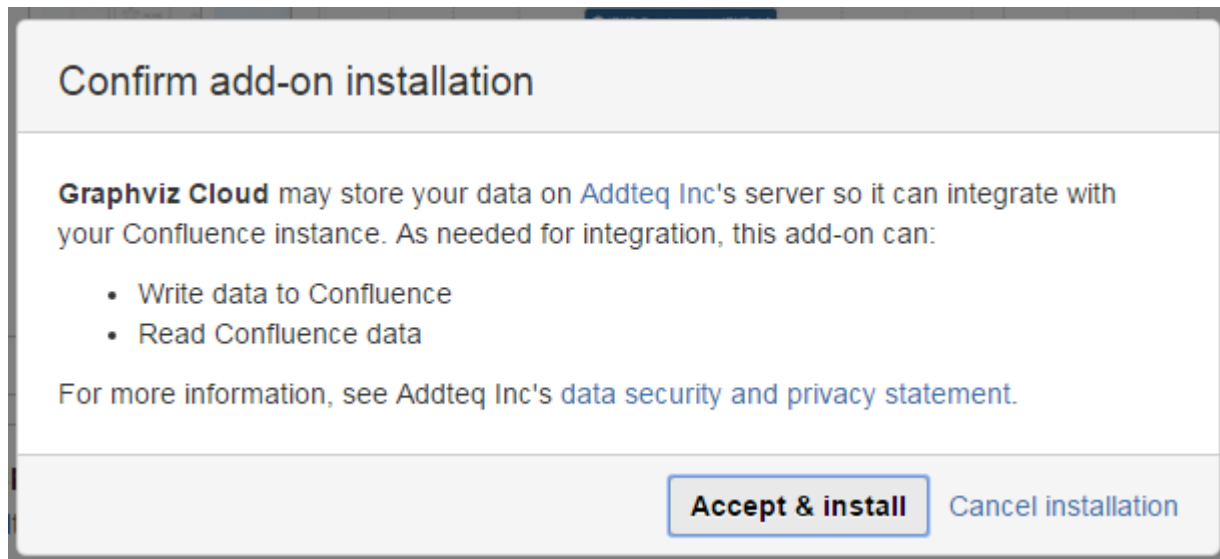
In questo post andiamo a fare la prova su strada dell'addon Graphviz, per Confluence Cloud, di cui abbiamo dato una breve introduzione nel seguente post.



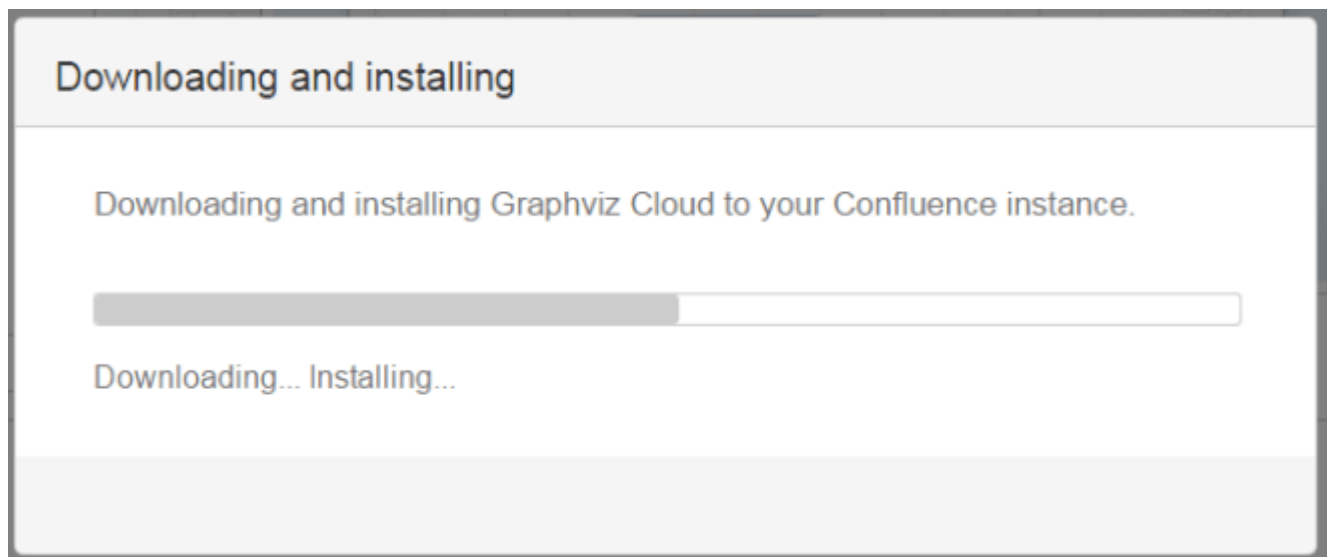
Procediamo l'installazione

con

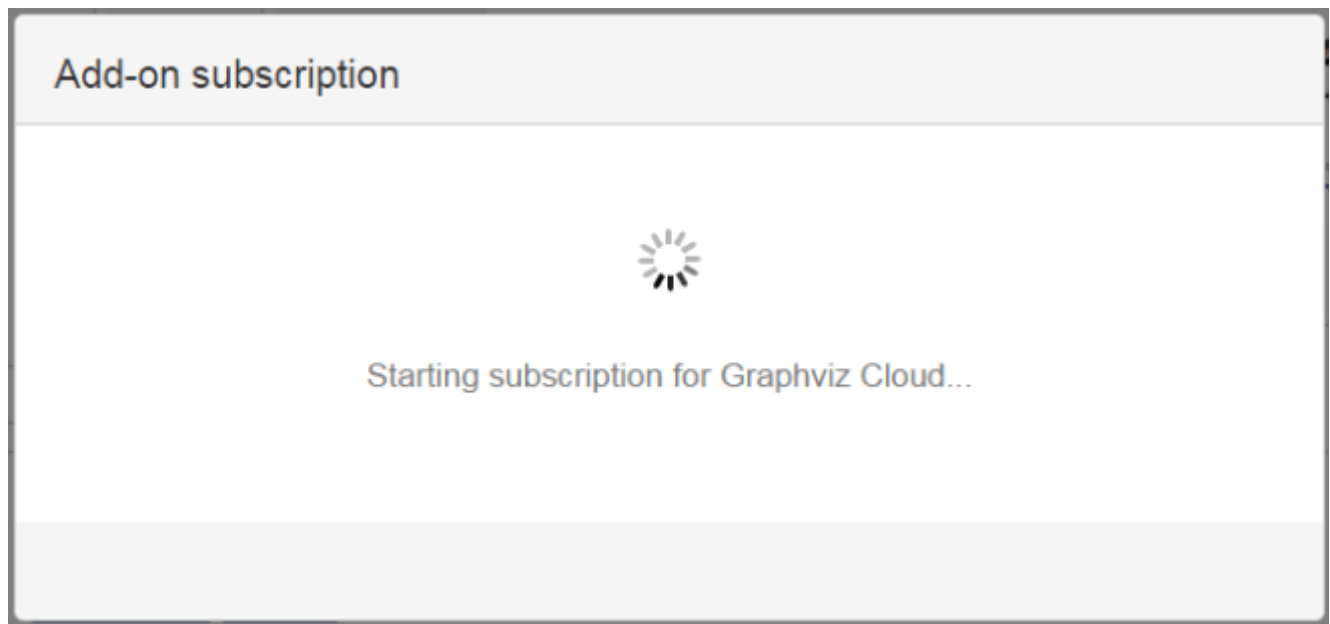
Installiamo l'addon, come siamo già abituati. Selezioniamo l'addon dalla apposita sezione di amministrazione degli addon di Confluence:



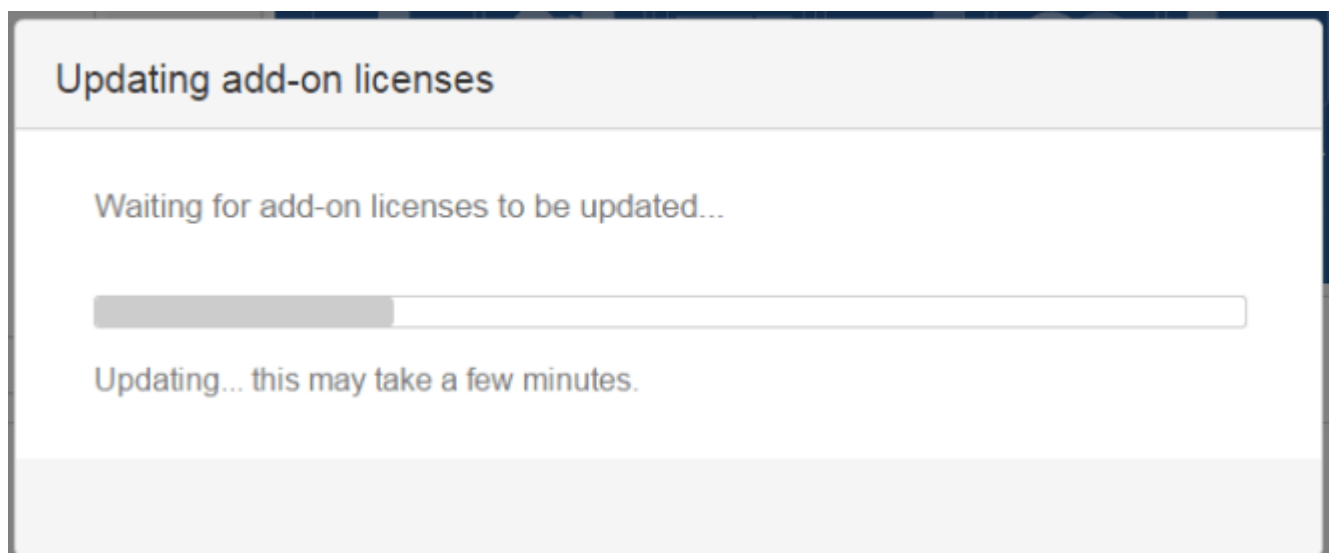
Selezioniamo **Accept & Install** e procediamo con l'installazione



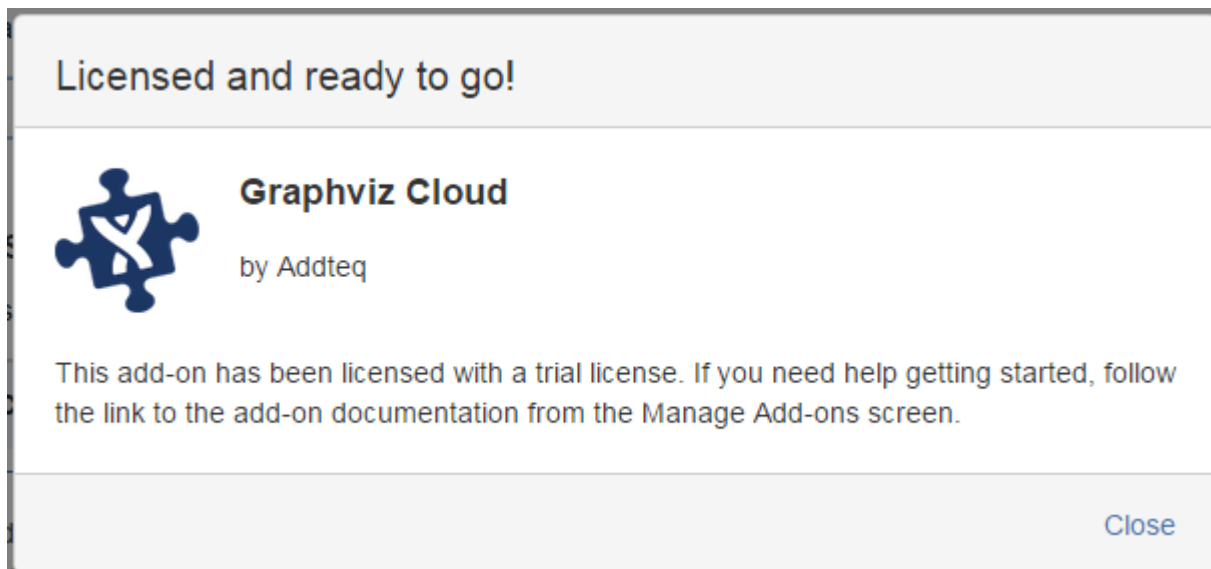
L'addon viene così installato sulla nostra installazione cloud. Viene quindi impostata la licenza



dopo qualche minuto

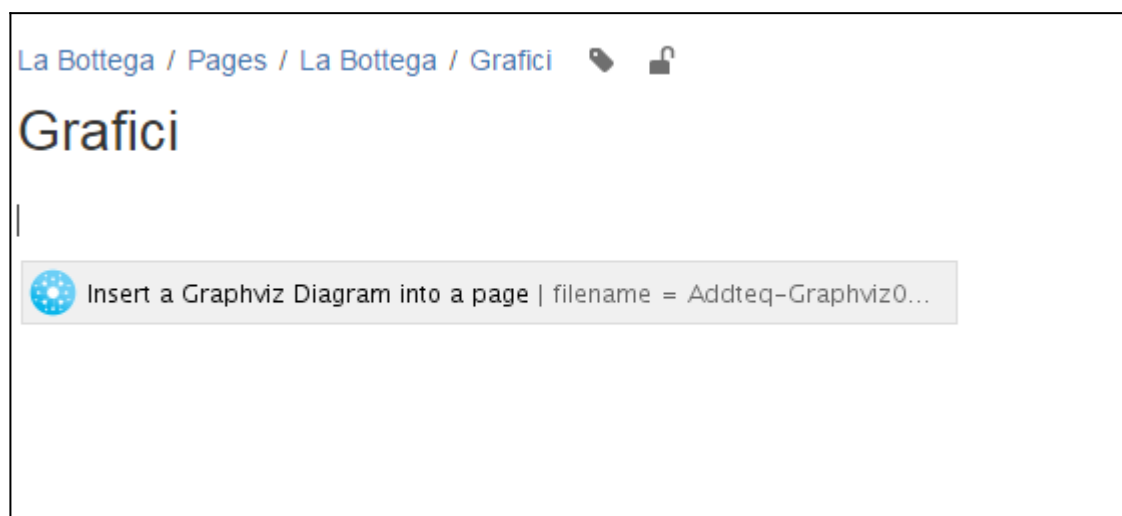


la licenza è disponibile e l'addon è pronto all'uso.



Test di utilizzo

A questo punto, creiamo la nostra pagina di prova e procediamo con il test.



Settiamo il nostro codice:

```

1 digraph G {
2
3   subgraph cluster_0 {
4     style=filled;
5     color=lightgrey;
6     node [style=filled,color=white];
7     a0 -> a1 -> a2 -> a3;
8     label = "process #1";
9   }
10
11  subgraph cluster_1 {
12    node [style=filled];
13    b0 -> b1 -> b2 -> b3;
14    label = "process #2";
15    color=blue

```



Preview

Il codice che abbiamo impostato è il seguente

```

digraph G {

  subgraph cluster_0 {
    style=filled;
    color=lightgrey;
    node [style=filled,color=white];
    a0 -> a1 -> a2 -> a3;
    label = "process #1";
  }

  subgraph cluster_1 {
    node [style=filled];
    b0 -> b1 -> b2 -> b3;
    label = "process #2";
    color=blue
  }

  start -> a0;
  start -> b0;
  a0 -> b0;
  a0 -> b1;
  a1 -> b3;
  a1 -> b2;
  a1 -> end;
  b2 -> a3;
  a3 -> a0;
  a3 -> end;
  b3 -> end;

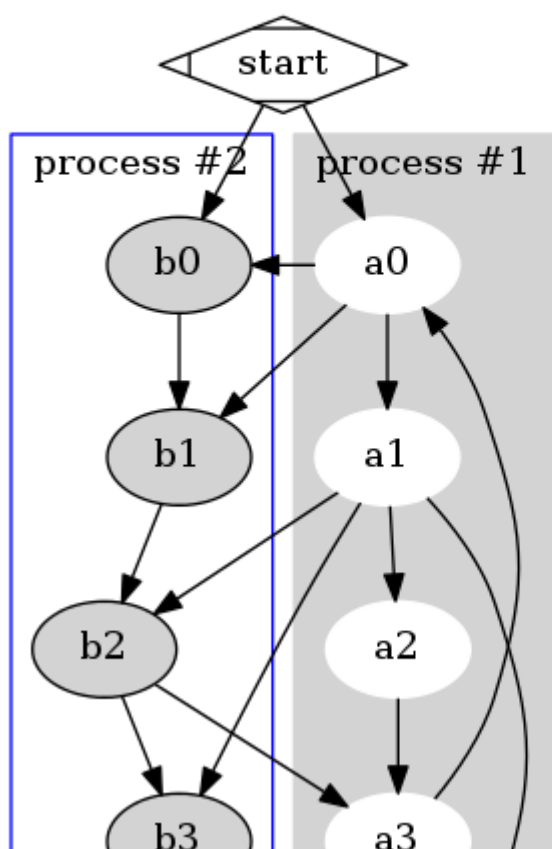
```

```
start [shape=Mdiamond];  
end [shape=Msquare];  
}
```

e come si vede non è difficilissimo da impostare. Il risultato è il seguente:

Grafici

Created by Fabio Genovese [Administrator], last modified about 7 hours ago



Conclusioni

Abbiamo un addon per disegnare dei grafici interessanti. Per coloro che sono abituati a scrivere codice, questo addon può essere un buon compromesso per poter disegnare dei grafi da inserire nelle pagine Confluence. Per coloro che non sono ... sviluppatori, questo addon potrebbe risultare un po' ostico e

di difficile utilizzo. Occorre quindi che gli utilizzatori lo provino, in modo da rendersi conto se è l'addon che fa per loro.

Kanoah – Addons per JIRA

Addons per JIRA

In questo post andremo ad esaminare due addon per JIRA, uno che consente di poter impostare delle Checklist nelle issue, ed uno che consente di poter impostare dei test case, sfruttando le potenzialità di JIRA.



Checklist

Uno degli addon prodotti, consente di poter impostare delle checklist all'interno delle ISSUE di JIRA.



Inserire delle checklist in una issue permette di poter inserire nell'issue stessa a modalità per poter verificare la effettiva risoluzione delle anomalie.



Con questo sistema è possibile inserire un sistema di verifica della anomalia, a prova di tutto. In questo modo, chiunque si occupi del system test è in grado di eseguire le verifiche del caso per poter verificare se il problema è stato

effettivamente risolto.



Test Case

Un altro add-on che la Kanoah mette a disposizione, consente di poter creare dei test case e associarli alla issue JIRA.



Questo add-on si integra perfettamente con JIRA, estendendone le funzionalità consentendo di poter aggiungere i test case necessari per verificare lo sviluppo. Questo add-on è stato studiato in modo da poter fornire una integrazione importante tra chi si occupa del system test e lo sviluppo.



Conclusioni

SI tratta di due addon, per il momento solo per installazioni server, che consentono di poter migliorare il system test ed il lavoro del personale addetto ai test. Nei prossimi post, andremo a fare la ... prova su strada, ma sono sicuro che non deluderanno le aspettative ☐

Riferimenti

Riferimenti agli addon:

- [checklist](#)
- [test](#)