

JExcel PRO – First LOOK

JExcel PRO – First Look

In questo post andremo ad esaminare la nuova versione PRO di [JExcel](#), addon molto interessante che fonde le funzionalità di un foglio elettronico con le potenzialità di JIRA. Andiamo a vedere quali sono le novità ☐



Andiamo al sodo

Andiamo subito ad esaminare le caratteristiche salienti di questo addon, andando subito al sodo :-), anche se JExcel PRO è sempre in continua evoluzione.

ATLASSIAN

Come dice la stessa parola PRO, si tratta della estensione della versione LITE, che abbiamo esaminato nei post:

- [JExcel – Un addon moooooooooolto interessante](#)
- [JExcel – Prova su strada](#)

che presenta anche il seguente elenco di ... *features*:

- Esportazione con caratteristiche avanzate (includono ***attachments, immagini e commenti***)
- Collaboration
- Cell History
- Priority Support
- More frequent releases
- Excel like UI with in-place edition and issue creation
- Copy-paste (also to/from MS Excel/Notepad)
- Average/Count/Sum of the selected cells
- Column renaming
- Excel like column filtering and sorting
- Dealing with thousands of issues without page reload (paging, filtering, sorting)
- Column freeze (importantissima, salva da tantissime situazioni ☐)
- Tab coloring (Importante anche
- Flag rows (come in Outlook)



Caratteristica importante è quella di avere a disposizione delle operazioni di *copia/incolla* direttamente dalle celle di Excel ☐

La collaboration permette di poter vedere le modifiche in tempo reale, come mostrato dalla precedente immagine.



Le caratteristiche avanzate di esportazione, permettono di avere a disposizione TUTTE le informazioni delle varie issue.



Possiamo assegnare un colore ai worksheet, per meglio identificarli :-). Lo stile richiama molto MS Excel. Anche questo aiuta tantissimo nella gestione.

Conclusioni

Siamo di fronte ad un addon molto importante. JExcel LITE dispone di features molto importanti, ma la versione PRO mette a disposizione delle features che rendono questo addon interessantissimo :-).

Raccomandazioni

We strongly recommend to use a Chrome, Firefox or Safari browser when using JExcel!

Questa è una raccomandazione che abbiamo sicuramente già visto in tante altre situazioni, e che sicuramente fa parte della nostra esperienza. Ma sono sicuro che la [MORESIMP](#) saprà sorprenderci con altrettanti effetti speciali ☐

Reference

Maggiori informazioni sono presenti [qui](#).

JExcel LITE – Ultime news

Ultime news su JExcel

In questo post andremo ad esaminare le ultime novità sulla versione LITE di JExcel. Abbiamo già avuto occasione di parlare di questo ottimo addon in:

- [JExcel – Un addon mooolto interessante](#)
- [JExcel – Prova su strada](#)



Dettaglio, Dettaglio

Dettaglio,

Andiamo subito al sodo. Nell'ultima versione viene rilasciata la seguente *feature*:

- ***Delete issue*** in context menu



Abbiamo anche la risoluzione di bugfix minori, ma che sono sicuramente ci aiutano nella vita di tutti i giorni:

- Link problems risolti ☐
- dialogs troppo piccoli
- Altri bugfix minori



Conclusione

Abbiamo visto le ultime novità di questo addon. Continuiamo a monitorare con interesse questo addon: Sono sicuro che ci riserverà delle bellissime sorprese.

Reference

Maggiori informazioni sono presenti [qui](#).

**Uno studio in JIRA – Scenario
e possibili soluzioni**

Uno studio in JIRA

In questo post cercheremo di descrivere un possibile scenario, dettato da una necessità reale, cui successivamente cercheremo di fornire alcune possibili soluzioni.



JIRA esposto verso i clienti

Partecipando ad un evento, mi fu indicata questa situazione: Si voleva esporre JIRA verso dei clienti. Tuttavia esisteva sempre il timore che i clienti potessero anche accedere, oltre ai progetti loro esposti, anche altri progetti, cui NON DOVEVANO ACCEDERE.

Si trattava ancora di JIRA 6 e JIRA 7 era appena stato rilasciato.



La necessità è reale. Avere a disposizione un unico JIRA è sicuramente utile. Non abbiamo la necessità di dover gestire più istanze e non ci complichiamo la vita con troppe operazioni di amministrazioni.

Quali azioni possiamo intraprendere?

Sicuramente andiamo a gestire i permessi ed i relativi [permission scheme](#) associato al progetto.

Project Permissions	Users / Groups / Project Roles	Operations
Administer Projects Ability to administer a project in JIRA.	Project Role (Administrators) (Delete)	Add
Browse Projects Ability to browse projects and the issues within them.	Project Role (Big Kids) (Delete)	Add
View Issue Source Tab	Project Role	Add

In quest modo riusciamo a stabilire chi fa cosa e perché. In aggiunta, lavorando con i Roles del progetto, stabiliamo i ruoli degli utenti.

ATLASSIAN

Questo sicuramente aiuta nel separare le persone dalle informazioni che possono leggere da quelle che non devono reperire.

Utile, ma??

Questo è sicuramente utile, ma tenendo tutto in un unico JIRA esiste sempre la possibilità che:

- una configurazione possa scappare e di conseguenza delle informazioni possono essere lette;
- si potrebbe sfruttare dei buchi di sicurezza (ce ne sono stati e ce ne saranno sempre) ed accedere a sezioni private;
- se abbiamo un numero molto alto di utenti da gestire, l'errore aumenta.



Ok. Che cosa possiamo fare?

Possiamo sfruttare le API REST che i prodotti della Atlassian mettono a disposizione. In questo modo possiamo mettere a disposizione una applicazione, ad uso interno, per mettere a disposizione degli utenti ciò di cui hanno bisogno, e niente ALTRO.

Le API sono molto complete e permettono di poter fare... praticamente tutto. Di conseguenza abbiamo una applicazione opportunamente bloccata e configurata. Il pericolo delle falle di sicurezza è limitato..... forse :-0



Tuttavia, questo significa che occorre SVILUPPARE e MANUTENERE questa applicazione. Di conseguenza aumentano i costi e i

tempi. Questo è un aspetto da non sottovalutare.

La facciamo più difficile? ovviamente sì

Se oltre a JIRA volessimo esporre anche JIRA Service Desk? Questo complica la vita. La versione 6 di JIRA differisce parecchio dalla versione 7 di JIRA, come abbiamo mostrato nei precedenti [post](#) e nei post successivi.

Nella versione 6, JIRA Service Desk è un ADDON di JIRA. Quindi, se si vuole esporre JIRA Service Desk abbiamo comunque lo stesso problema.



In questo caso, le API di JIRA Service Desk sono state introdotte solo dopo. Di conseguenza la strada di un applicativo a se stante non è percorribile.

Che alternative abbiamo?

Se non abbiamo a disposizione di un team di sviluppo, se siamo magari una azienda medio piccola e l'informatica non è il core business, che cosa possiamo fare? Cercheremo di dare delle risposte nei prossimi post.

Conclusione

Abbiamo introdotto un nuovo argomento molto delicato ma

mooolto interessante ☐ . Cercheremo di svilupparlo molto bene nei prossimi post.

Estendiamo JQL con altre funzionalità

Estendiamo le funzionalità JQL

In questo post andremo ad esaminare come possiamo ancora [estendere](#) le funzionalità di [JQL](#) Standard.



Come possiamo fare?

Vediamo quali possibilità abbiamo per estendere le funzionalità JQL nel nostro JIRA. Abbiamo già visto alcuni esempi esaminando diversi addon, che abbiamo recensito nel corso del tempo, quali ad esempio [scriptrunner](#), di cui abbiamo ampiamente parlato. vediamo adesso quali altre soluzioni sono possibili:



[JQL Tricks Plugin](#) è sicuramente una prima soluzione. Mette a disposizione tante funzioni aggiuntive per il JQL, permettendo di poter eseguire moltissime interrogazioni specifiche.



Abbiamo la possibilità di monitorare chi ha eseguito determinate operazioni in maniera banalissima, come mostrato dalla precedente immagine.



In aggiunta abbiamo la possibilità di poter indicare quali utenti/gruppi/progetti ne possono fare uso, con una agevole form di configurazione.



[Craftware Search Linked Issues for JIRA](#) è la nostra seconda indicazione. Si tratta di un addon che estende, a sua volta, il JQL con opportuni funzioni. Permette di poter eseguire delle ricerche sulle linked issue in maniera molto semplice.



Dalla immagine precedente, vediamo che abbiamo delle funzioni che, nella vita lavorativa, ci aiutano a semplificarcela :-D.



[JQL Functions Collection](#) mette a disposizione un insieme di funzioni JQL anche queste molto semplificative della vita di ogni giorno.



Si tratta di funzioni che trattano formato di dati, di stringhe etc. Queste sono sicuramente molto utili :-), come mostrato dalla immagine successiva.



Conclusioni

Abbiamo fatto una carrellata molto rapida di addons che permettono di estendere JQL con altre funzioni che possono aiutare la vita di tutti i giorni.

Novità su JIRA Software

Ultime novità su JIRA

Software

Prosegue la serie di post dedicati alle ultime novità dei prodotti della Atlassian. Ci occupiamo adesso di JIRA Software.



Cosa abbiamo di nuovo?

Iniziamo con la prima schermata che viene visualizzata all'inizio di tutte le attività è la seguente:



Tutte le prime schermate che venivano proposte, sono state unificate nella precedente schermata, che risulta di più agevole e rapida nell'utilizzo.



Per dare maggiore flessibilità al team Agile, viene messa a disposizione la possibilità di poter gestire e delegare le responsabilità delle Sprint. Di conseguenza, è stata introdotta una nuova permission: ***manage sprint***; come mostrato nella precedente Gif.

Questo ovviamente presuppone una grande fiducia ed un team molto affiatato, altrimenti questa opzione potrebbe generare delle situazioni molto inconsistenti.



Altra novità è quella di ***Flag and comment*** una issue contemporaneamente. In questo modo è possibile marcarla come importante (Flag) e successivamente inserire un commento. Si tratta di una piccola feature, ma utile per coloro che lavorano tutti i giorni su questi punti.

Conclusioni

Si tratta di piccole modifiche, ma che semplificano notevolmente il lavoro.

References

Maggiori informazioni sono disponibili [qui](#).

Component & Subcomponents – Alternative

Quali alternative esistono?

In questo post andremo ad esaminare quali alternative sono disponibili per l'addon descritto [qui](#).Stiamo parlato di

[Component/Subcomponent/Bundle Versions.](#)



Che cosa offre?

Questo addon mette a sua volta a disposizione la possibilità di poter inserire una gerarchia di componenti



direttamente dallo screen del dettaglio in maniera molto agevole. In aggiunta abbiamo anche la possibilità di associare delle componenti a specifiche versioni



in maniera molto semplice, da quanto mostra la scheda del Marketplace



Inoltre abbiamo la possibilità di inserire nei Worklow Validator la possibilità di poter eseguire dei controlli sulle

versioni da associare in fase di risoluzione di una issue



In aggiunta abbiamo una estensione del JQL per poter ...
referenziare queste informazioni



Conclusioni

L'addon si presenta con delle funzionalità di tutto rispetto.
Sono molto interessanti e consentono di poter implementare
delle features molto carine.

References

Maggiori informazioni sono reperibili [qui](#).

Component & Subcomponents – Test su strada

Test su strada

In questo post andremo a fare il test di questo addon, cercando di trovare ciò che ci serve.



Installazione

Partiamo sempre dalla installazione del nostro addon, che come sempre non ci deve spaventare :-). In questo caso andremo ad utilizzare una installazione di **JIRA CORE**. Selezioniamo il nostro addon e procediamo, selezionando **Free trial** dalla nostra schermata di **Manage Addons**.



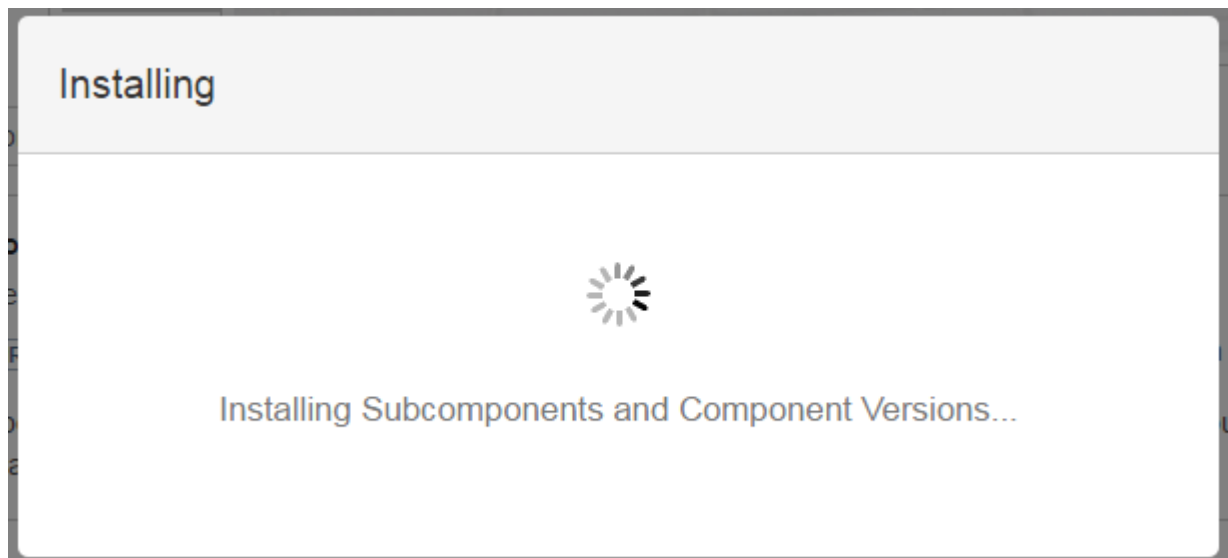
Subcomponents and Component Versions
Broken Build • Vendor supported
PROJECT MANAGEMENT

★★★★☆ (17)
207 installations
Paid via Atlassian

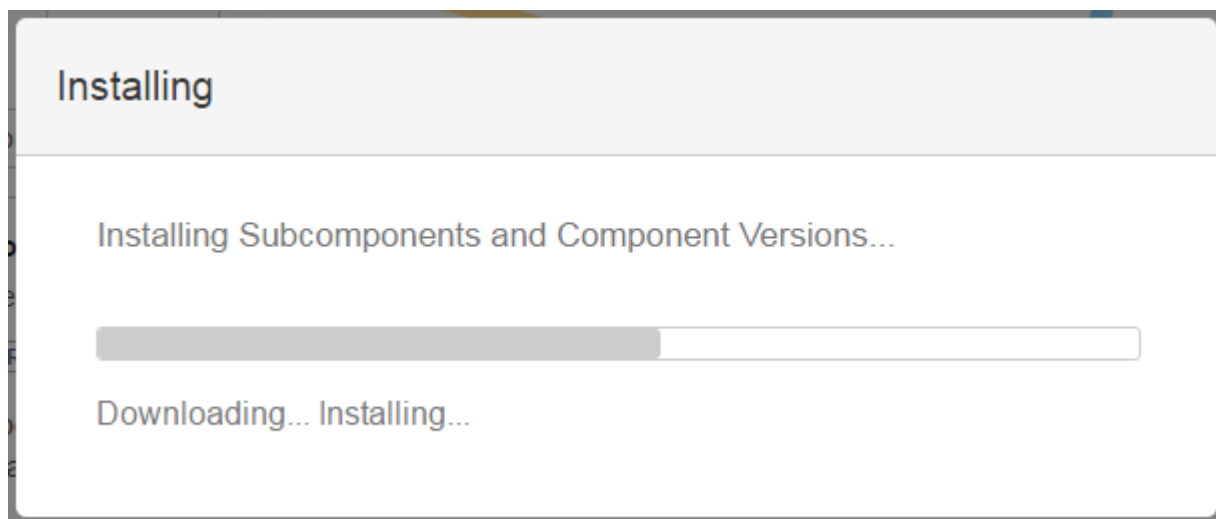
[Free trial](#)
[Buy now](#)

Add more granularity to your project with subcomponents hierarchy, plan releases on a component level with component versions

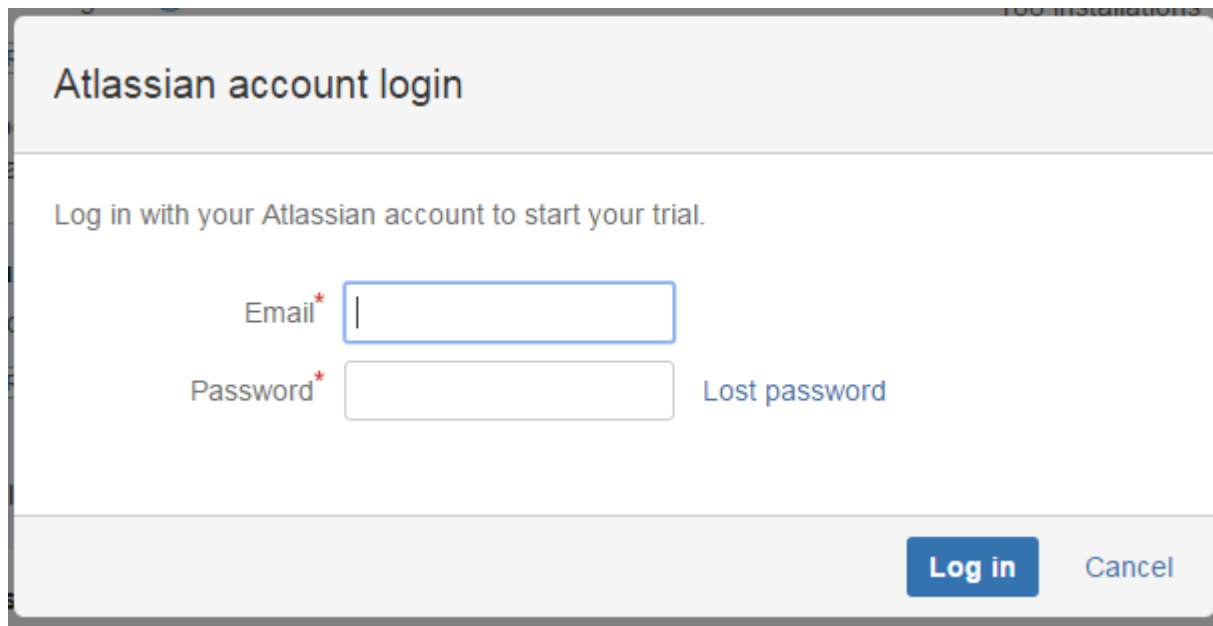
Lasciamo andare la nostra installazione...



... facciamo eseguire il download e la relativa installazione ...



... inseriamo le credenziali per accedere al ***my.atlassian.com*** ...



Atlassian account login

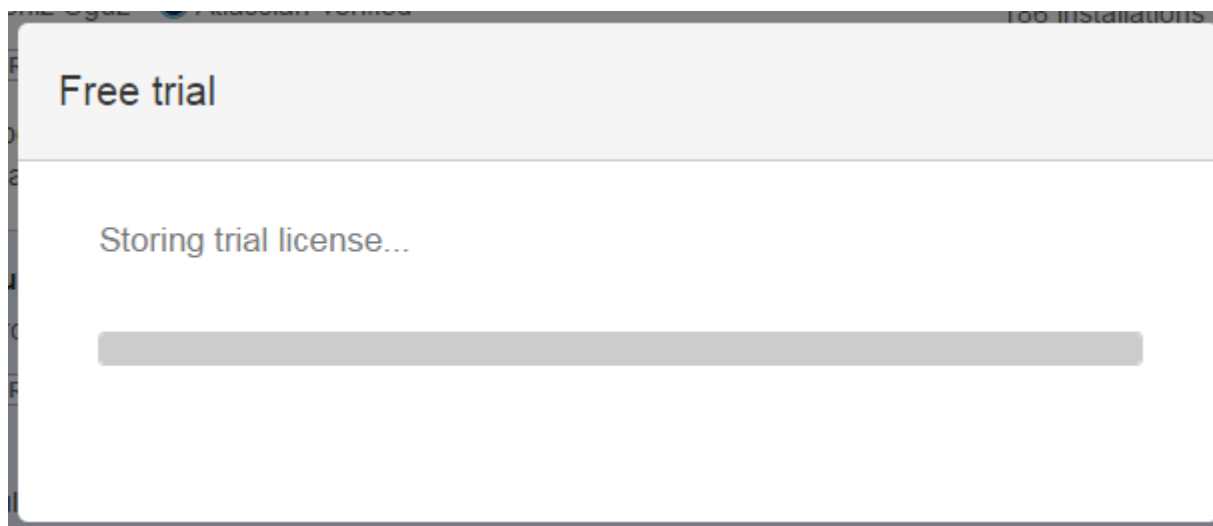
Log in with your Atlassian account to start your trial.

Email*

Password* [Lost password](#)

Log in Cancel

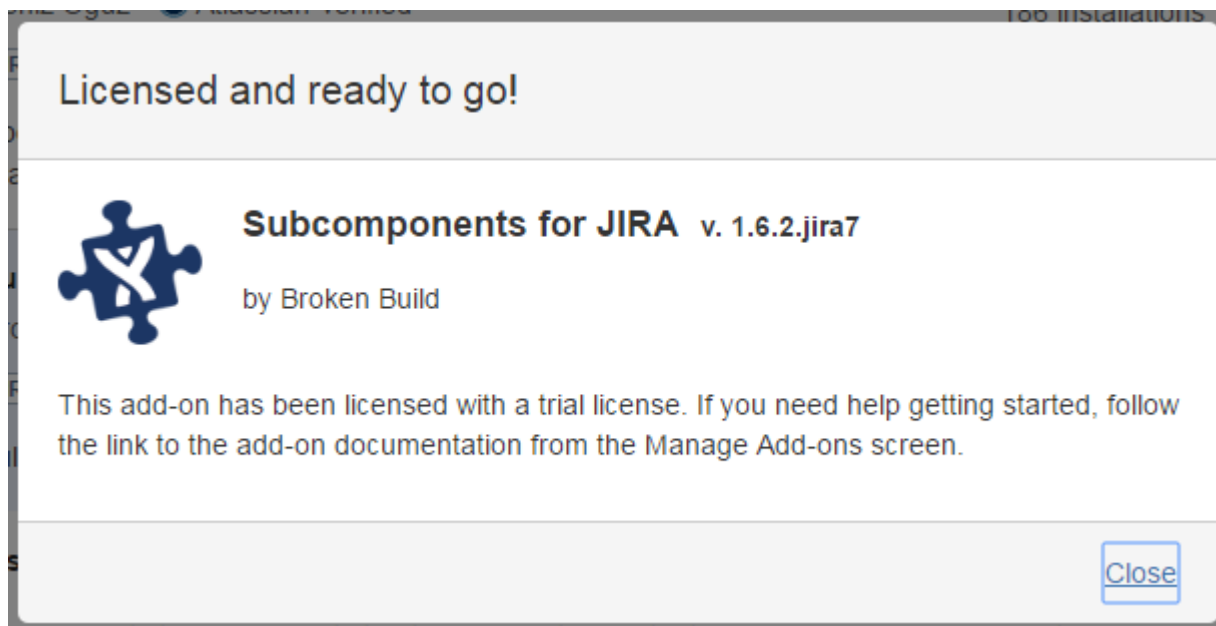
... lasciamo che installi la licenza ...



Free trial

Storing trial license...

... ed il nostro addon è pronto all'uso.



Configurazione

Passiamo alla fase successiva: Come configuriamo questo addon?? La prima cosa che notiamo è che non è presente alcuna sezione di configurazione generale, nella sezione addon. Troviamo la sezione di configurazione tra le opzioni di configurazione del progetto, trattandosi di una configurazione specifica di un progetto.



Esempio

Key: ES - Lead:  JIRA Administrator - Project

[Back to project](#) **Administration**

Summary

[Issue types](#)

- Sub-task
- Task

[Workflows](#)

[Screens](#)

[Fields](#)

[Versions](#)

[Components](#)

[Subcomponents](#)

[Component Versions](#)



Issue

Keep tra
type can

Scheme:
ES: Tas

☒ Task

☐ Sub-t



Workfl

Issues c
defines t
"Resolve

Scheme:

Tentiamo una prima configurazione sul nostro progetto di ***Esempio***. Supponiamo di avere delle componenti che identificano il nostro sistema da gestire.



Esempio
 Key: ES - Lead:  JIRA Administrator - Project type: Business - Category: None - URL: No URL

[Back to project](#)
Administration

[Summary](#)

[Issue types](#)
 - [Sub-task](#)
 - [Task](#)

[Workflows](#)
[Screens](#)
[Fields](#)

[Versions](#)
Components
[Subcomponents](#)
[Component Versions](#)

[Users and roles](#)
[Permissions](#)

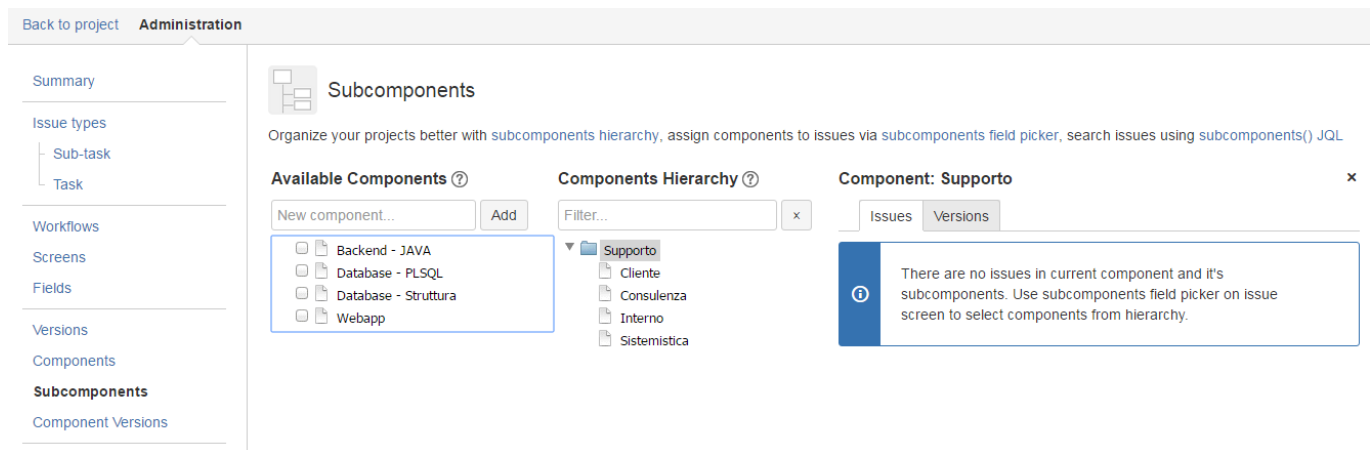

Components

Projects can be broken down into components, e.g. "Database", "User Inte

	Name	Description	Cor
			
	Supporto	richieste di supporto / manutenzione	
	Backend - JAVA	Definizione del backend Java	
	Database - PLSQL	Definizione delle Stored procedure	
	Database - Struttura	Definizione della struttura del DB	

Supponiamo di avere la necessità di distinguere le richieste di supporto, quali segnalazioni utente o segnalazione divisione interna, etc. Se aggiungiamo nuove componenti, potremmo avere delle difficoltà. Il numero delle componenti si moltiplicherebbe e alla fine si avrebbe difficoltà nel censire ed usare, nonché nell'identificare.

Creiamo le nostre sottocategorie selezionando l'opzione **Subcomponent**. Viene mostrata la form per compilare la nostra gerarchia.



Basta un semplice drag 'b' drop per costruire la gerarchia. Possiamo anche estendere direttamente le nostre componenti e agevolmente nella gerarchia.

Ma le componenti?

Che cosa succede alle nostre componenti? Come sono organizzate? Fondamentalmente l'addon si innesta nel nostro JIRA e gestisce come un ulteriore meccanismo che inserisce la gerarchia e la gestisce. Infatti, se andiamo a vedere la componenti, vediamo che sono aumentate, come mostrato in figura.

				Project Default (Unassigne ▾)	Add
	Backend - JAVA	Definizione del backend Java		Project Default	Delete
	Cliente			Project Default	Delete
	Consulenza			Project Default	Delete
	Database - PLSQL	Definizione delle Stored procedure		Project Default	Delete
	Database - Struttura	Definizione della struttura del DB		Project Default	Delete
	Interno			Project Default	Delete
	Sistemistica			Project Default	Delete
	Supporto	richieste di supporto / manutenzione		Project Default	Delete
	Webapp	Front end web		Project Default	Delete

Come *usiamo* questa gerarchia?

Adesso vediamo come usarla. Creiamo una nuova Issue e andiamo ad inserire la nostra gerarchia di componenti

Open Issues [Switch filter](#) ▾

Order by Priority ▾ ▾

☒ ES-7
Questo è un esempio

☒ ES-4
Esempio 4

☒ ES-6
esempio 6

☒ ES-5
esempio 5

☒ ES-3
Esempio 3

☒ ES-2
Esempio 2

☒ ES-1
Esempio 1

 Esempio / ES-7

Questo è un esempio

Edit

Comment

Assign

More ▾

Done

Admin ▾

Details

Type: ☒ Task

Status: **TO DO** (View Workflow)

Priority: ↑ Medium

Resolution: Unresolved

Component/s:

Labels: None

+

✓

✕

Description

Task di esempio

Se andiamo ad editare il campo **Component/s** notiamo, come mostrato dalla figura precedente, una nuova icona che indica la gerarchia.

Subcomponents: ✕

▼

☐ Supporto

☐ Cliente

☐ Consulenza

☐ Interno

☐ Sistemistica

Use context menu for smart selection

Apply

Cancel

Se lo selezioniamo possiamo andare a selezionare le nostre componenti.

Subcomponents:

☒

Supporto

☐

Cliente

☒

Consulenza

☐

Interno

☐

Sistemistica

Use context menu for smart selection

Apply

Cancel

Se andiamo a confermare, il risultato è il seguente:

Open Issues

Switch filter

Order by Priority

☒ ES-7

Questo è un esempio

☒ ES-4

Esempio 4

☒ ES-6

esempio 6

☒ ES-5

esempio 5

☒ ES-3

Esempio 3

☒ ES-2



Esempio / ES-7

Questo è un esempio

Edit

Comment

Assign

More

Done

Admin

Details

Type:

☒ Task

Status:

TO DO (View Workflow)

Priority:

↑ Medium

Resolution:

Unresolved

Component/s:

Consulenza, Supporto

Labels:

None

ovvero vediamo le due componenti presenti, ma se andiamo in editazione del campo, abbiamo comunque la visione della gerarchia.

Che altro?

Se andiamo nella screen di dettaglio, possiamo vedere che in basso a destra abbiamo il dettaglio delle subcomponents.

Esempio / ES-7
Questo è un esempio

Edit Comment Assign More Done Admin Export

Priority: Medium Resolution: Unresolved
Component/s: Consulenza, Database - Struttura, Sistemistica, Supporto ...
Labels: None

Reporter: JIRA Administrator
Votes: 0
Watchers: 1 Stop watching this issue

Description
Task di esempio

Attachments

Structure

Activity
All Comments Work Log History Activity

There are no comments yet on this issue.

Comment

Dates
Created: 36 minutes ago
Updated: Just now

HipChat discussions
Do you want to discuss this issue? Connect to HipChat.
Connect Dismiss

Subcomponents
Supporto
Consulenza
Sistemistica

Conclusioni

Abbiamo visto un esempio di utilizzo di questo addon. Abbiamo visto che cosa fa e come lavora. Sicuramente ci sarà utile per meglio classificare le nostre componenti. Questo componente è disponibile sia per installazioni **Server** che per le installazioni **cloud**.

References

Maggiori informazioni sono reperibili su:

- [Server](#)
- [cloud](#)

Gerarchia di Versioni in JIRA – First look

Altro addon interessante

In questo post andremo a vedere un altro addon molto interessante, prodotto sempre da [Broken Build](#). Abbiamo già visto un altro addon (T0 D0: Inserire il link).



Che cosa offre?

Questo addon permette di poter creare delle gerarchie di versioni, molto utili nello sviluppo di software.



Come notiamo, abbiamo lo stesso layout dell'addon già esaminato (T0 D0: Inserire il link al primo post).



Si integra molto con le funzionalità standard di JIRA, come possiamo ben vedere ☐



Vediamo che questo addon ci permette di poter generare delle gerarchie di versioni.



Possiamo anche eseguire le ricerche attraverso JQL ☐

Conclusioni

Abbiamo visto un addon veramente interessante, soprattutto per coloro che sviluppano software, in particolare per organizzare meglio il tutto.

Components & Subcomponents – First look

Estendiamo le componenti

In questo post andremo ad esaminare un nuovo addon, che permette di poter gestire meglio le componenti, aggiungendo una marcia in più: ***una gerarchia***.



First look



Questo addon permette di poter estendere la piatta lista delle componenti, consentendo di poter creare delle gerarchie di componenti e sottocomponenti.



Riusciamo facilmente ad associare le sottocomponenti attraverso una treeview molto semplice da usare.



Riusciamo anche ad eseguire delle ricerche per ...
sottocomponente □



La stessa view della issue viene estesa in maniera opportuna,
consentendo la visualizzazione di entrambe le nuove
informazioni.



Queste informazioni sono facilmente fruibili anche sotto **JIRA
Service Desk** .



Conclusione

Abbiamo scoperto un nuovo addon che permette di poter gestire
in maniera più attinente alla realtà le componenti di un
progetto. Possiamo, grazie a lui, creare delle gerarchie e
costruire meglio le nostre issue. Nei prossimi post andremo a
testarlo sul campo e vedremo anche meglio le caratteristiche.

Reference

Maggiori informazioni sull'addon sono presenti [qui](#).

Una piccola review di addon per JIRA

Una piccola review

In questo post andremo ad esaminare alcuni addons per JIRA.

Iniziamo...



Partiamo con il primo addon: [Enlarge Your Excel Export](#), un addon che ci aiuta a superare un limite che ci tocca quando andiamo a eseguire una esportazione su foglio elettronico.

Nelle grandi istanze di JIRA, gli amministratori possono limitare la visualizzazione delle issue. Per esperienza posso affermare che molto spesso queste limitazioni aiutano gli amministratori ma non aiutano affatto gli utenti che (ebbene si lo ammetto), vogliono sempre estrarre tutti i dati per fare poi loro le elucubrazioni ☐



molto ci aiuta.

Proseguiamo con il secondo addon [New Status Colors](#), che permette di creare un effetto grafico interessante. Questo può essere di aiuto quando abbiamo bisogno di un colpo d'occhio sulle nostre Issue. Questo addon permette di aggiungere una nota di colore che



basta semplicemente creare degli stati con un nome specifico.
Ad esempio:

- **[red]** per ***Cancel*** oppure ***Cancelled***, ***Suspend*** oppure ***Suspended***, ***Reject*** oppure ***Rejected***
- **[orange]** per ***Wait*** oppure ***Waiting***, ***Hold***, ***On Hold***, ***Pause*** o ***Paused***



Proseguiamo con [Standalone Project Template](#), un addon che ci può sicuramente aiutare nella creazione di nuovi progetti. Consente, infatti, di poter generare dei progetti basati su determinati permission scheme template.



ATLASSIAN

La creazione del progetto NON cambia rispetto allo standard, ma si innesta perfettamente nel flusso di lavoro di JIRA.



ATLASSIAN



Completiamo il nostro giro con [Issue Card Printer for JIRA](#), che permette di poter stampare delle card dalla nostra AGILE BOARD

NATLASSIAN

Conclusioni

Abbiamo fatto un piccolo tour su degli addon, per installazioni server, che aggiungono delle nuove funzionalità per JIRA. Si tratta di funzionalità che aiutano tantissimo nell'uso di tutti i giorni. Ve li consiglio ☐