Mind Maps per Confluence – Diamo una prima occhiata

Confluence – Usiamo le Mind Maps

In questo post andremo ad esaminare un addon che permette di poter generare delle Mind Maps. Vediamo come questo addon ci può aiutare attraverso Confluence.



Una piccola definizione ci aiuta

*Una **mappa mentale** è una forma di rappresentazione grafica del pensiero teorizzata*
dal cognitivista inglese Tony Buzan, a partire da alcune riflessioni sulle tecniche
per prendere appunti. Il fine consiste nell'implementare la

memoria visiva e quindi
la memorizzazione di concetti e informazioni in sede di richiamo.

Le mappe mentali (mind maps) non vanno confuse con altri tipi di mappe come le mappe
concettuali dalle quali si differenziano sia per la strutturazione, sia per il modello
realizzativo, sia per gli ambiti di utilizzo.

(fonte wikipedia)



Che cosa offre l’addon?

L’addon mette a disposizione una serie di strumenti che permettono di creare delle mappe mentali nelle nostre pagine Confluence

permettendo di poter scegliere il tipo di mappa mentale. Il risultato che abbiamo nelle nostre pagine è il seguente:



Questo ci aiuta tantissimo nel predisporre il nostro ragionamento o nel trasmetterlo ai nostri colleghi.

Che altro?

Il produttore dell'addon ha messo a disposizione un video, che riporto

Conclusione

Abbiamo iniziato ad esaminare un addon molto interessante, che ci permette di poter fissare, nelle nostre pagine Confluence, le nostre idee e di poterle trasmettere attraverso le mappe mentali. Nei prossimi post andremo ad eseguire delle prove sul campo. L'addon è al momento disponibile solo per le installazioni Cloud ma, come già indicato in altri post, sicuramente il produttore ci sorprenderà con la versione server :-).

Reference

Maggiori informazioni sono reperibili 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.



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 :-O

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 si

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

mooooooooooolto interessante . Cercheremo di svilupparlo molto bene nei prossimi post.

Nuove features in Confluence

Nuove features in Confluence

In questo post andremo ad esaminare una piccola novità di Confluence, che aiuta molto gli utilizzatori nello svolgere il proprio lavoro.

Resizable table columns in Confluence

Una delle ultime caratteristiche introdotte, già presenti sulla versione Cloud, è la possibilità di poter inserire delle tabelle ridimensionabili, come mostrato dalla seguente GIF animata:

Abbiamo la possibilità di poter disporre di tabelle a dimensione fissa o responsive :-). Usando le tabelle tutti i giorni, come utente, posso assicurarvi che è una feature che aspettavo da un pò di tempo.

Avendo a disposizione una istanza cloud, che uso regolarmente, l'uso di tabelle responsive è UTILE :-D.

Conclusioni

Una piccola novità, un ottimo aiuto per noi utenti di Confluence. Restiamo in attesa di ulteriori … sorprese.

Reference

Maggiori informazioni sono reperibili qui.

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.

Webinar su Atlassian

Webinar su Atlassian

Segnalo il seguente webinar su Atlassian JIRA Service Desk da parte di ibuildings. Maggiori informazioni sono presenti qui.

Component & Subcomponents –

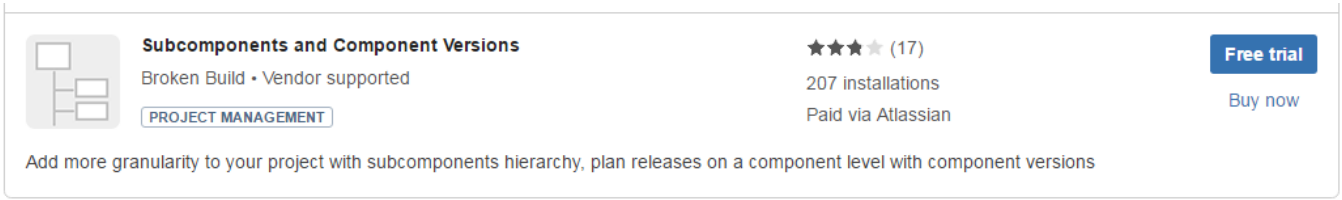
Test su strada

Test su strada

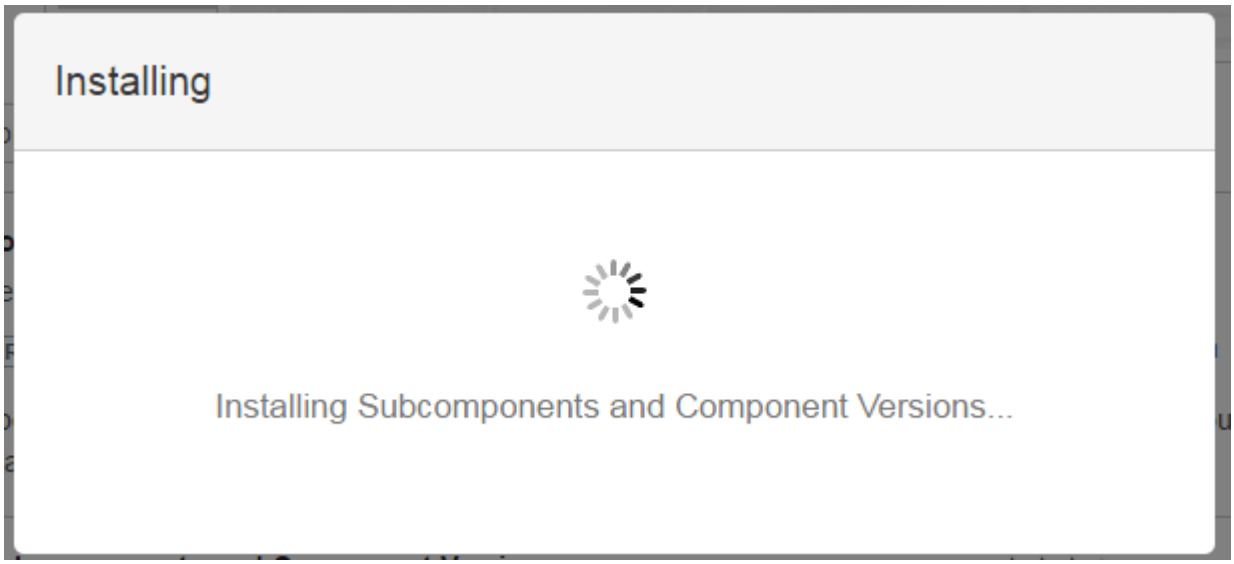
In questo post andremo a fare il test di questo addon, cercando di trovare ciò che ci serve.

Installazione

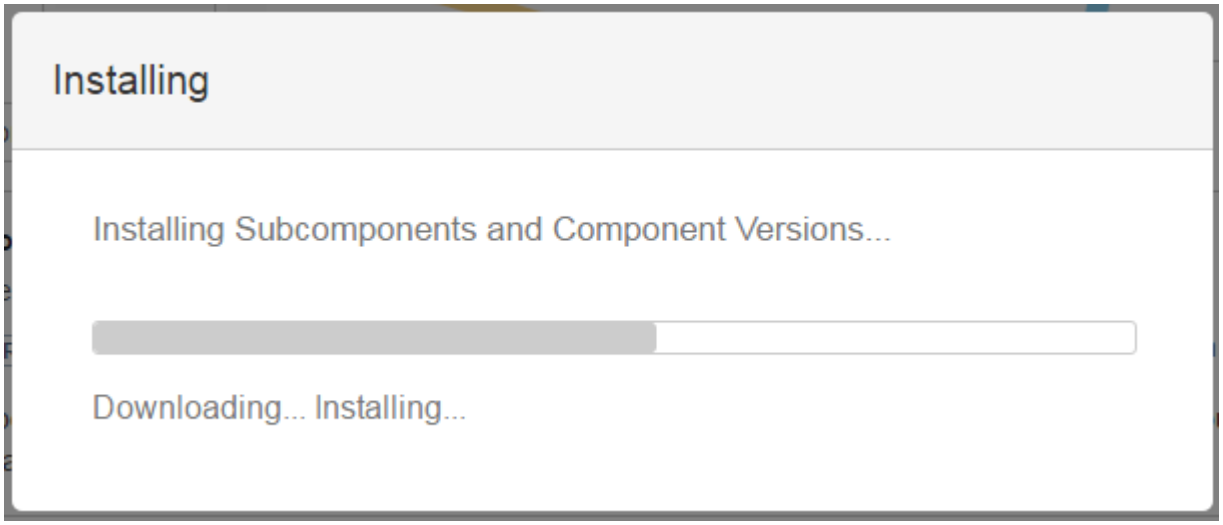
Partiamo sempre dalla installazione del nostro addon, che come sempre non ci deve spaventare :-). In questo caso andremo ad utilizzare una installazione di ***JIRA CORE.*** Selezioniamo il nostro addon e procediamo, selezionando ***Free trial*** dalla nostra schermata di ***Manage Addons***.



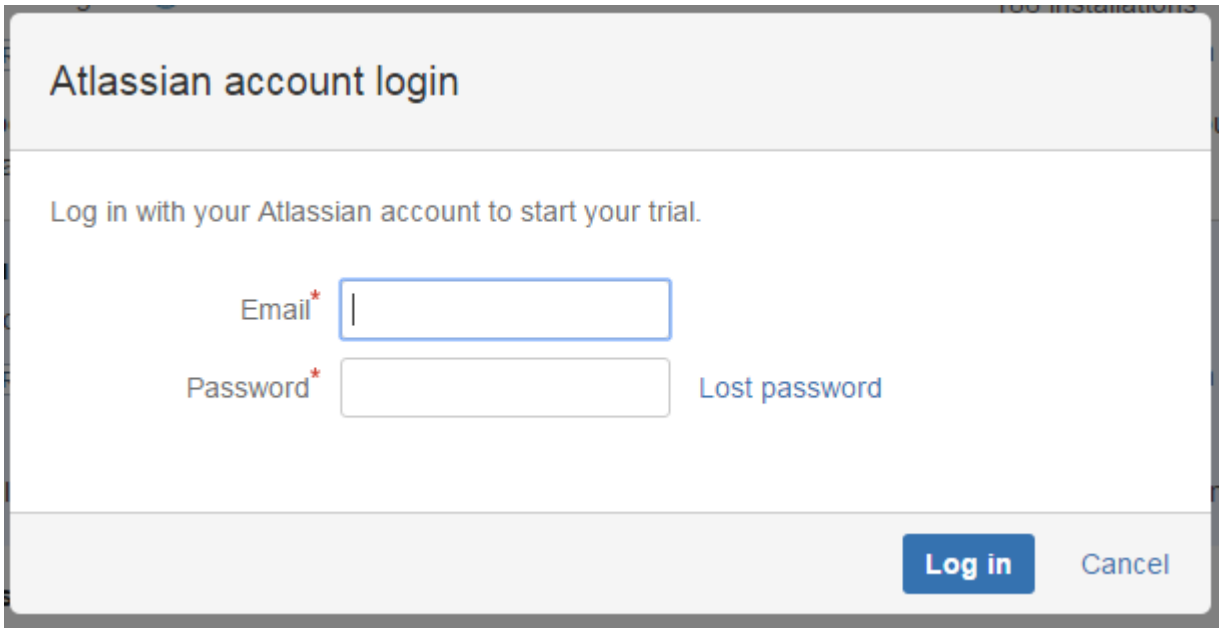
Lasciamo andare la nostra installazione….



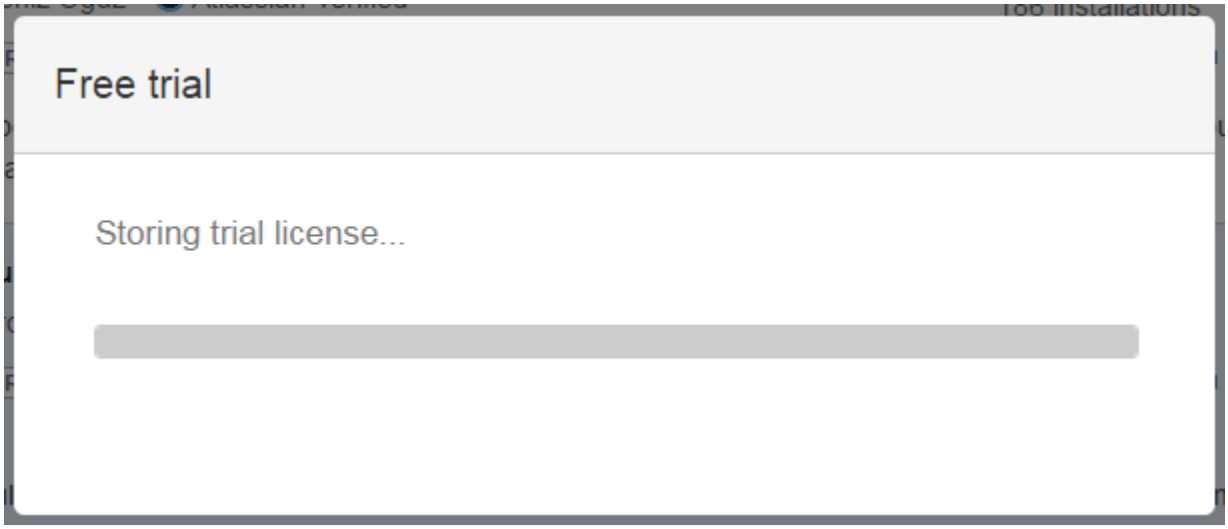
…. facciamo eseguire il download e la relativa installazione …



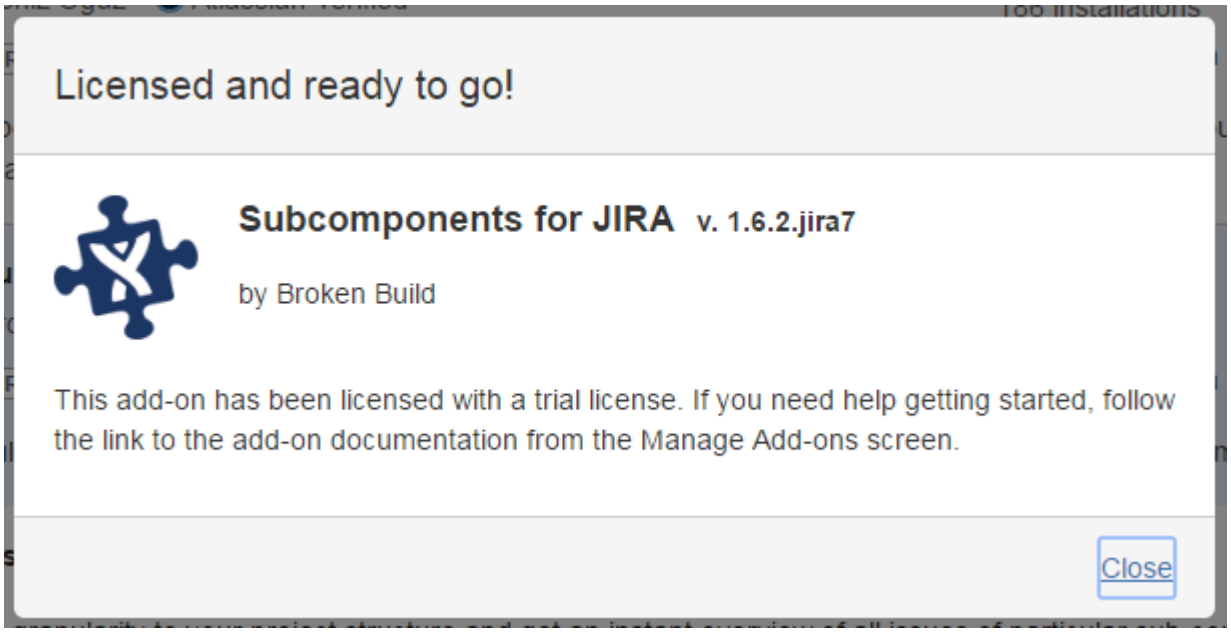
… inseriamo le credenziali per accedere al ***my.atlassian.com*** …



… lasciamo che installi la licenza ….

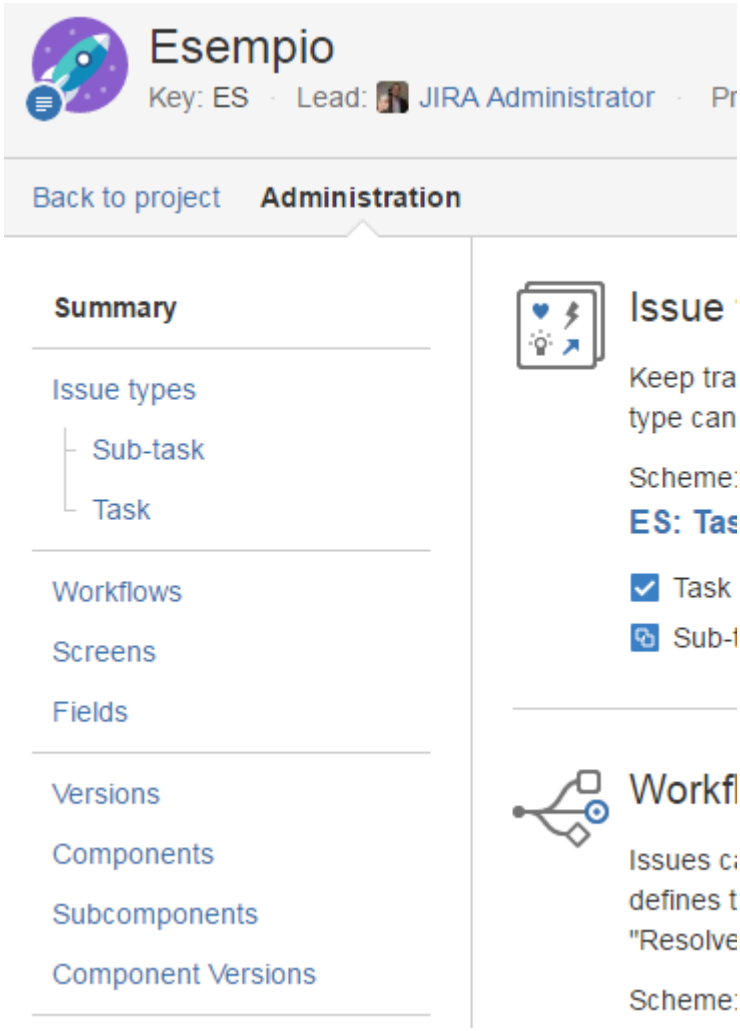


…. ed il nostro addon è pronto all’uso.

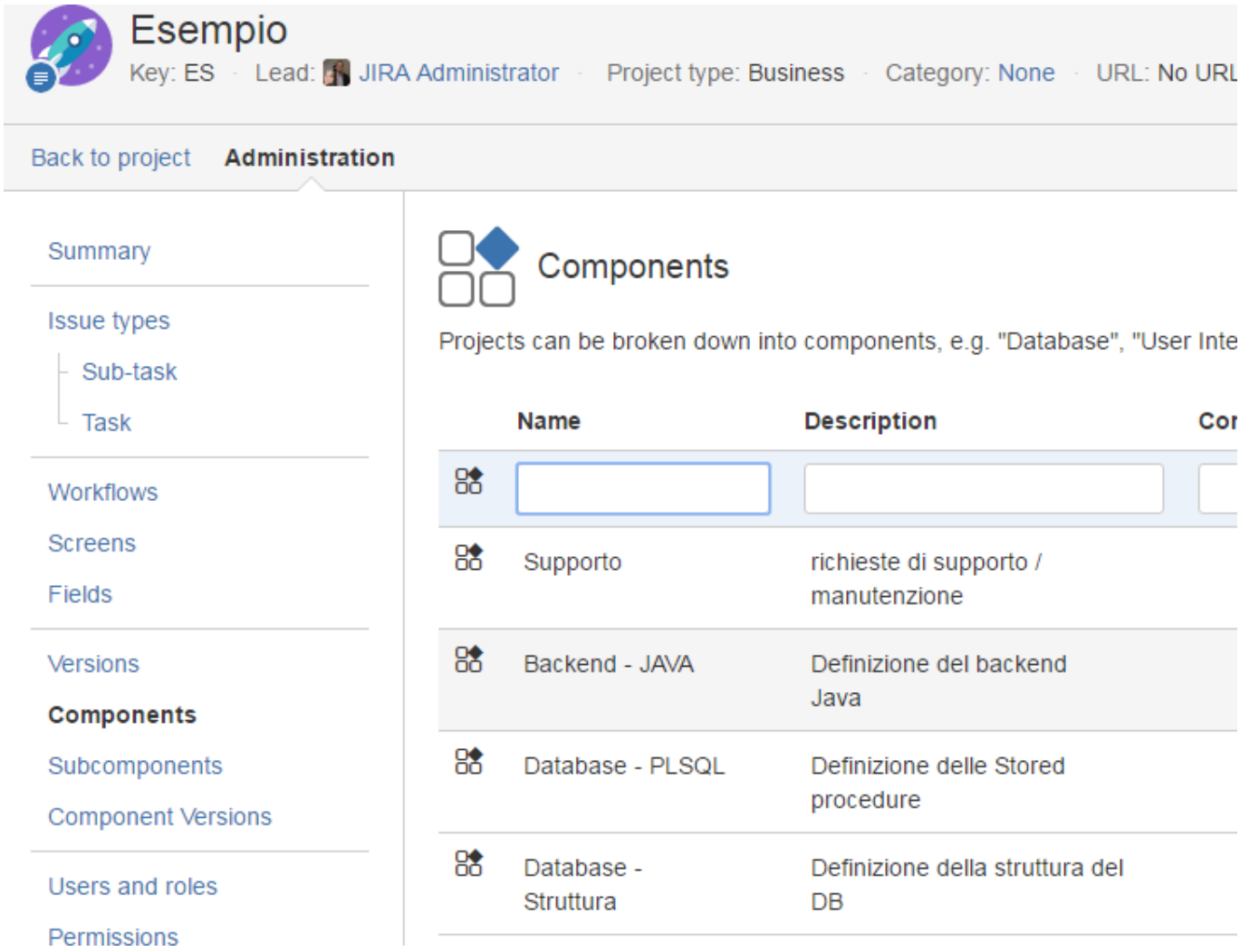


Configurazione

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.

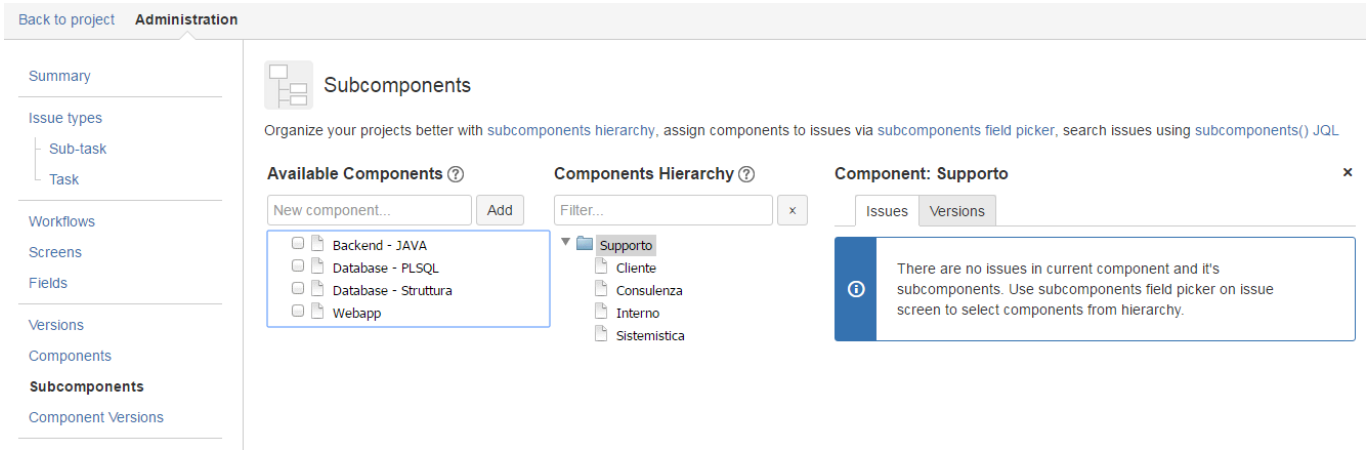


Tentiamo una prima configurazione sul nostro progetto di ***Esempio***. Supponiamo di avere delle componenti che identificano il nostro sistema da gestire.



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.

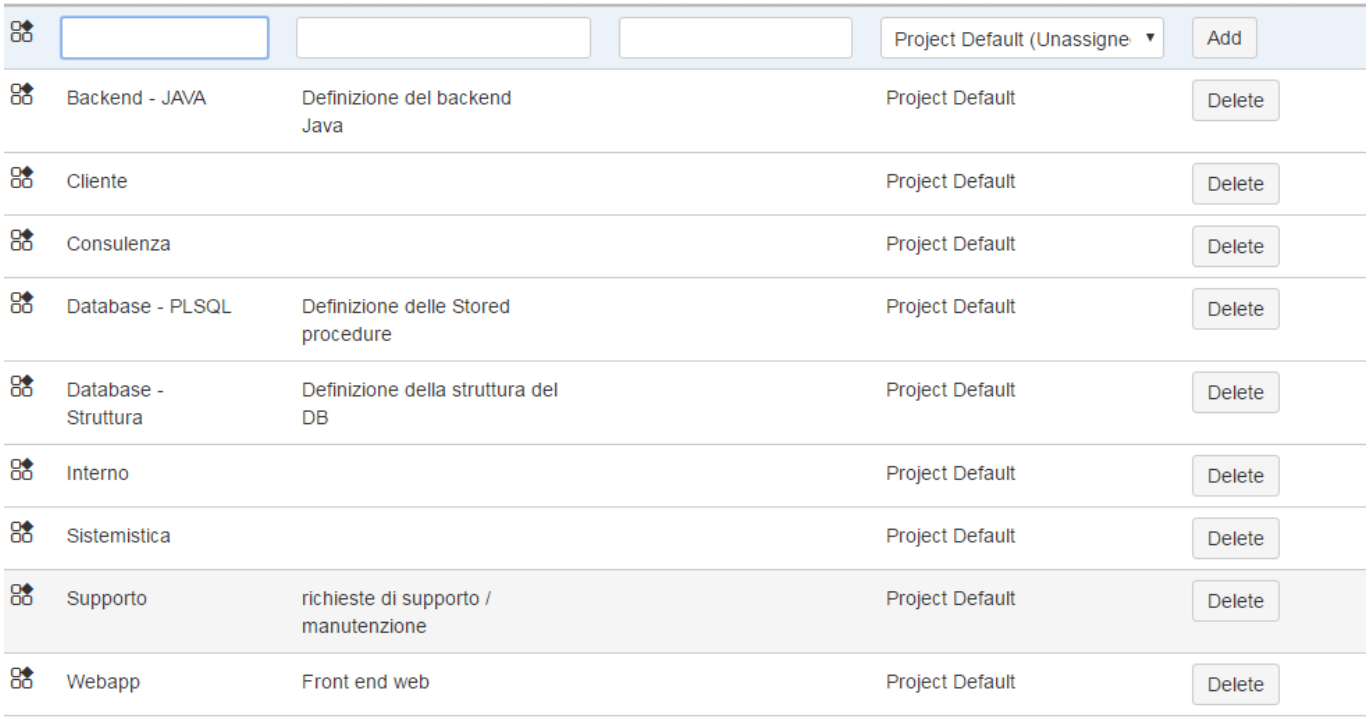
Creiamo le nostre sottocategorie selezionando l'opzione ***Subcomponent***. Viene mostrata la form per compilare la nostra gerarchia.



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

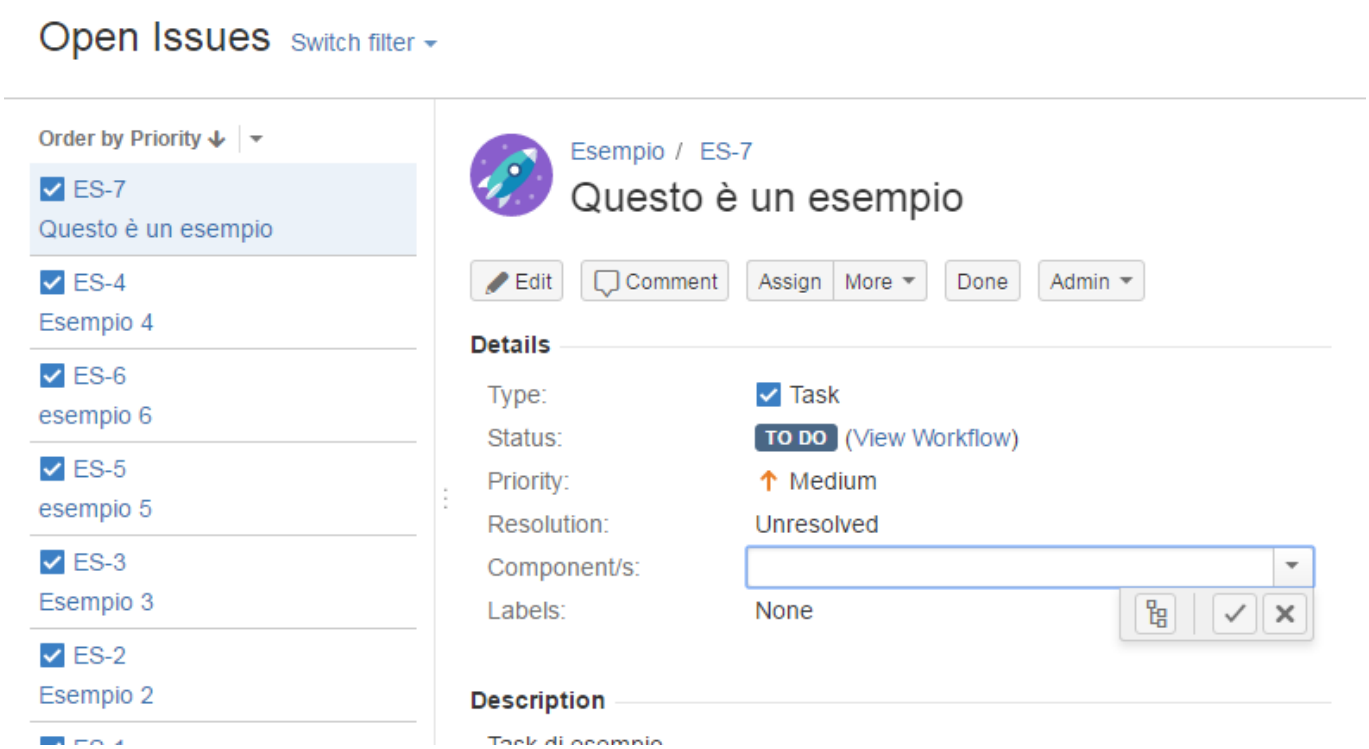
Ma le componenti?

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.

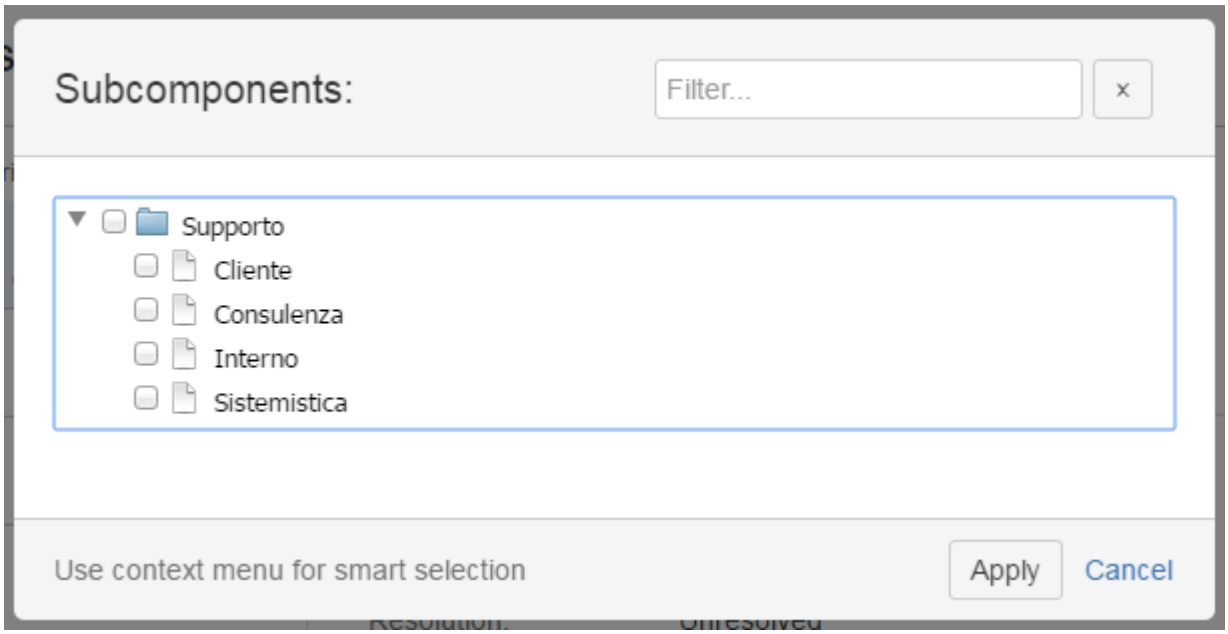


Come usiamo questa gerarchia?

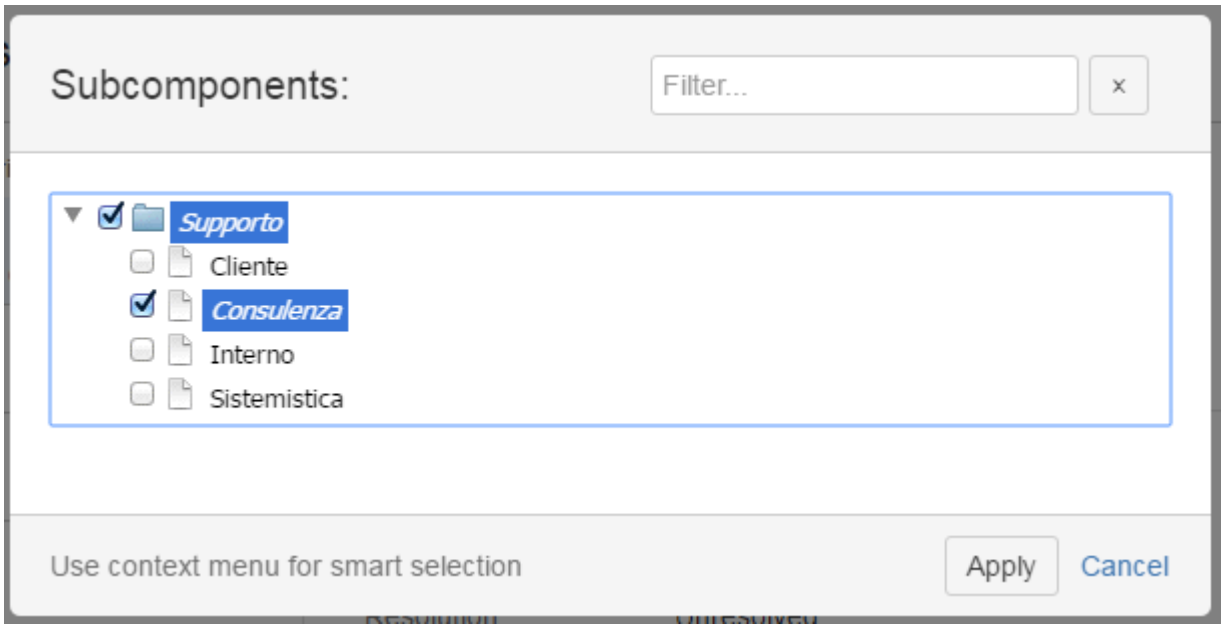
Adesso vediamo come usarla. Creiamo una nuova Issue e andiamo ad inserire la nostra gerarchia di componenti



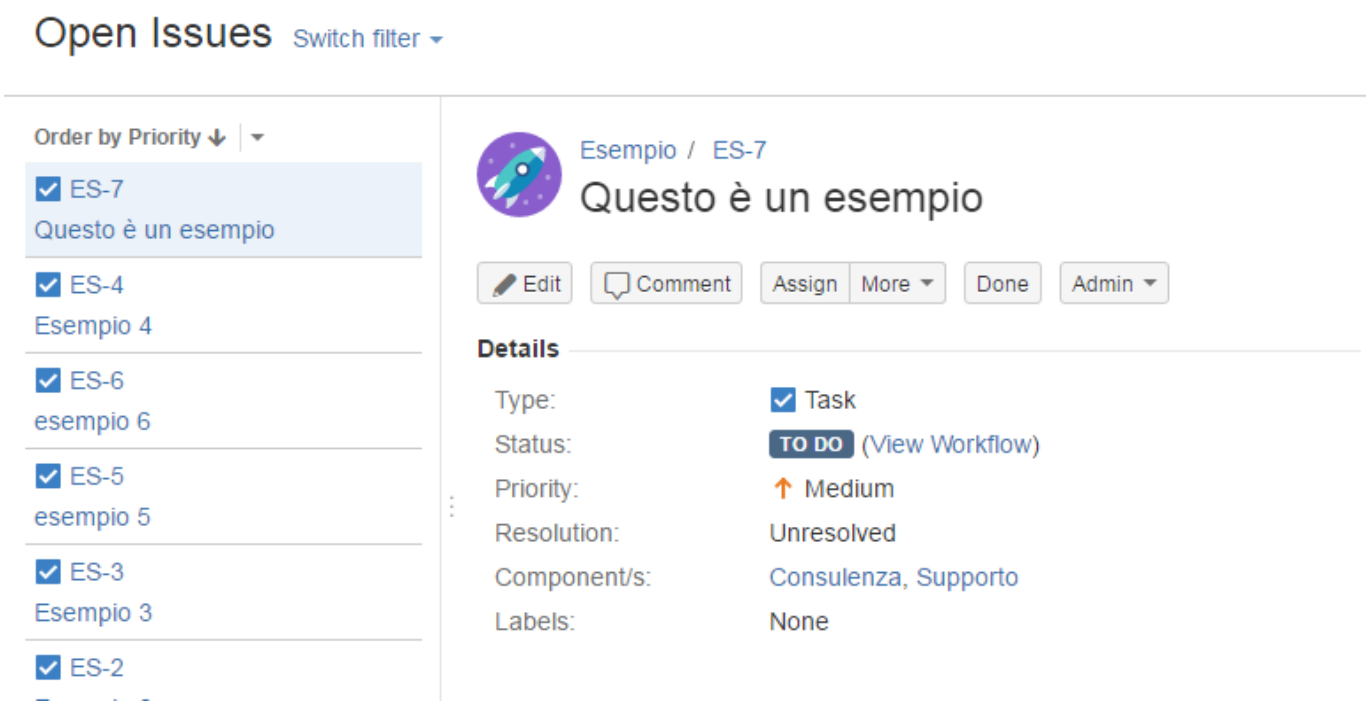
Se andiamo ad editare il campo ***Component/s*** notiamo, come mostrato dalla figura precedente, una nuova icona che indica la gerarchia.



Se lo selezioniamo possiamo andare a selezionare le nostre componenti.



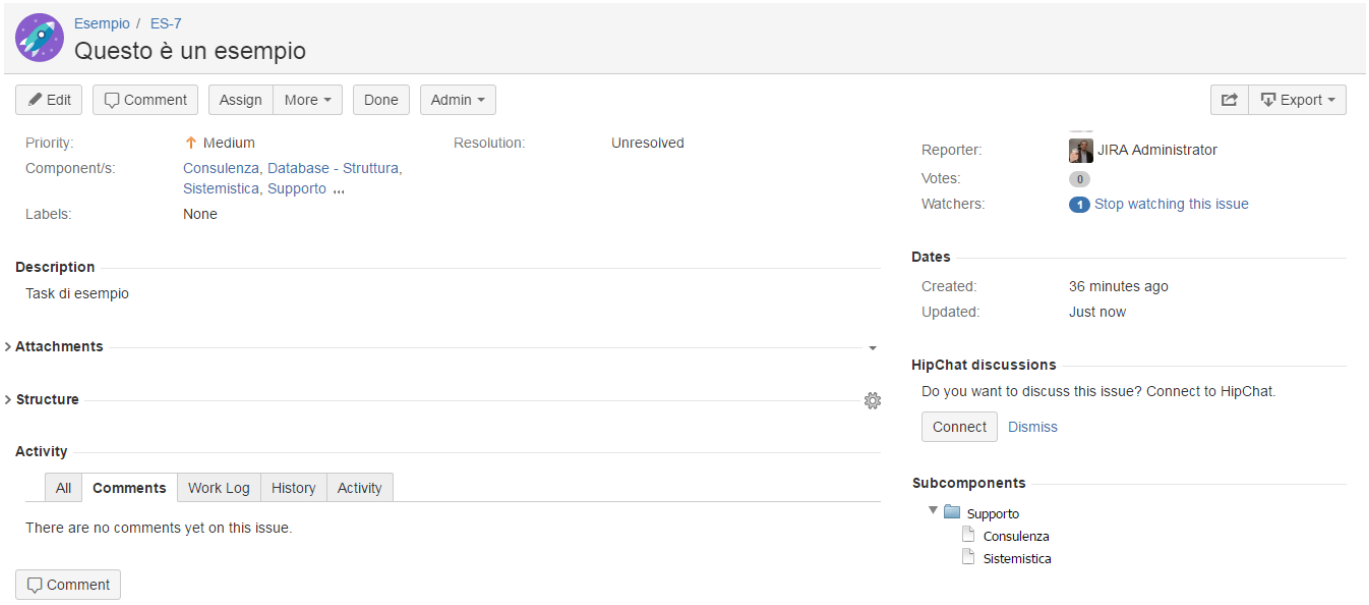
Se andiamo a confermare, il risultato è il seguente:



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.



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

Spreadsheet su Confluence – Un aggiornamento

Spreadsheet su Confluence – Un Aggiornamento

In questo post andremo a verificare le ultime novità su un addon che abbiamo già avuto modo di analizzare: Spreadsheets for Confluence. Ne abbiamo parlato in diversi Post. Qui approfondiremo le ultime novità.

ATLASSIAN
logo addon

Andiamo subito in dettaglio

Come prima cosa segnaliamo che è stata migliorata la fase di interazione e visualizzazione dei fogli elettronici. In precedenza questa era la visualizzazione:

Precedente visualizzazione

Con le ultime versioni, questo è quello che vedremo:

Nuova visualizzazione

ovvero, i fogli elettronici sono visualizzati come tabelle di Confluence. Possiamo vedere che abbiamo una migliore integrazione con le nostre pagine. Questa è la visualizzazione standard. Esiste anche la possibilità di poter visualizzare la sola icona:



logo addon

all'interno della pagina.

Altra novità introdotta è la possibilità di poter stampare i fogli elettronici, ovvero con un semplice CTRL+P è possibile mostrare l'anteprima di stampa.

Conclusioni

Abbiamo delle piccole ma efficaci novità. Prossimamente eseguiremo una prova su strada per valutarne i risultati

Reference

Maggiori informazioni sull'addon sono presenti sul blog degli autori e sul marketplace.

GIT Vs Mercurial

GIT Vs Mercurial

In questo post continuiamo la nostra esplorazione di BitBucket, andando ad analizzare GIT e Mercurial, ovvero i sistemi di controllo versione del codice che Bitbucket utilizza.



Esaminiamo i nostri candidati

Come ogni confronto, andiamo ad analizzare i nostri candidati, dove cerchiamo di dare le caratteristiche principali di entrambi e cercheremo di evidenziare i punti cardine, che come ben sappiamo sono quelli che ci aiutano a scegliere uno o l'altro.

Precisazione….

*L'obbiettivo di questo post non è giudicare quale sia il migliore, tra **GIT** e **Mercurial**, ma solo di descriverli, in modo da poi poter ….. scegliere quale dei due utilizzare*

nel nostro Bitbucket :-).

Mercurial



> *Mercurial is a free, distributed source control management tool. It efficiently handles projects of any size and offers an easy and intuitive interface.*

***Mercurial** è un sistema di controllo versione distribuito, gratuito e molto semplice da usare. Ho rintracciato una guida molto carina che potete reperire da questo [link](#).*

*Fondamentalmente, questo sistema, ragiona per **repository**, ovvero per cartelle dove sono contenuti tutti i file inerenti l'applicativo che stiamo gestendo. La differenza rispetto ad un altro sistema, come SVN, quando eseguiamo una operazione di **clone**, letteralmente cloniamo il repository, con tutta la sua storia, senza doverci ogni volta ricollegare al repository centrale, trasformando il pc locale in un server. Ogni singola operazione, a meno di indicazioni esplicite, sarà eseguita nel repository locale. Concludendo, abbiamo a disposizione una copia locale dove possiamo eseguire le nostre sperimentazioni.*

Git



Git is a [free and open source](#) distributed version control system designed
to handle everything from small to very large projects with

speed and
efficiency.

*Come **Mercurial**, anche **Git** è un sistema di controllo versione distribuito, gratuito e open source, semplice da usare.*

*Come **Mercurial**, anche **Git** trasforma il pc locale nel server di sviluppo. **Git** realizza delle istantanee vere e proprie, totalmente indipendenti dalla copia del server centrale, su cui possiamo eseguire le nostre prove e, una volta terminato, possiamo eseguire la copia nel server centrale che, a sua volta , è un repository **Git**.*

Ho rintracciato una guida in italiano, molto diretta, ed una guida molto completa da cui possiamo iniziare ad esplorare il funzionamento di questo sistema.

Cosa possiamo usare?

Consultando la documentazione, emerge quanto segue:

*Possiamo ancora utilizzare **Mercurial** per la versione Cluod di Bitbucket, per cui viene garantita la compatibilità;*

*La versione server fa invece uso di **Git**, come riportato qui.*

*Di conseguenza, la scelta è abbastanza scontata e procederemo con l'installazione della versione server con **Git**.*

Conclusioni

Nel prossimo post descriveremo l'installazione vera e

propria di tutto ciò che serve, più SourceTree, che andremo ad approfondire.

Reference

Vi consiglio le seguenti letture:

http://blogs.atlassian.com/2012/03/git-vs-mercurial-why-git/

http://blogs.atlassian.com/2012/02/mercurial-vs-git-why-mercurial/

http://gpiancastelli.altervista.org/hgbook-it/

Bitbucket – altri addons

Altri addons per Bitbucket

In questo post proseguiamo la nostra esplorazione del mondo Bitbucket: andiamo ad esaminare altri addons.

Iniziamo senza indugio

Partiamo con Bitbucket Server Archive Plugin, che ci consente di poter eseguire il Download su ZIP o TAR file di tutto il nostro sorgente. L'addon è gratuito (al momento in cui viene scritto questo post). Fondamentalmente ci aggiunge un tasto Download, come mostrato in figura:

semplificandoci la vita (sopratutto se siamo abituati con SVN).

Proseguiamo con Bitbucket Webhook to Jenkins. Questo addon open source, permette di poter triggerare Jenkins ogni volta che viene eseguita una commit.



In questo modo possiamo sfruttare le potenzialità di Jenkins con Bitbucket.

Proseguiamo con una piccola sorpresa: ScriptRunner for Bitbucket Server/Stash.

ATLASSIAN

Come per JIRA abbiamo visto diversi post, esaminando ed usandolo. Abbiamo anche qui questo addon che ci consente di automatizzare diverse operazioni.



Continuiamo con Hipchat Plugin for Bitbucket Server.

ATLASSIAN

Possiamo integrare HipChat con Bitbucket in maniera semplice e sfruttarne le potenzialità.

ATLASSIAN

Completiamo questo secondo giro di esplorazione con Bitbucket Command Line Interface (CLI).



Si tratta della estensione CLI aggiunta per Bitbucket.

ATLASSIAN

Anche questo addon ci aiuta nella automatizzazione di compiti.

Conclusioni

Abbiamo concluso questo secondo giro. Proseguiremo nei prossimi post le prove di questi ed altri addons

Bitbucket – Addons specifici

Esaminamo gli addon di Bitbucket

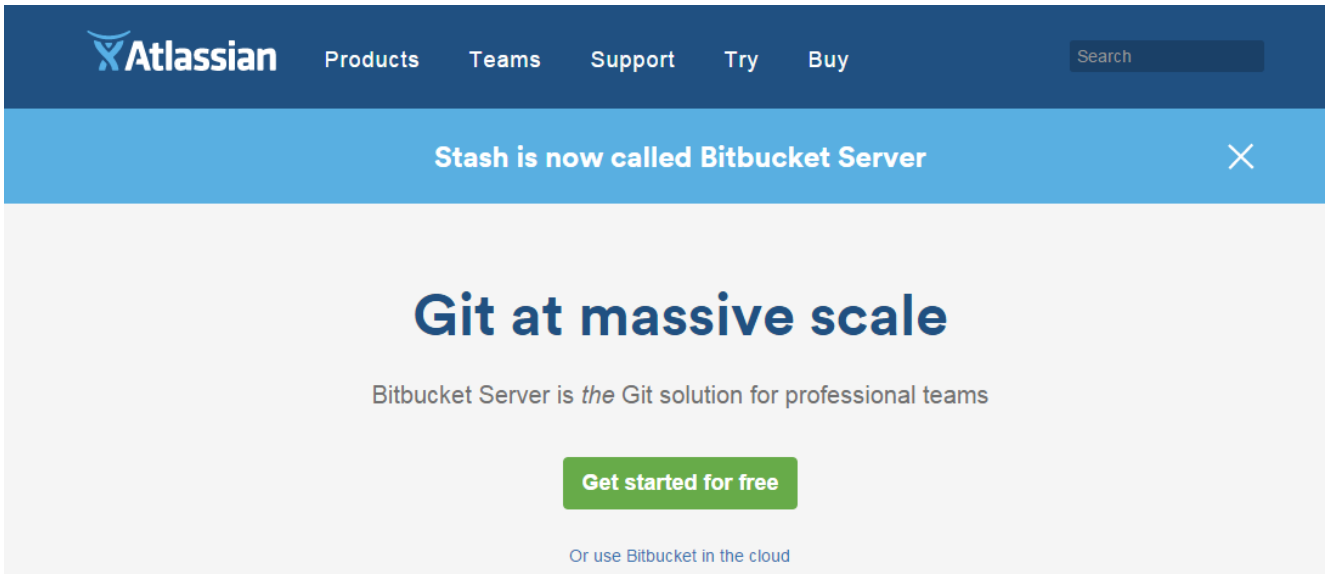
In questo post andremo ad esaminare quali addon esistono per Bitbucket, cercando di fare una prima analisi di quanto è disponibile.



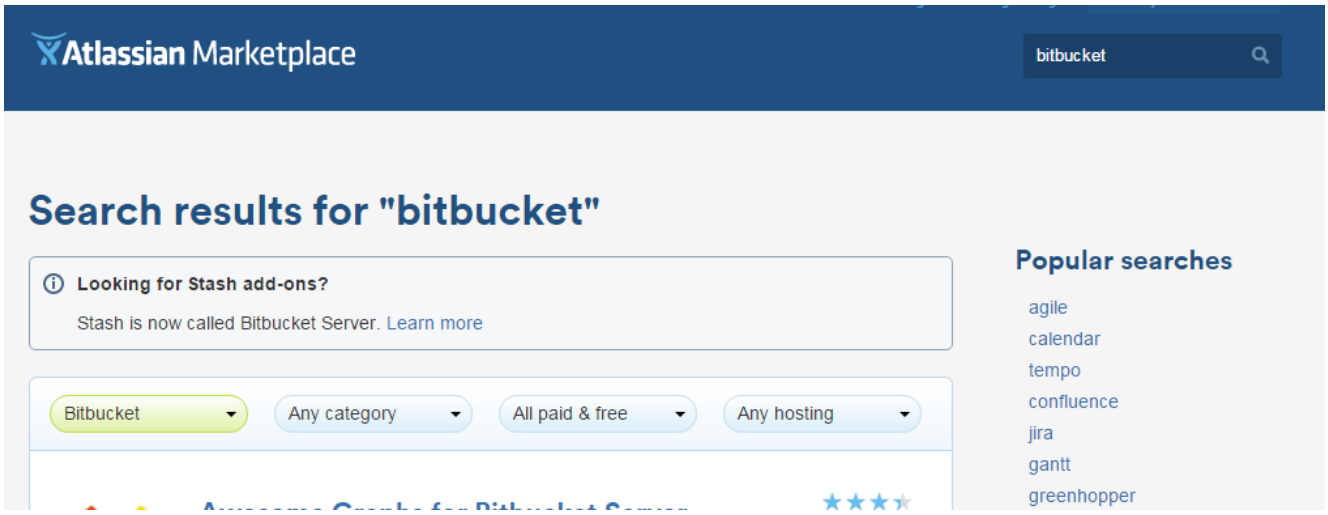
Una piccola precisazione….

In principio era Stash, la prima soluzione che la Atlassian

mette a disposizione come soluzione per gestire dei repositori GIT. Adesso il prodotto è stato chiamato …. Bitbucket.



Se, partendo dalla pagina wikipedia sopra agganciata, andiamo a vedere il link al sito ufficiale della Atlassian, il risultato che si ottiene è la precedente immagine :-). Lo stesso lo otteniamo quando, impostata una ricerca, cerchiamo che cosa offre il Marketplace:



Se volete approfondire le ragioni del rebranding, vi rimando al seguente [link della documentazione ufficiale](#).

Dettagliamo gli addons

Iniziamo la nostra carrellata con Awesome Graphs for Bitbucket Server. Si tratta di uno degli Addons vincitori della codegeist 2012. L'addon permette di poter monitorare e tracciare l'attività degli utenti, attraverso opportune reportistiche e grafici, come mostrato in figura:



Abbiamo a disposizione diverse reportistica:



permettendo anche di tracciare i commit



Maggiori dettagli alla seguente pagina.

Continuiamo la nostra carrellata con SVN Mirror for Bitbucket Server. Si tratta di un addon che consente di poter eseguire dei mirroring tra GIT e SVN, oltre che consentire un import SVN -> GIT.



Abbiamo un pannello di controllo molto semplice



e possiamo anche monitorare anche un singolo elemento.



Terminiamo la nostra carrellata con [Editor for Bitbucket (Stash)](). Si tratta di un addon (gratuito al momento in cui viene redatto l'articolo) che permette di poter modificare, attraverso apposita funzionalità, i sorgenti.



Viene anche data la possibilità di poter eseguire una preview prima di salvare il tutto



e che possiamo committare senza problemi direttamente.



Questo ci consente di poter eseguire delle modifiche minori direttamente su Bitbucket.

Conclusioni

Terminiamo qui questa prima panoramica degli addon. Nei prossimi post continueremo ad esaminare altri addons.