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.

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 altri 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.

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 – Alternative

Quali alternative esistono?

In questo post andremo ad esaminare quali alternative sono disponibili per l'addon descritto qui.Stiamo parlado 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

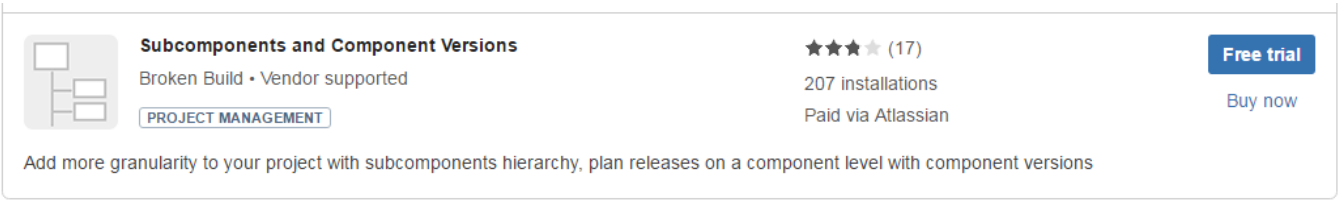
Test su strada

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

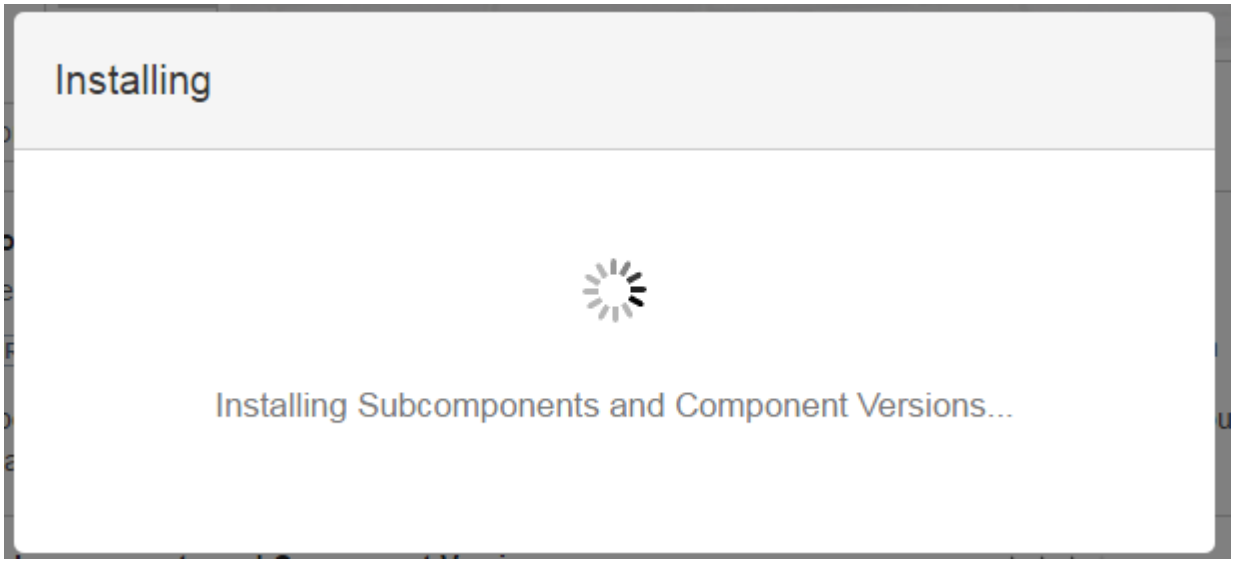


Installazione

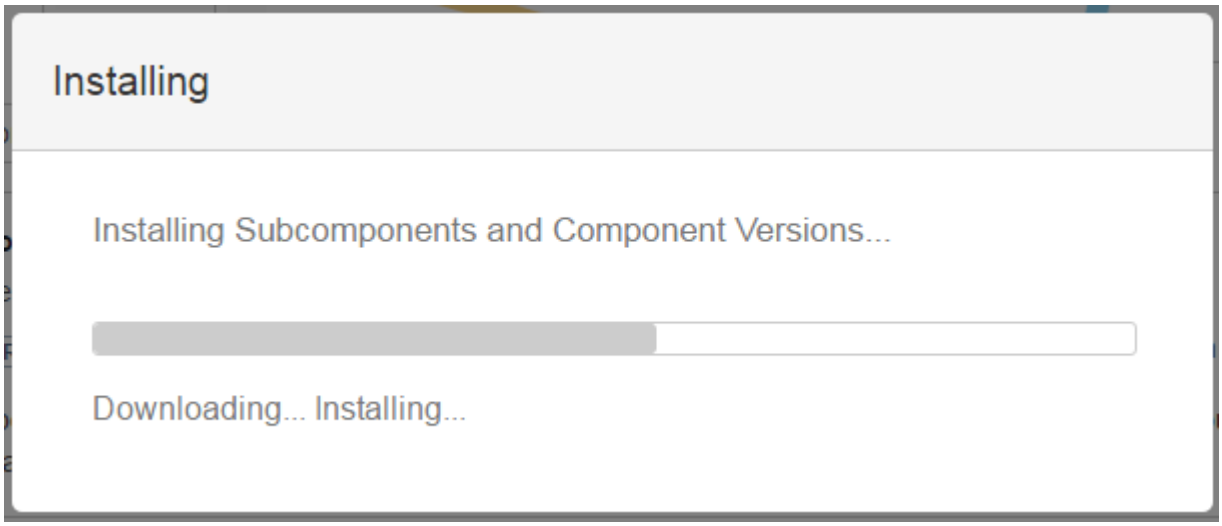
Partiamo sempre dalla installazione del nostro addon, che come sempre non ci deve spaventare :-). In questo caso andremo ad utilizzare una installazione di ***JIRA CORE.*** Selezioniamo il nostro addon e procediamo, selezionando ***Free trial*** dalla nostra schermata di ***Manage Addons***.



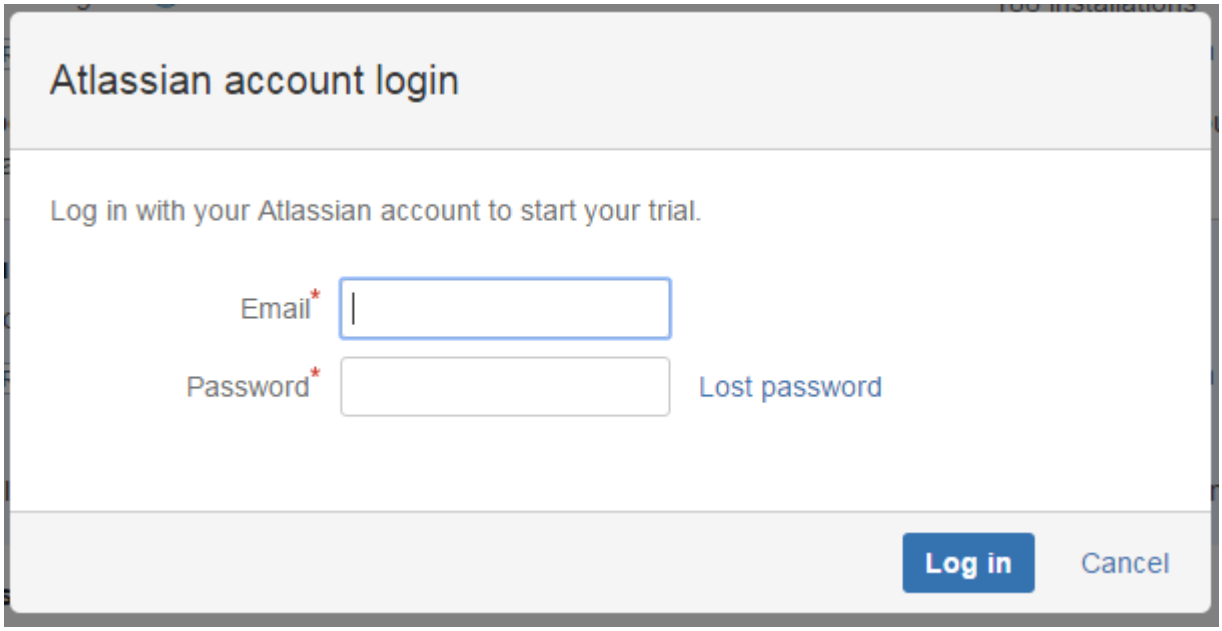
Lasciamo andare la nostra installazione….



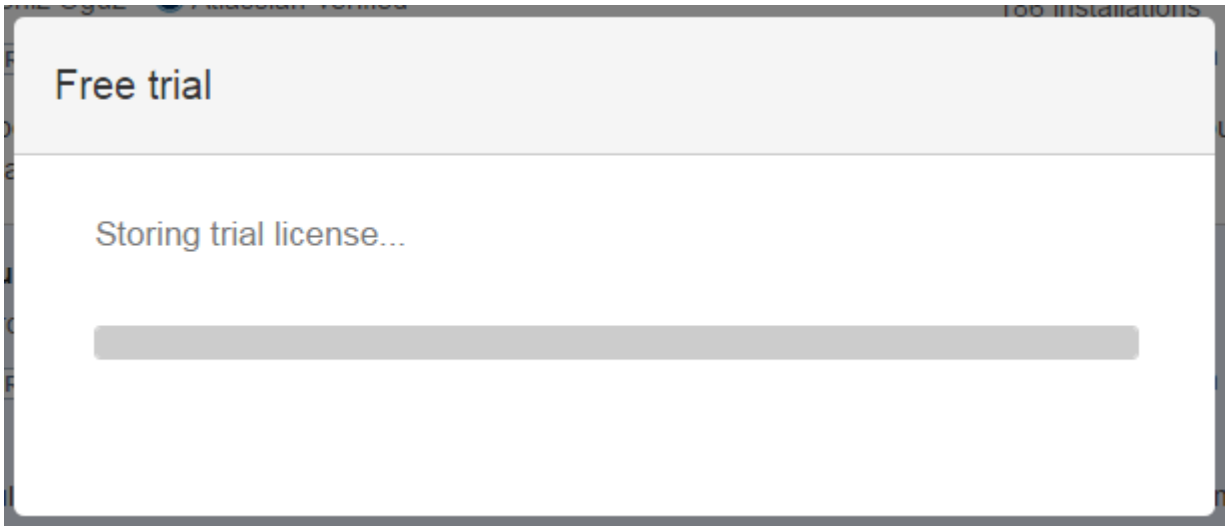
…. facciamo eseguire il download e la relativa installazione …



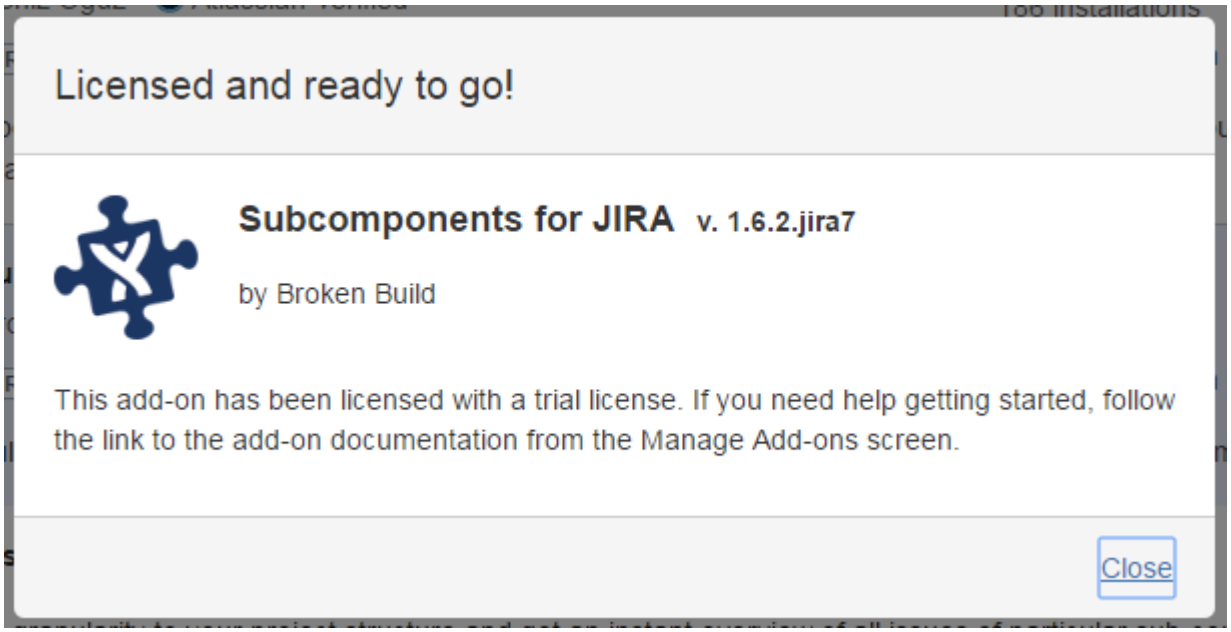
… inseriamo le credenziali per accedere al ***my.atlassian.com*** …



… lasciamo che installi la licenza ….

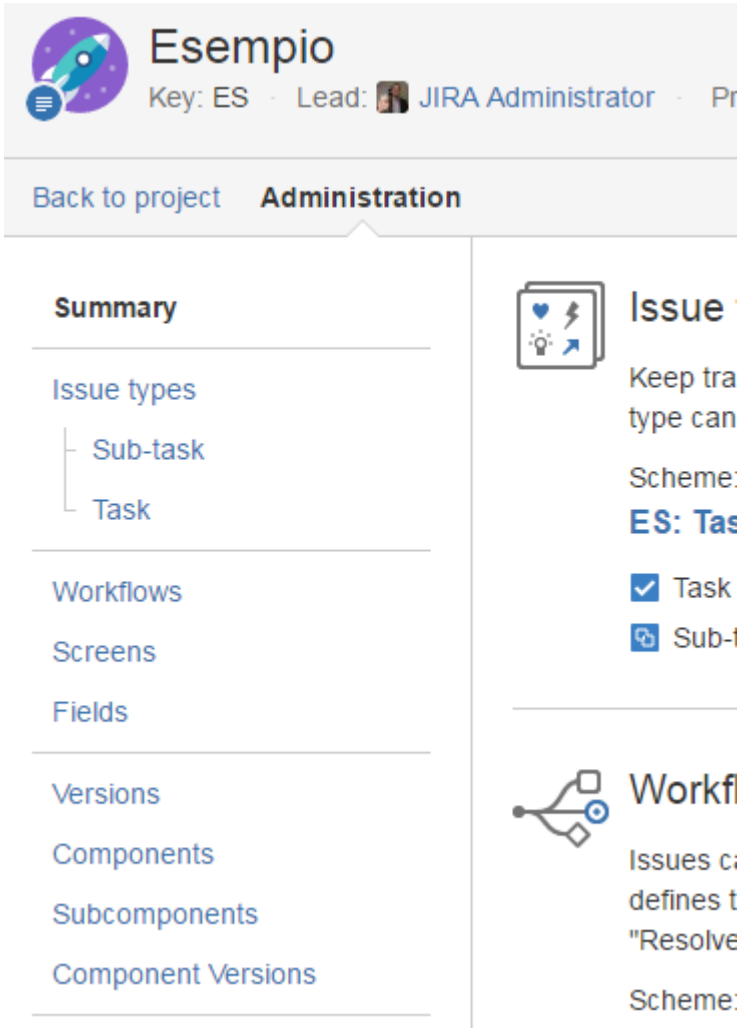


…. ed il nostro addon è pronto all’uso.

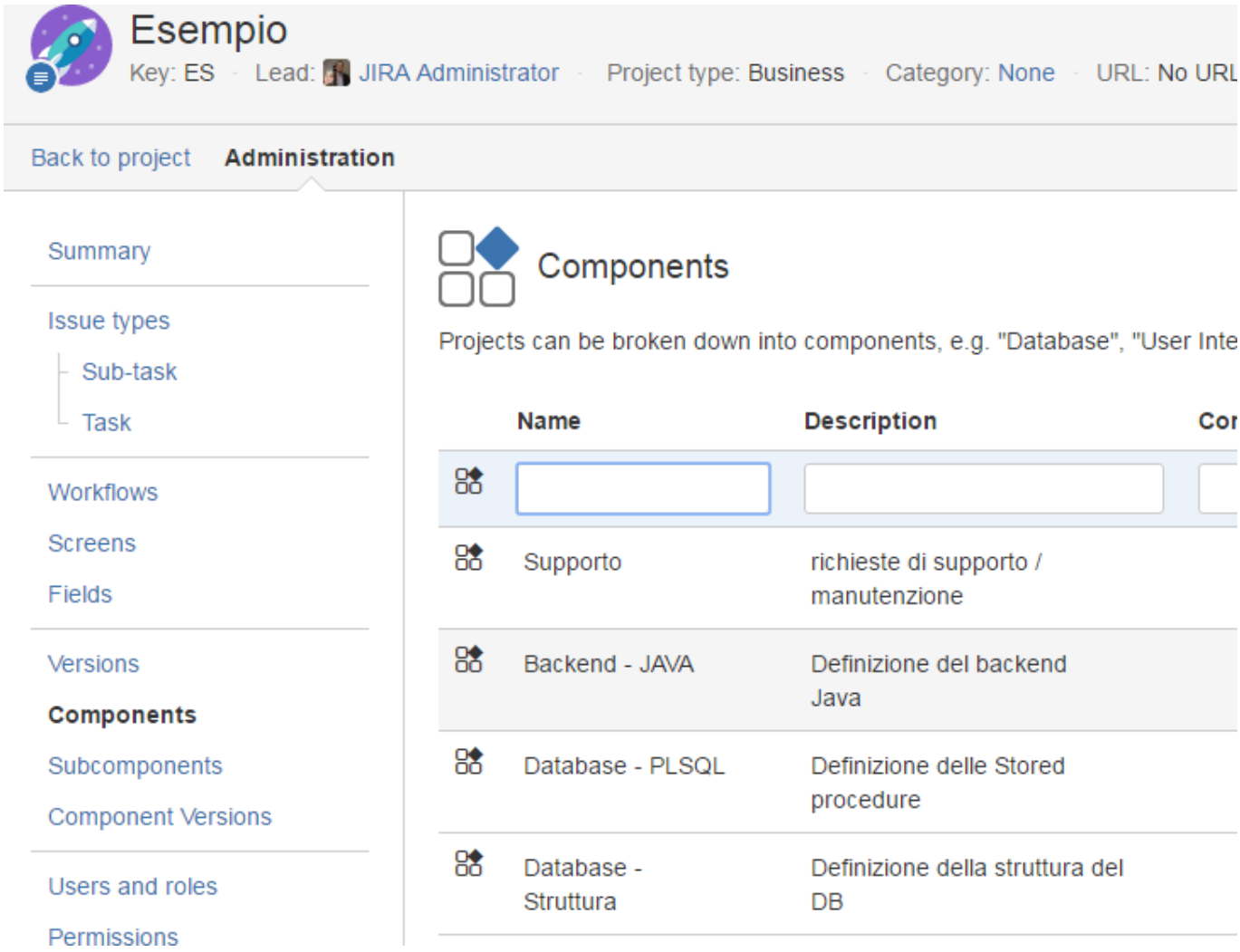


Configurazione

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.

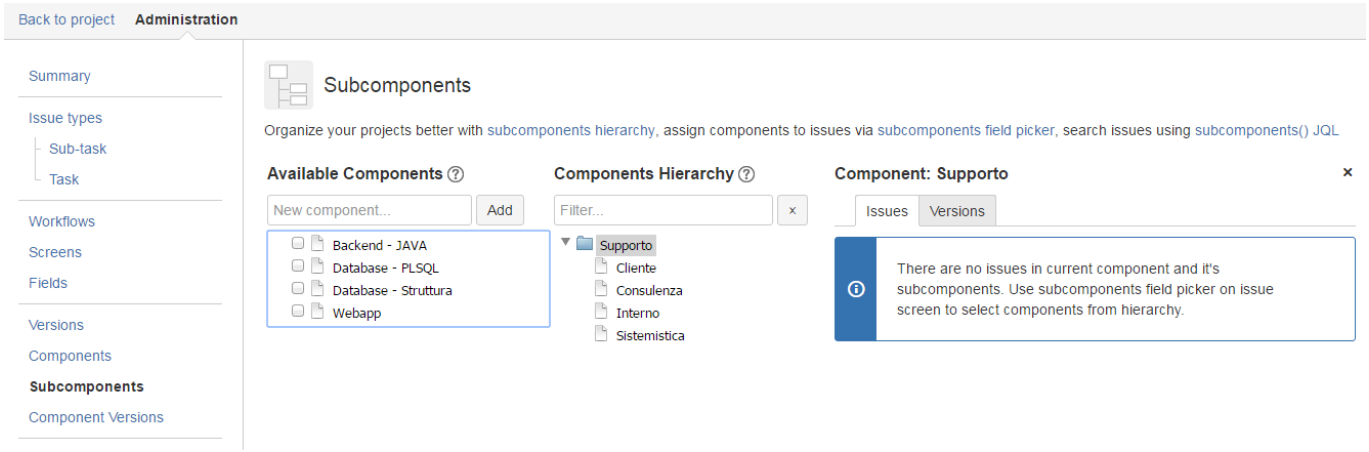


Tentiamo una prima configurazione sul nostro progetto di ***Esempio***. Supponiamo di avere delle componenti che identificano il nostro sistema da gestire.



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.

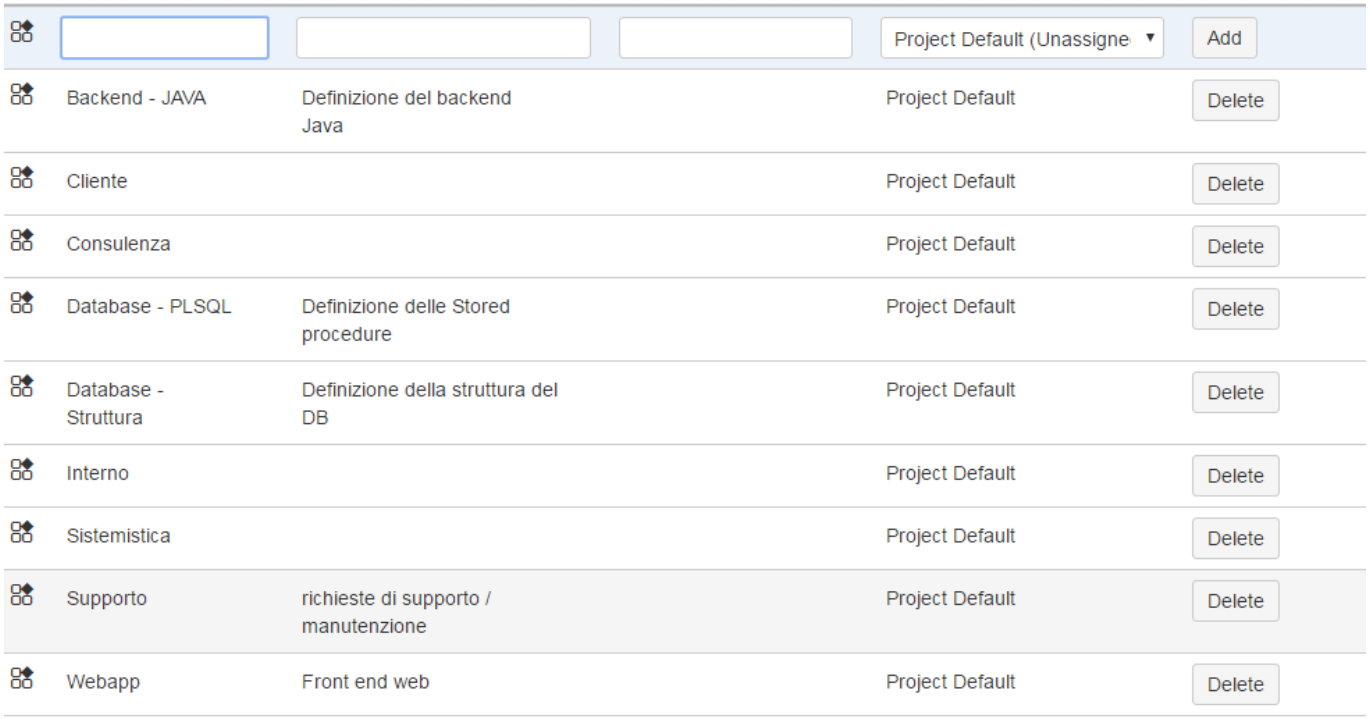
Creiamo le nostre sottocategorie selezionando l'opzione ***Subcomponent***. Viene mostrata la form per compilare la nostra gerarchia.



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

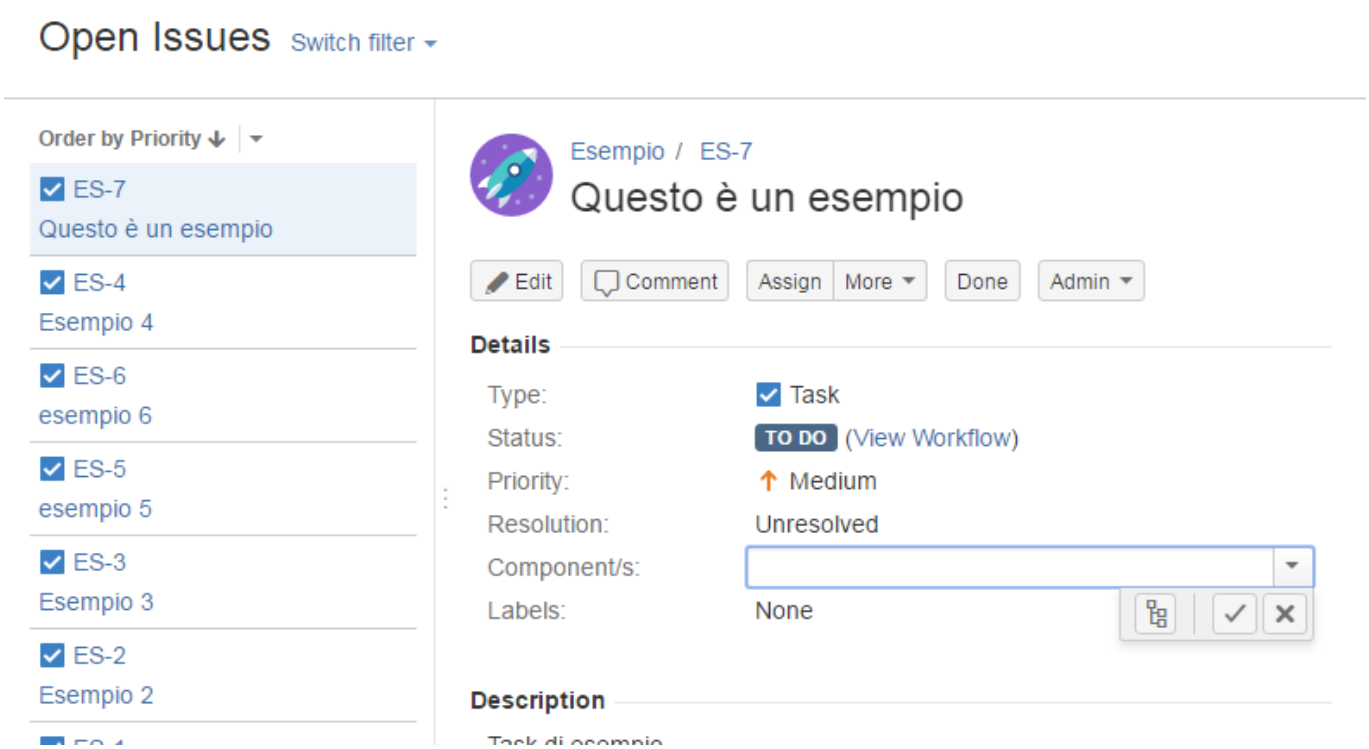
Ma le componenti?

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.

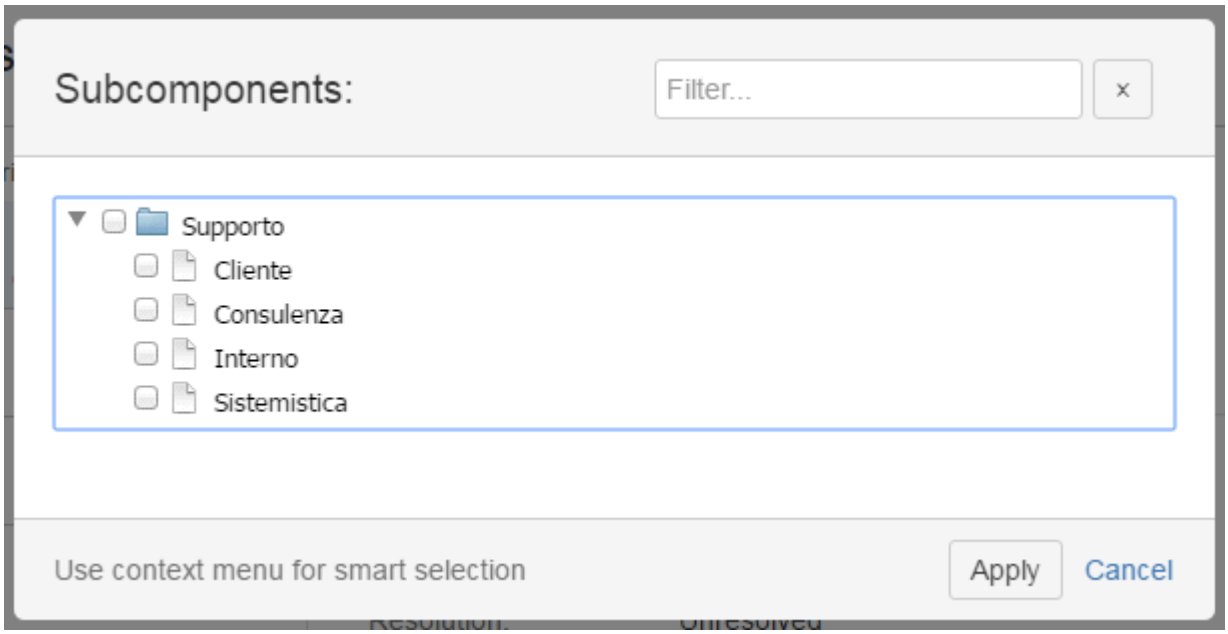


Come usiamo questa gerarchia?

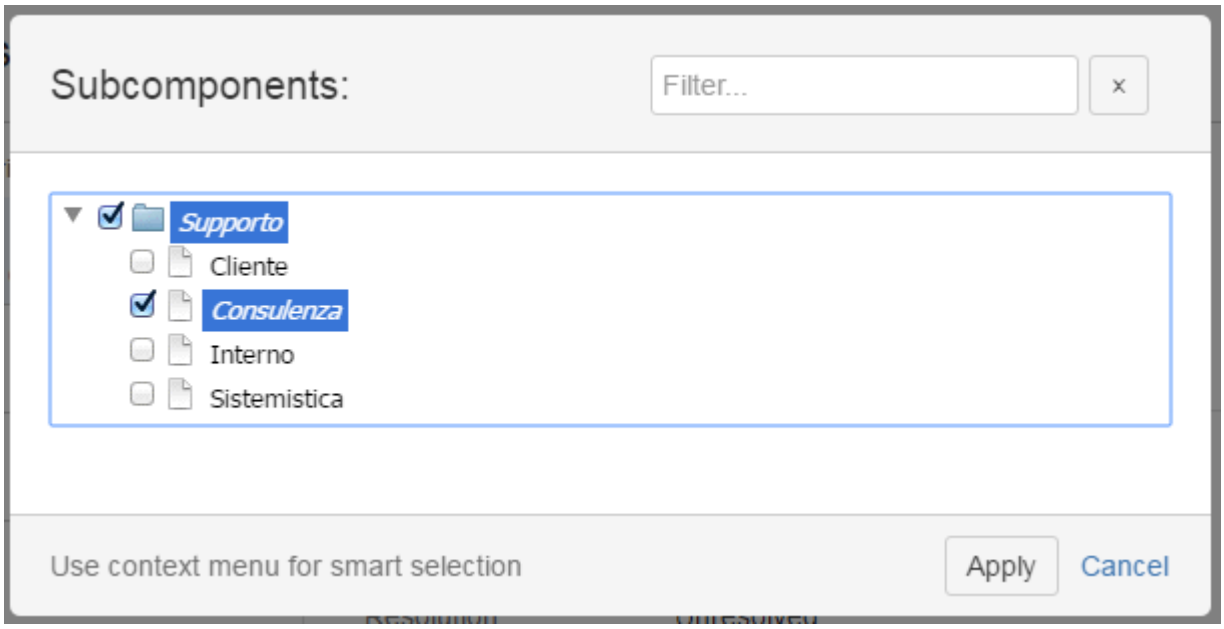
Adesso vediamo come usarla. Creiamo una nuova Issue e andiamo ad inserire la nostra gerarchia di componenti



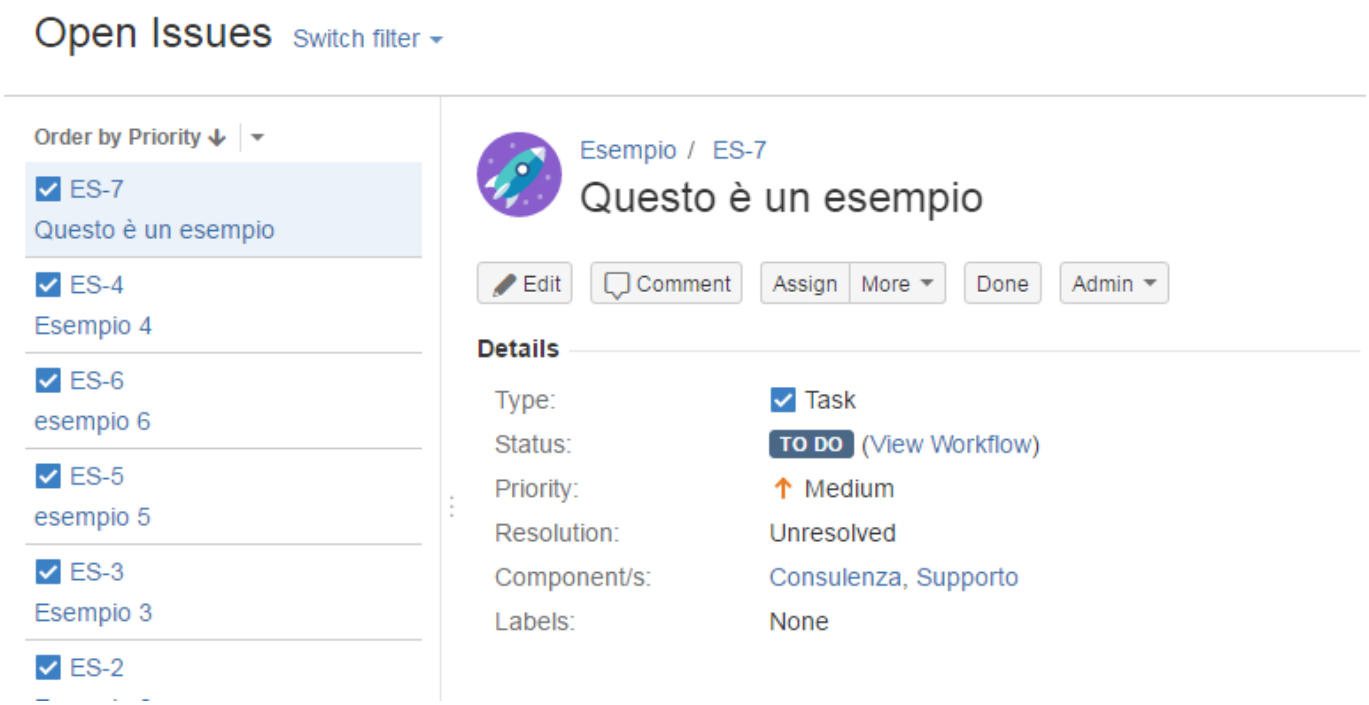
Se andiamo ad editare il campo ***Component/s*** notiamo, come mostrato dalla figura precedente, una nuova icona che indica la gerarchia.



Se lo selezioniamo possiamo andare a selezionare le nostre componenti.



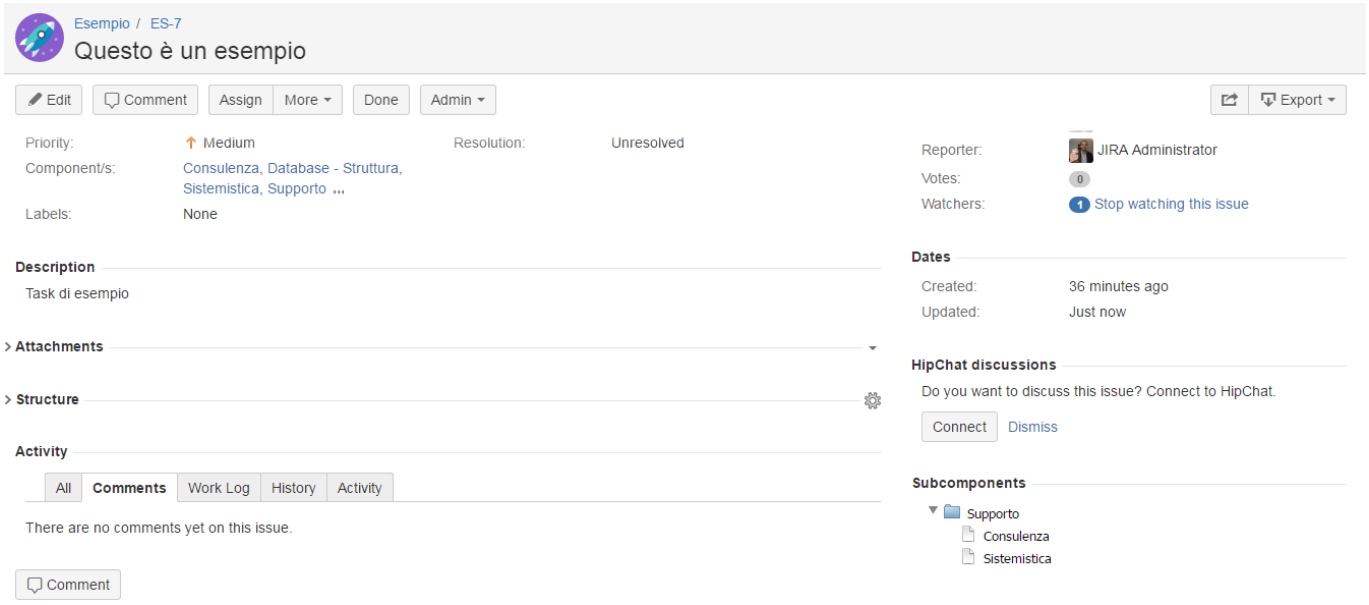
Se andiamo a confermare, il risultato è il seguente:



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

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 (TO DO: 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 (TO DO: 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, sopratutto 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 instanze 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





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 molto ci aiuta.



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.



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





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



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

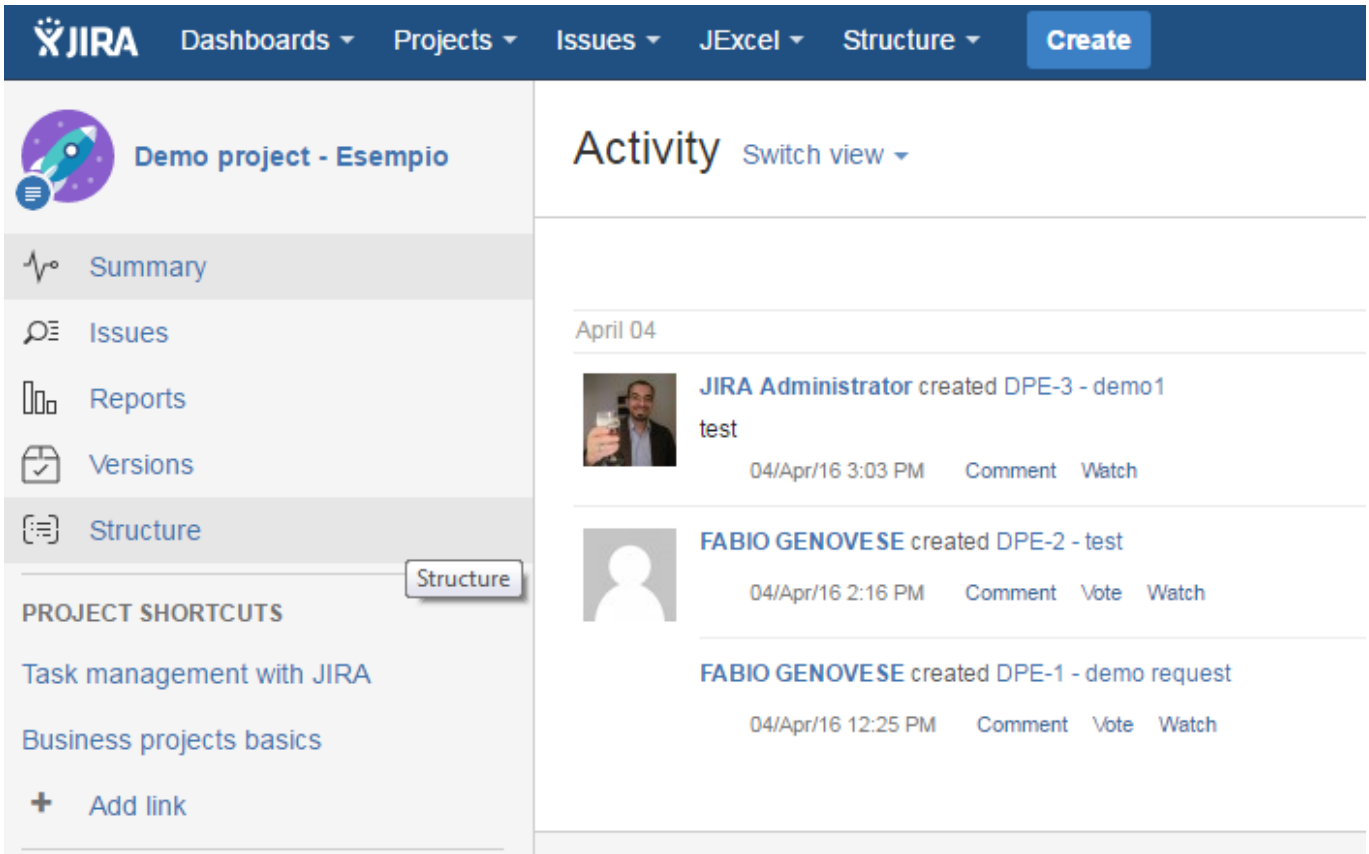
Issue Organizer – Prova su strada 2

Prosegue la prova

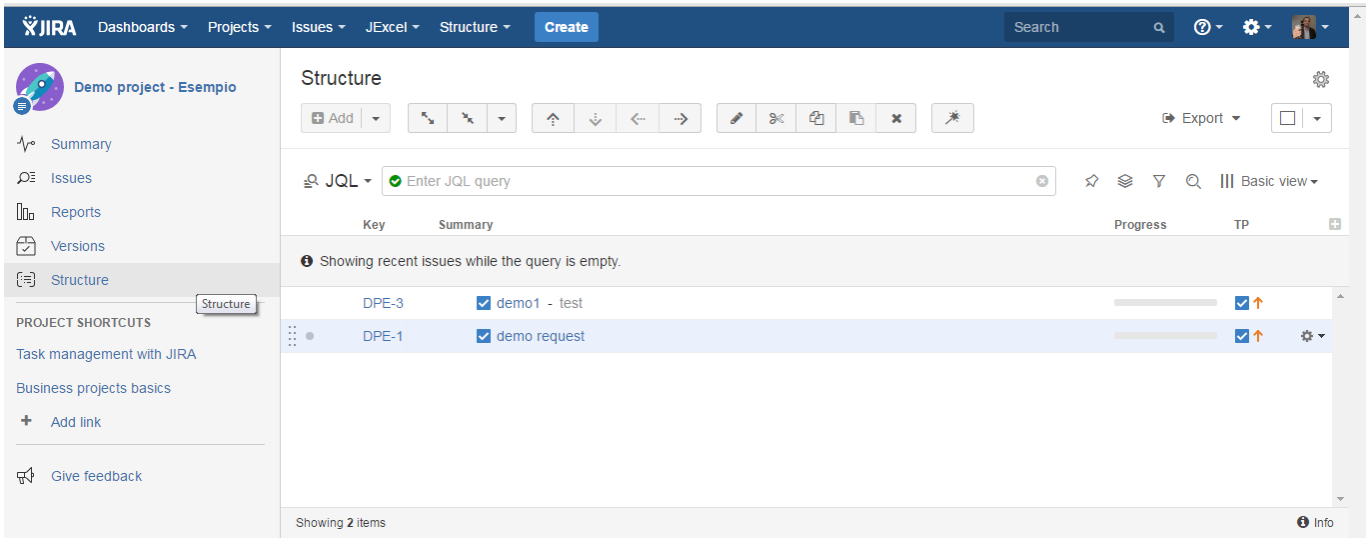
In questo post proseguiamo il test dell'addon Issue Organizer, andando a completare quanto iniziato nel precedente post.

Prova

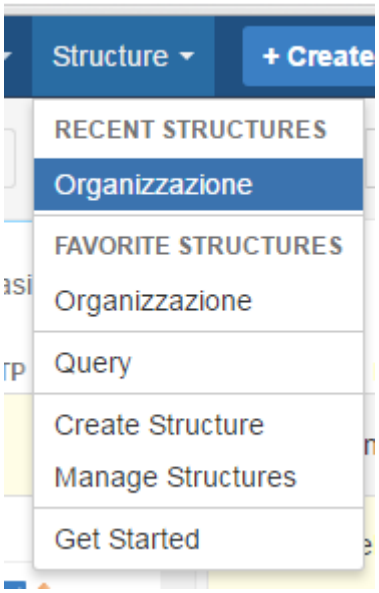
Dopo aver visionato la configurazione dell'addon, proseguiamo con le operazioni di test vero e proprio. La prima cosa che notiamo è che nella toolbar laterale della schermata del progetto si presenta una nuova opzione: ***Structure***.



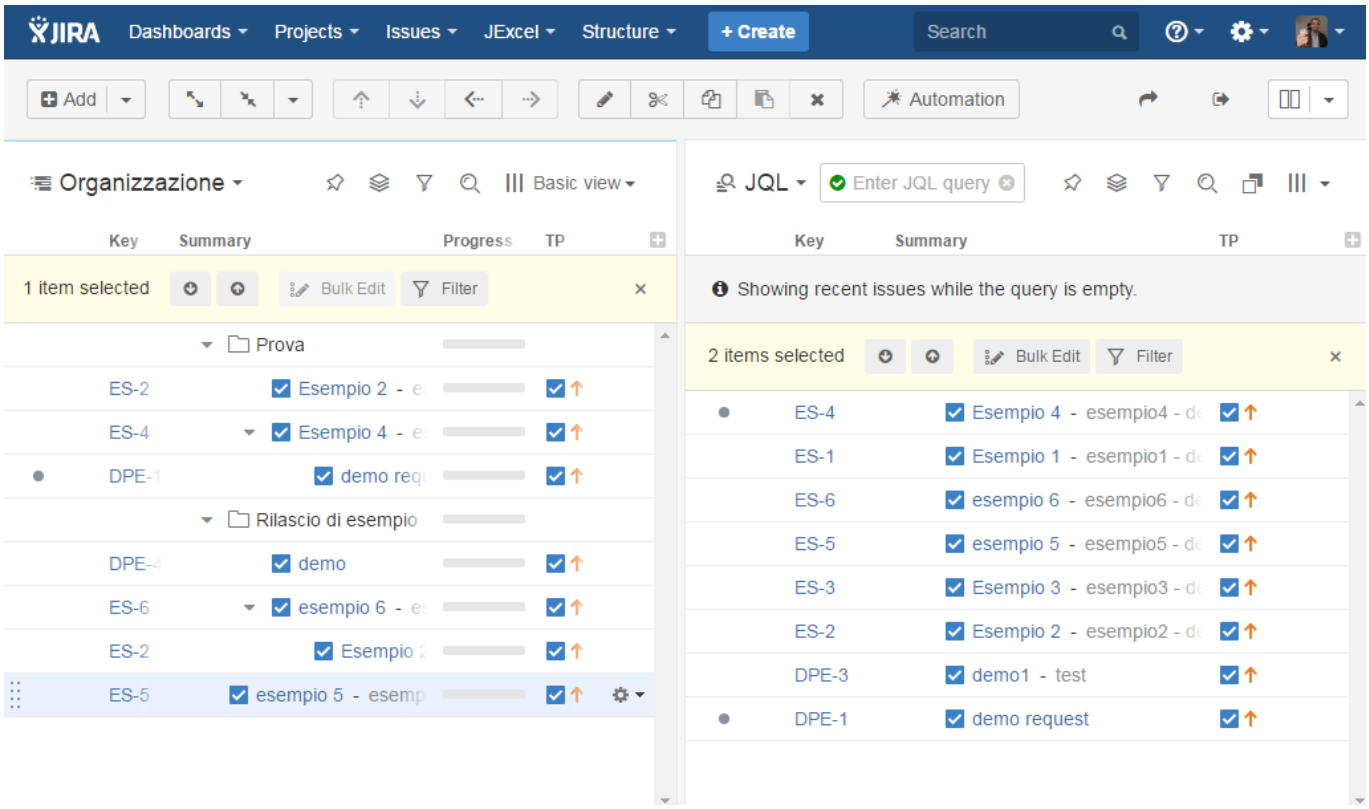
Se la andiamo a selezionare questo è quello che andiamo a scoprire



ovvero entriamo nella gestione delle issues. Attraverso questa schermata è possibile selezionare le issue, utilizzando JQL per la ricerca, per comporre la nostra gerarchia. Adesso proviamo ad usarla. Come prima cosa andiamo a selezionare l'opzione ***Create Structure*** del nuovo menù ***Structure.***

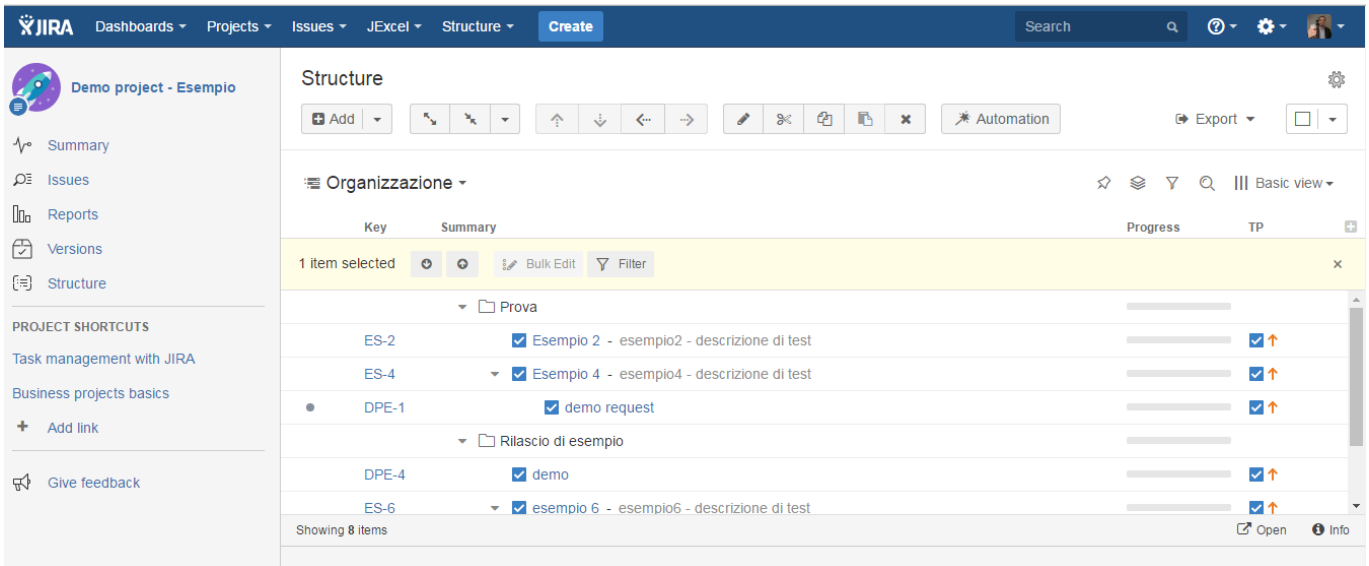


a questo punto possiamo iniziare a gestire la nostra struttura gerarchica:



dalla immagine precedente vediamo che possiamo, con un semplice drag & drop, andare a comporre la nostra struttura gerarchica, selezionando le varie issue e costruendo la nostra struttura.

Al termine, sempre dalla sezione ***Structure*** della pagina principale di gestione progetti, possiamo andare a gestire il tutto.



Conclusioni

Abbiamo visto da vicino questo addon. Possiamo creare in maniera molto semplice una struttura gerarchica che, come struttura di supporto, ci permette di superare la limitazione di task/subtask andando a comporre quanto ci serve. Possiamo anche utilizzare issue di progetti differenti, componendo anche situazioni che richiedono la sincronizzazione di più progetti.

Issue organizer – prova su strada

Test su strada

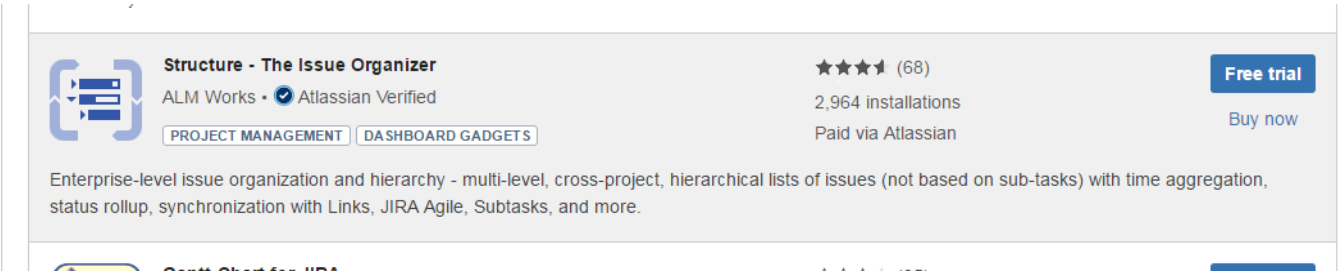
In questo post andremo ad eseguire il test dell'addon Issue Organizer, in modo da verificarne e saggiarne le capacità.



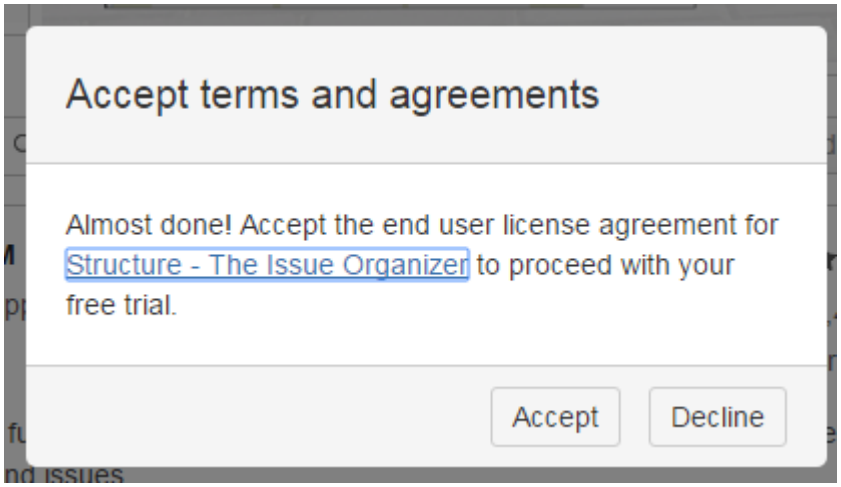
Installiamo l'addon

Come sempre, partiamo dalla fase di installazione. Per questo test utilizzeremo una installazione di JIRA CORE, già precedentemente utilizzata per svolgere altri test

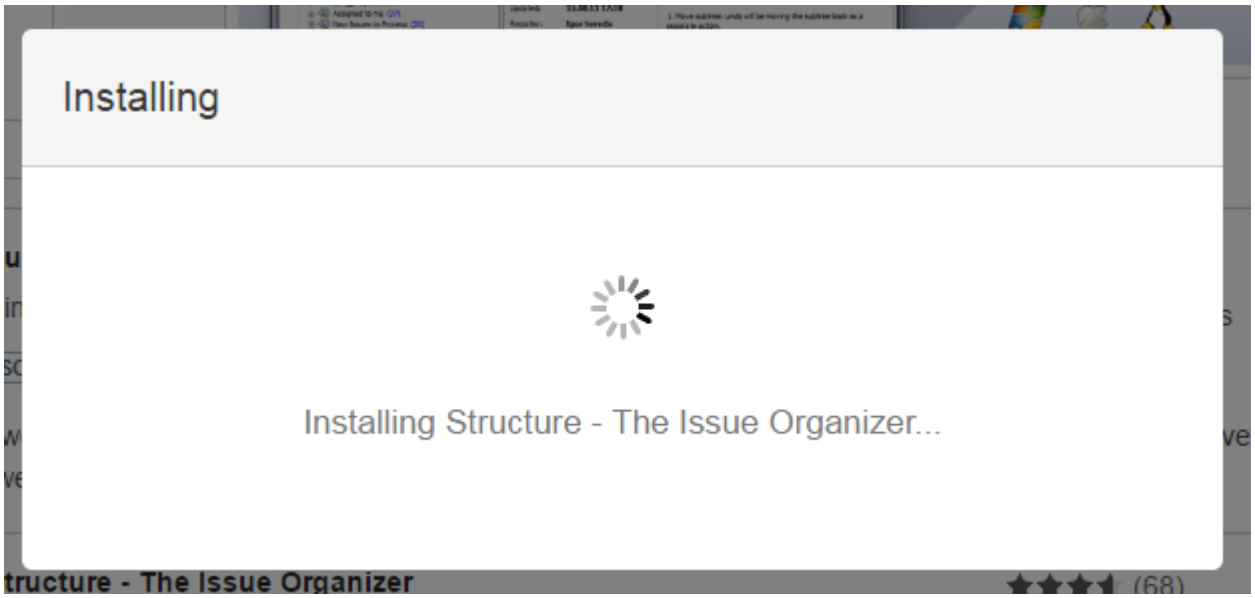
Accediamo alla sezione di amministrazione degli addon e selezioniamo il nostro addon.



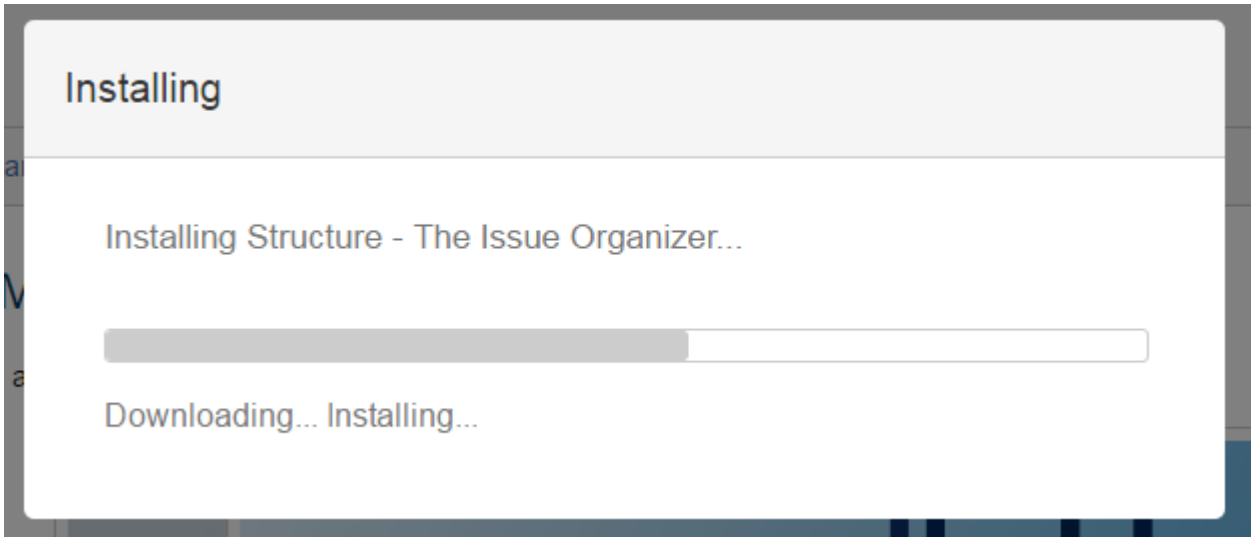
Selezioniamo Free trial e successivamente ….



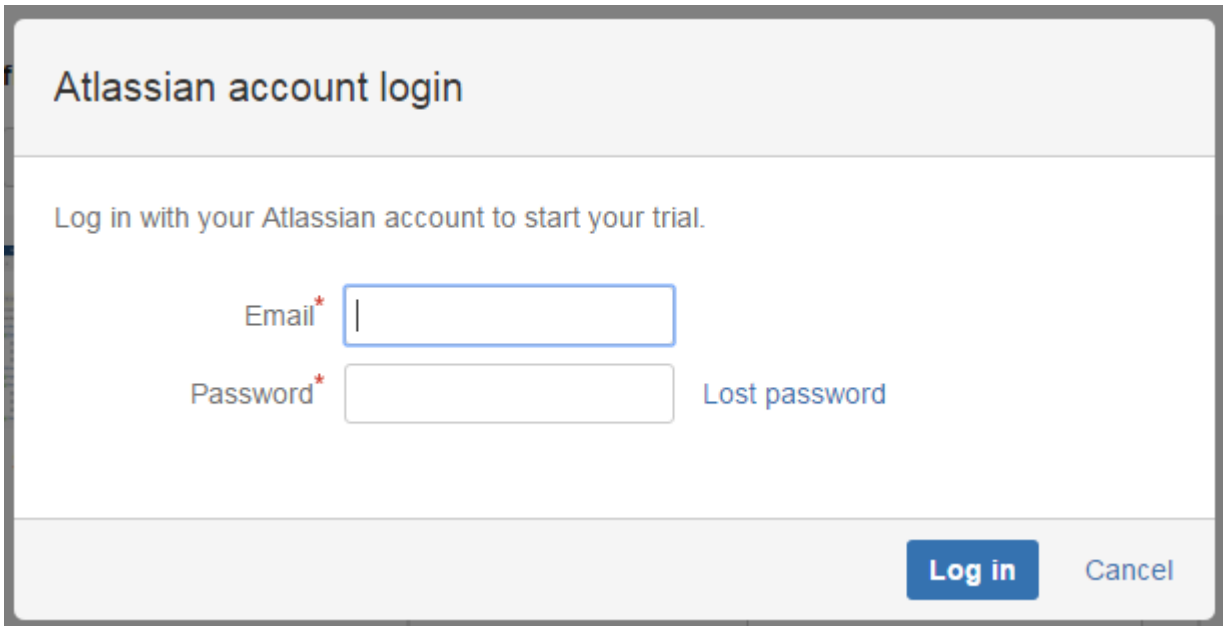
… quindi accettiamo i termini della licenza e procediamo con l'installazione



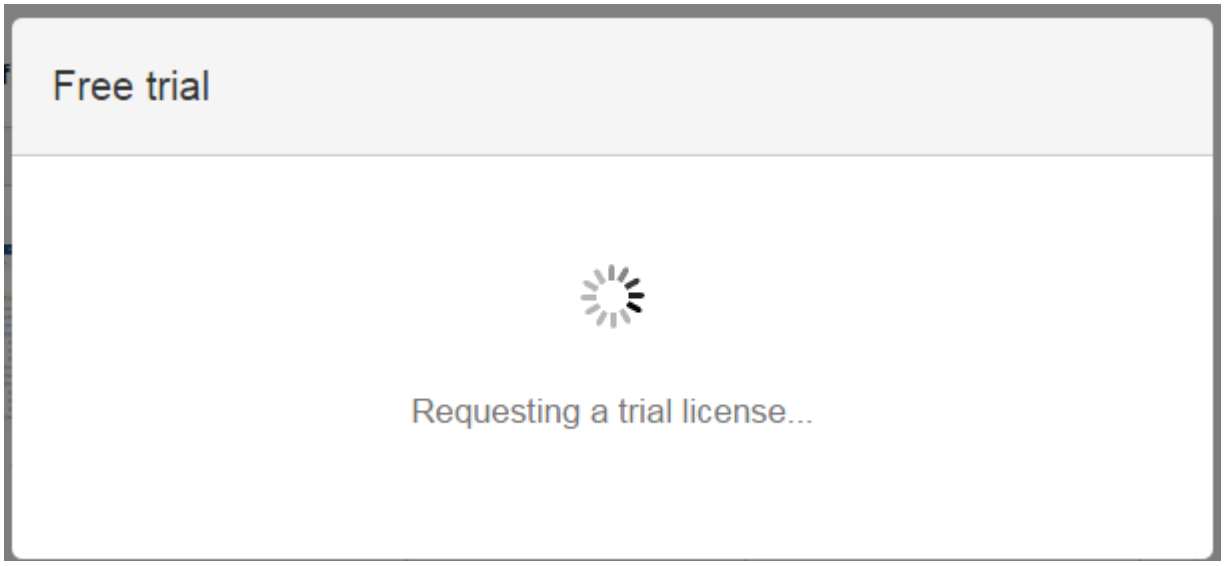
Procediamo con il download automatico dell’addon, come siamo già abituati, quindi procediamo con l’installazione vera e propria.



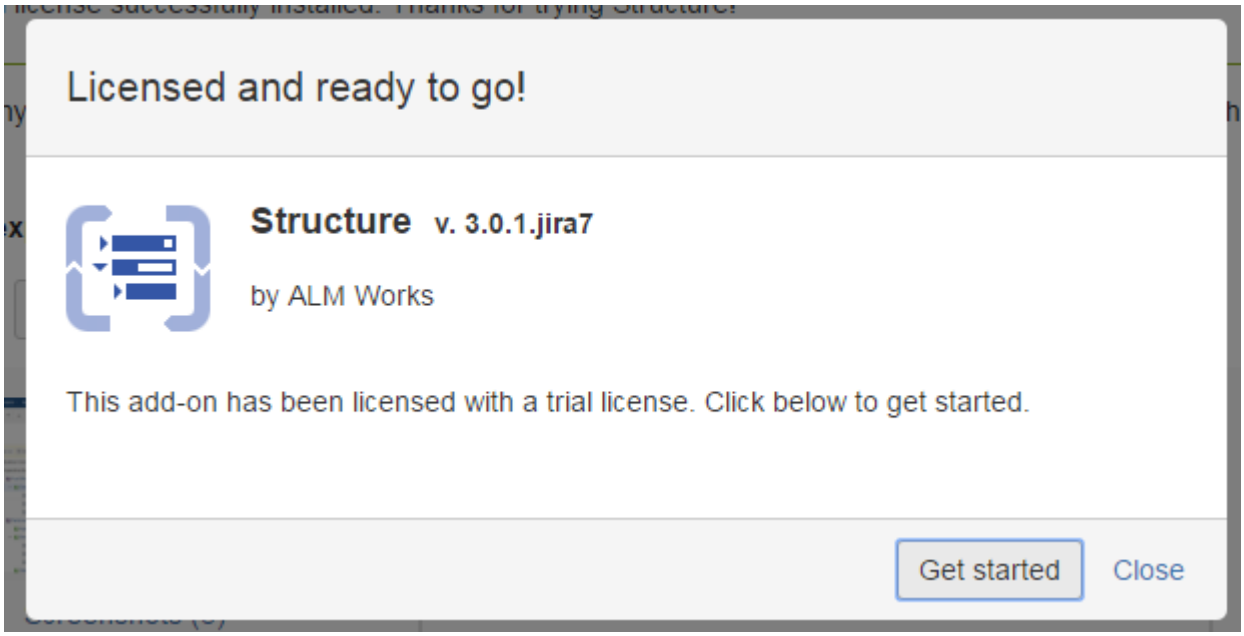
Al termine, dato che abbiamo richiesto una installazione trial, procediamo con ottenere la licenza vera e propria



forniamo quindi le nostre credenziali ….

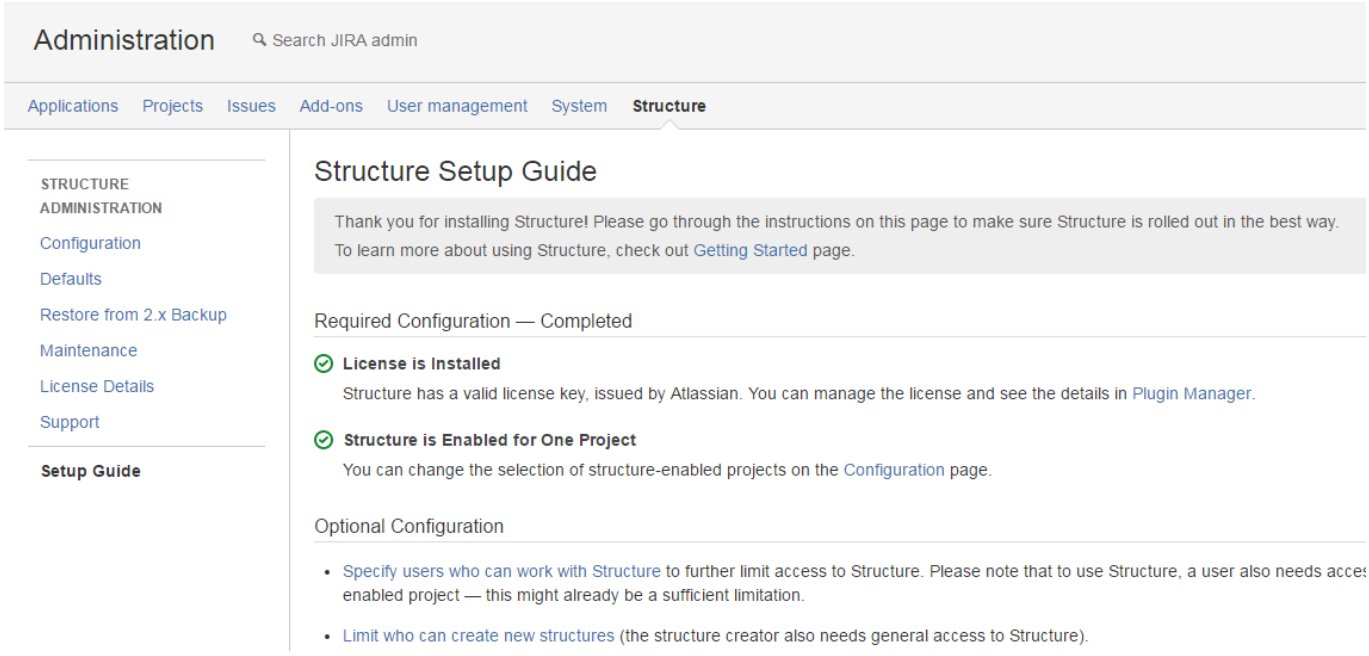


… quindi terminiamo la fase di installazione. Siamo adesso pronti a gestire il tutto.



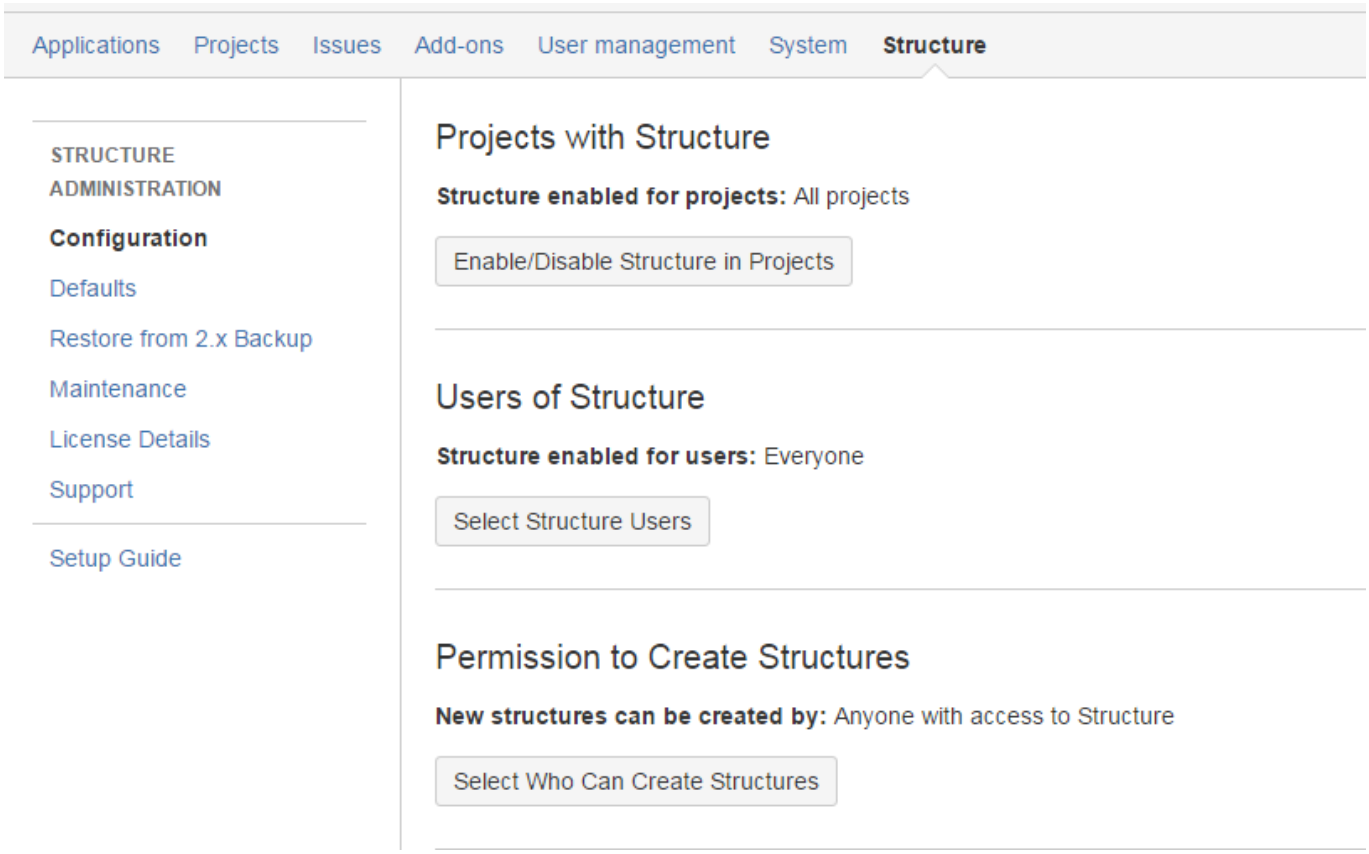
Configurazione

Terminata la fase di installazione, viene subito presentata la parte di configurazione, come mostrato in figura.

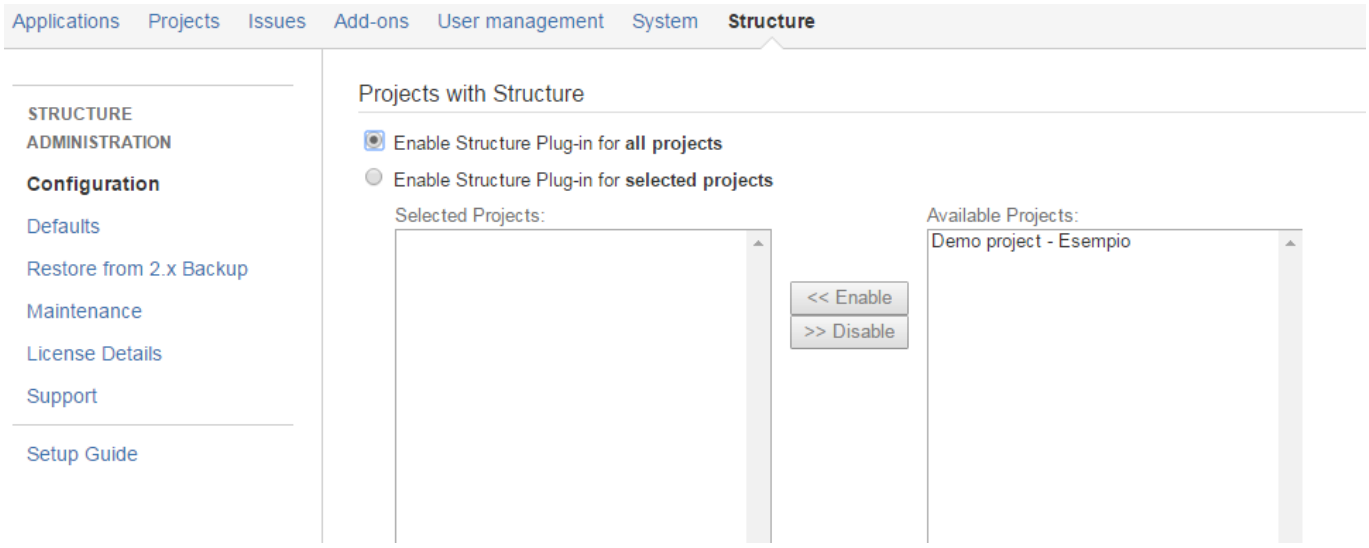


Una piccola annotazione: Notiamo che è stato aggiunto un nuovo TAB, ***Structure***, da cui accedere alla configurazione particolare. Questo significa che NON TUTTI gli addon aggiungono le proprie configurazioni nella sezione di ***Addons***.

Esaminiamo le varie opzioni della configurazione.

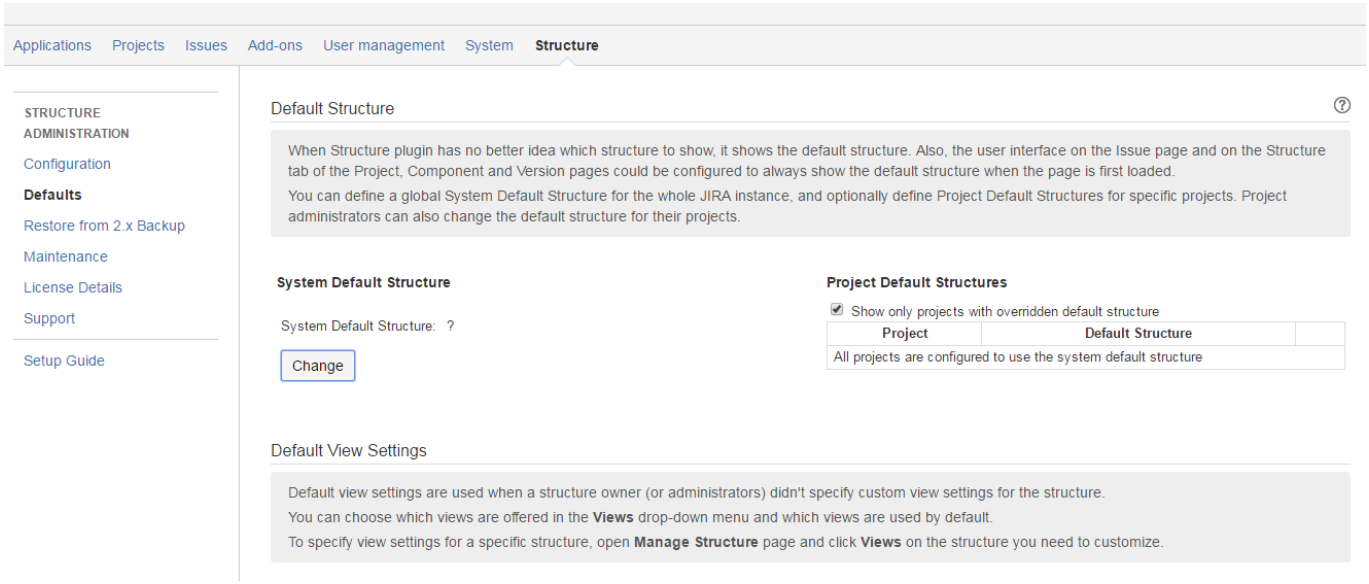


L'opzione ***Configuration*** , permette di settare quali progetti possono usare l'addon, quali utenti e quali permission, come mostrato nella figura precedente. Nella figura successiva abbiamo il dettaglio per comprendere come si settano le varie opzioni.



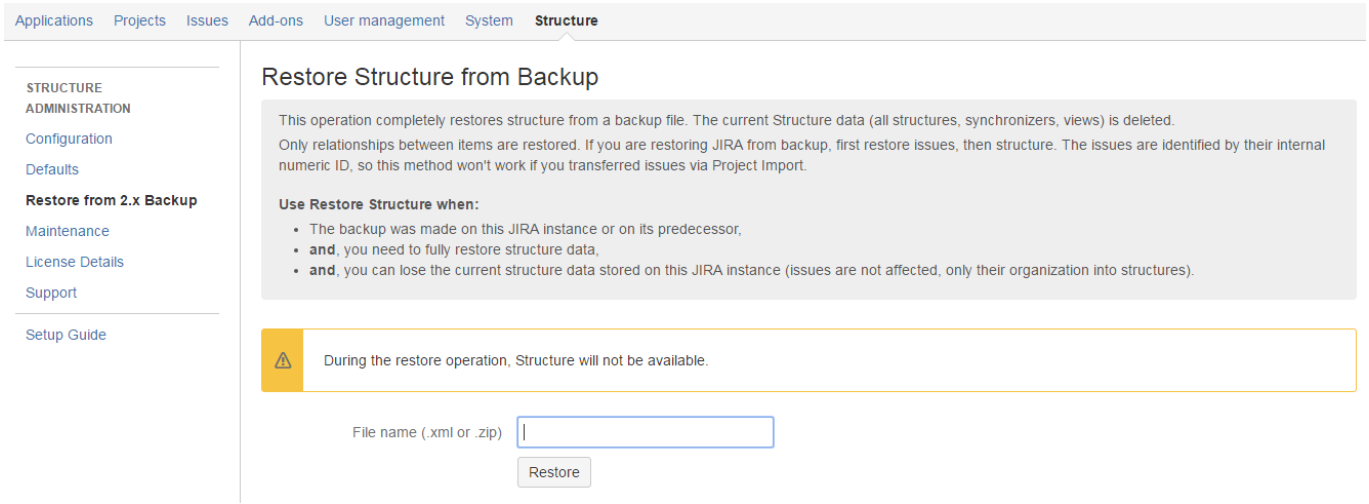
Selezionando l'opzione ***Defaults***, abbiamo le indicazioni di che

cosa sarà visualizzato dall'addon.



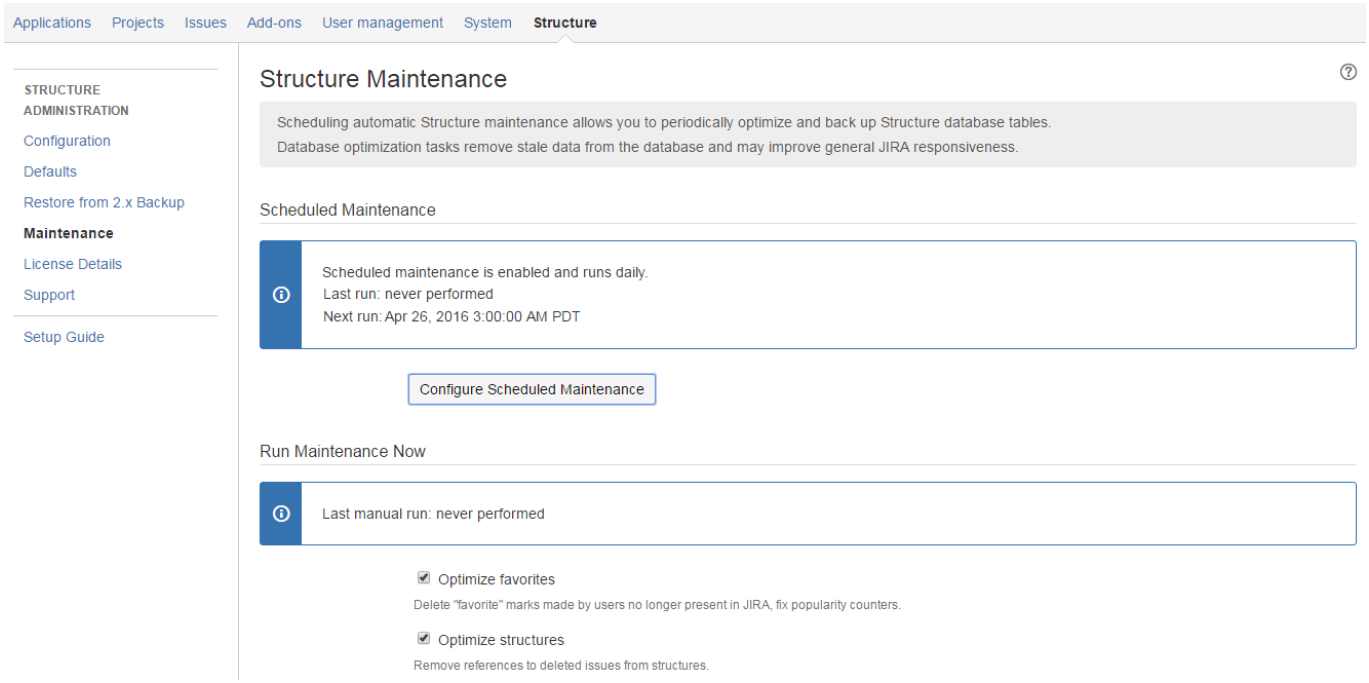
Come mostrato in figura precedente, abbiamo la possibilità di poter modificare la struttura da mostrare e la view di default da mostrare. Se non viene specificato nulla, allora l'addon propone una sua view standard, che potrebbe anche rispondere alle esigenze di molti.

Selezionando l'opzione ***Restore from backup 2.X…***, viene messa a disposizione una opzione per poter importare tutte le configurazioni che sono state esportate da una precedente versione dell'addon.

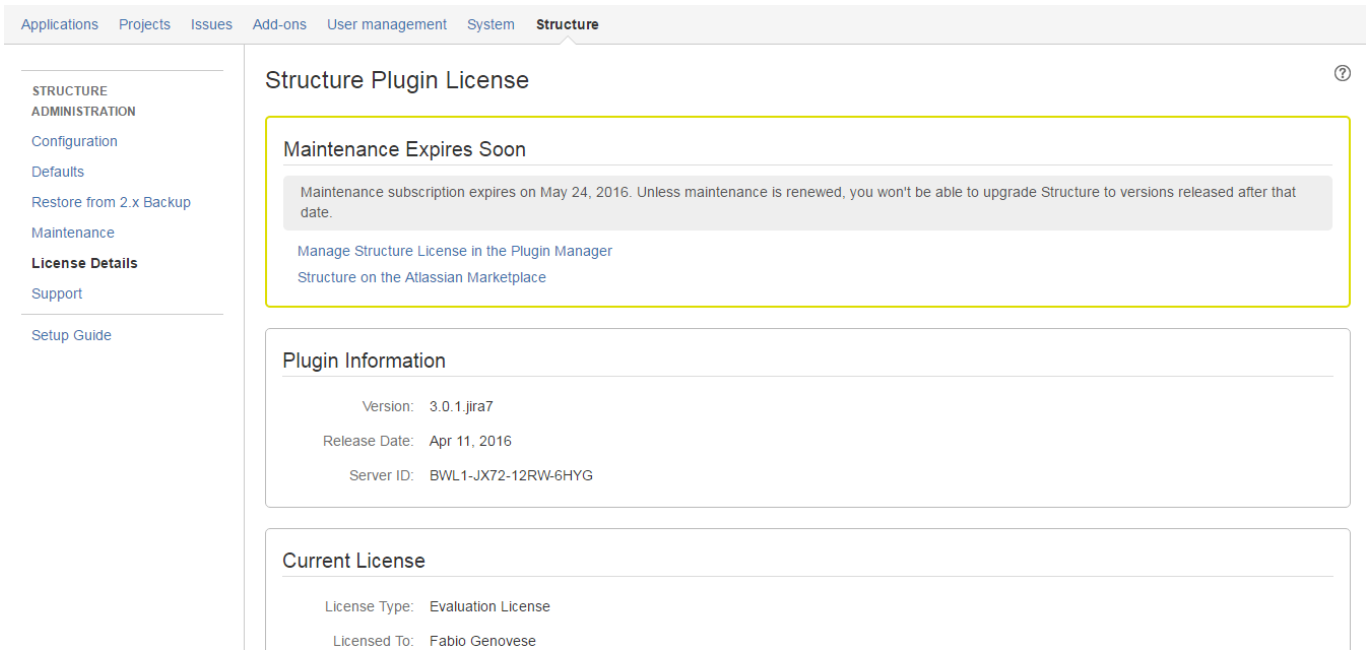


Selezionando l'opzione ***Mantainance***, accediamo al pannello di controllo di manutenzione dell'addon, da dove è possibile

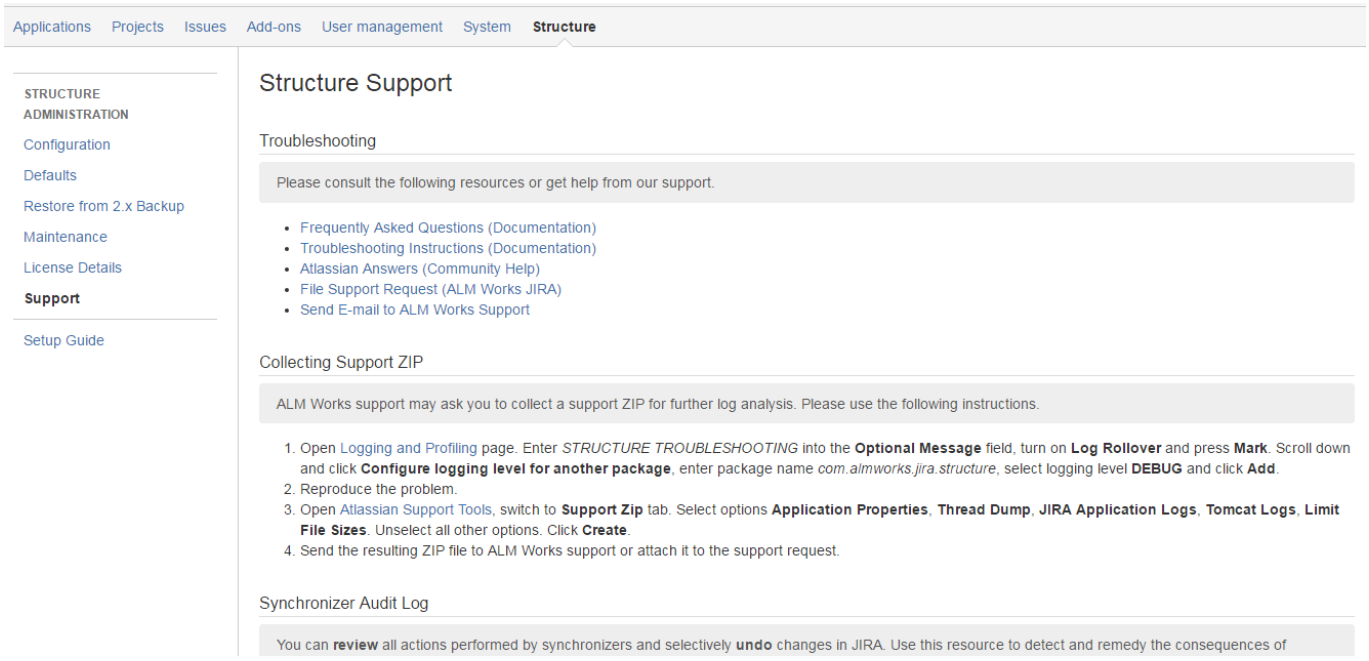
eseguire delle operazioni di ottimizzazione e sistemazione dei dati dell'addon.



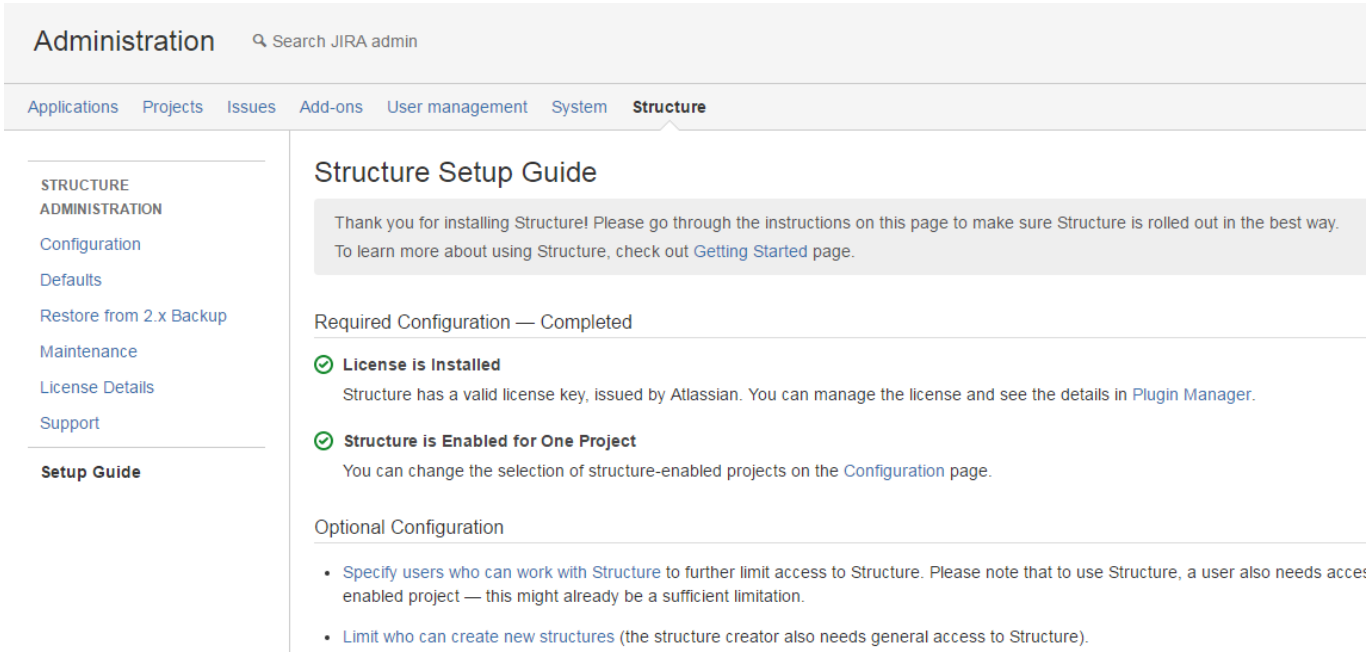
Se selezioniamo l'opzione ***Licence Details***, saranno mostrate le informazioni di licenza dell'addon, come mostrato in figura.



Selezionando l'opzione ***Structure Support***, si accede ad una pagina che fornisce tutte le indicazioni per richiedere il supporto



Selezionando l'opzione ***Setup Guide***, mostriamo nuovamente questa pagina:



Conclusione

Terminiamo qui questo post. Abbiamo visto la fase di installazione e abbiamo descritto le opzioni di configurazione, che sono abbastanza complesse. Proseguiamo il

test nel prossimo post.

Reference

Maggiori informazioni sull’addon sono presenti qui.