

Product Requirements

Product Requirements

In questo post andremo ad esaminare come Confluence ci può aiutare nella redazione di product requirements, basandoci sull'articolo del [blog ufficiale di Confluence](#)



Andiamo al sodo

Sfruttiamo le funzionalità che Confluence ci mette a disposizione :-). Per realizzare il tutto abbiamo a disposizione opportuni template che ci aiutano nel creare questi contenuti particolari. Semplicemente andiamo a selezionare il template del **Product Requirements** e andiamo a costruire la nostra pagina.



Seguiamo l'autocomposizioni, cui la Atlassian ci ha abituati in tutti questi anni.

Let's get started

With product requirements you can...

- 1 Define document properties**
Add properties like status and owner to your document to make it easy to organise and sort your product requirements.
- 2 Create requirements**
The customisable template brings structure and consistency to your product requirements. If you use JIRA you can create stories right from the requirements page.
- 3 Track progress**
See the status of all your requirements at a glance. Sort by properties like status and release, or access linked JIRA issues from your requirements pages.

Target release	1.0
Epic	TIS-7 - Mobile IN PROGRESS
Document status	DRAFT
Document owner	@ Ryan Lee
Designer	@ Cassie Owens
Developers	@ William Smith

Don't show this again

Back **Create** Close

Come possiamo vedere, ci vengono fornite tutte le indicazioni del caso. Non possiamo sbagliare ☐

Quindi selezioniamo **Create**. Il risultato è il seguente

La Bottega / Pages / La Bottega   

Titolo della pagina

Page properties

Target release	<i>Release name or number</i>
Epic	<i>Link to related JIRA epic or feature</i>
Document status	DRAFT
Document owner	Fabio Genovese [Administrator]
Designer	<i>Lead designer</i>
Developers	<i>Lead developer</i>
QA	<i>Lead tester</i>

A questo punto compiliamo i vari campi e lasciamo libera la nostra fantasia, andando a scrivere tutte le informazioni che servono :-).

Che altro possiamo fare?

Ovviamente questo è il solo punto di partenza. Confluence ci mette a disposizione tutta una serie di strumenti che permettono di poter personalizzare in base a tutte le esigenze. Possiamo infatti costruire dei template ad hoc, espressamente dedicati per determinati Product Requirements, oppure avere delle diversificazioni in base ai progetti o alle tipologie di progetti. Oppure possiamo sfruttare altre caratteristiche, di cui abbiamo più che ampiamente parlato, quale il [Yoikee Creator, Mind Mapping](#) □

In questo Confluence ci aiuta in questa operazione. Possiamo aggiungere altri addon, come descritto in altri [post](#), per poter estendere le funzionalità e aiutarci nel nostro lavoro.

The logo for ATLASSIAN is rendered in a bold, blue, sans-serif font. The letter 'A' is significantly larger and more stylized than the other letters, featuring a sharp, upward-pointing peak. The remaining letters 'T', 'L', 'A', 'S', 'S', 'I', 'A', 'N' are uniform in size and style, following a clean, modern aesthetic.

Ripeto il consiglio: Lasciate che il vostro unico limite sia la fantasia :-D.

Conclusioni

Abbiamo visto un ulteriore esempio di come Confluence si dimostra uno strumento eccezionale, sempre rispondente alle nostre esigenze e con caratteristiche superiori.

Reference

- [Articolo del blog](#)

Kanoah Checklist – Ultime novità

Ultime novità

In questo post riportiamo le ultime novità su Kanoah Checklist.



Segnaliamo che è stato aggiunto il supporto per JIRA 7, come gli stessi autori segnalano nella [Release Note](#) del componente.

L'addon adesso si integra meglio con le ultime caratteristiche introdotte con la versione 7, di cui abbiamo parlato nei precedenti post e nel [post](#) dedicato alla presentazione a Bologna.



Abbiamo una ulteriore conferma del fatto che la Kanoah mette molta cura nei propri prodotti, cercando sempre di fornire il meglio ai propri utenti e mettendoli sempre in condizione di poter lavorare al meglio. Questo è sicuramente un indice di garanzia nei confronti del cliente ☐

Script Runner per JIRA – First Look

ScriptRunner – First Look

In questo post andremo ad esaminare lo ScriptRunner, un addon per JIRA che consente di poter fare delle cose ... molto interessanti.



Di che cosa si occupa questo addon?

Fondamentalmente consente di poter automatizzare alcune operazioni su JIRA, sfruttando le potenzialità di [Groovy](#), un potente linguaggio che è simile al Java:

```
Command Prompt - groovysh
C:\rmoug\td2010>groovysh
Groovy Shell (1.6.5, JVM: 1.6.0_14)
Type 'help' or '\h' for help.
-----
groovy:000> "This is a string."
===> This is a string.
groovy:000> "This is a string.".length()
===> 17
groovy:000> "25"
===> 25
groovy:000> <50+50>.toString()
===> 100
groovy:000> "abcdef".multiply(5)
===> abcdefabcdefabcdefabcdefabcdef
groovy:000> "Abc" + "Def"
===> AbcDef
groovy:000>
groovy:000> "abcdef"*5
===> abcdefabcdefabcdefabcdefabcdef
groovy:000>
```

che consente di poter aggiungere del ... valore in più sul nostro JIRA



Vediamo cosa possiamo fare con questo addon.

- Aggiungere delle funzionalità **Built-in**, che ci aiutano nel semplificarci la vita:



- Estende JQL con alcune funzioni che ci ... aiutano nel nostro lavoro: **▲ ATLASSIAN**
- Possibilità di aggiungere funzioni custom:



- Possibilità di poter estendere i Workflow con funzionalità aggiuntive;
- Rendere certi campi di JIRA obbligatori, sotto certe condizioni;
- tante tante altre ancora ☐

Conclusioni

Abbiamo un valido aiuto nel nostro lavoro di tutti i giorni. Uno strumento che ci consente di estendere le funzionalità di JIRA come vogliamo. Nei prossimi post, andremo a testare sul campo, come siamo già abituati :-D. Vedremo di saggiarne le potenzialità ed i limiti.

Atlassian Partner

***Annuntio Vobis Gaudium
Magnum***

E' con grande gioia che annuncio che il ***buon artigiano*** è adesso ***Atlassian Partner***.

Voglia di fare e perseveranza sono stati premiati con questo grande risultato :-), ma si tratta solo del punto di partenza per nuove imprese e nuovi risultati.



JQL – Esempi di uso

JQL – Esempi di uso

Vediamo in questo post alcuni esempi di uso del [JQL](#), ovvero del JIRA Query Language. Si tratta di alcuni esempi di utilizzo che possono aiutare nella vita di tutti i giorni ☐



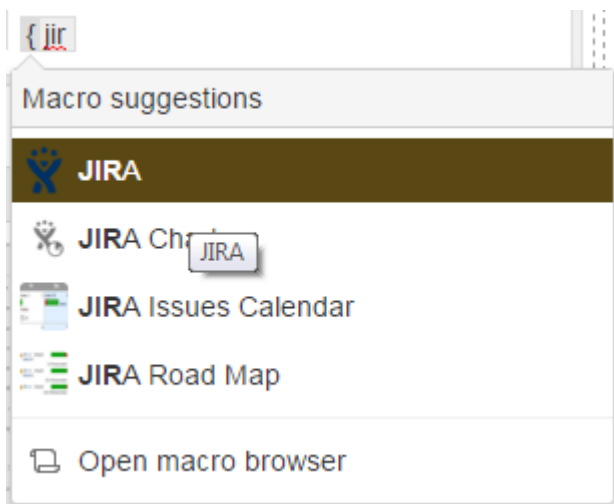
Andiamo al dunque

Dopo aver presentato il JQL, vediamo dove è possibile utilizzarlo per svolgere il nostro lavoro.



Confluence

Su Confluence abbiamo diverse possibilità di utilizzo. Abbiamo già mostrato alcuni di essi proprio nella gestione degli Asset Management, dove abbiamo anche collegato alle pagine dell'asset le Issue di riferimento attraverso una apposita macro: **JIRA**; come mostrato in figura:



Quindi, sfruttiamo il JQL per andare a selezionare le Issue o la Issue che ci interessa.

Insert JIRA Issue/Filter

Search

Create New Issue

Recently Viewed

OTHER JIRA CONTENT

JIRA Charts

project = SA

Search using any issue key, search URL, JIRA link, JQL, plain text or filter

No search results found.

Display options

Display as

- Single issue
Display the macro as a single issue.
- Total issue count
Display total number of issues as a link. E.g. 12 issues
- Table
Customise your columns below.

Maximum issues: 20

Leave empty to get all issues.

Columns to display: Key x, Summary x, Issue Type x, Created x, Updated x, Due Date x, Assignee x, Reporter x, Priority x, Status x, Resolution x

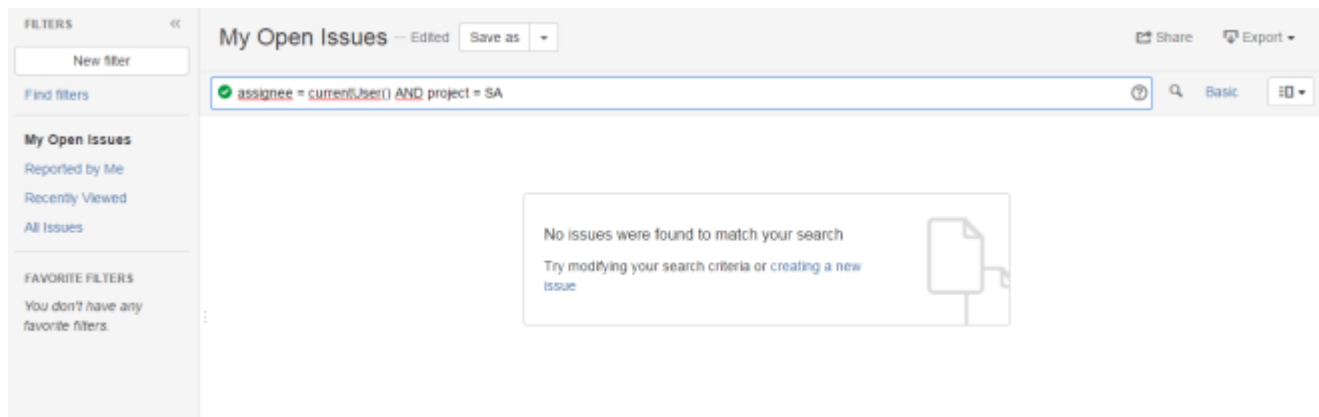
Select Macro Hint: type "Ctrl+Shift+J" in the editor to quickly access this dialog.

Insert Cancel

Come possiamo vedere, esaminando l'ultima immagine, quello che notiamo è che andiamo a selezionare le issue semplicemente andando a comporre la nostra query JQL in maniera opportuna. Quindi, senza fare alcuna fatica, settiamo le proprietà e le informazioni che vogliamo vedere, in modo quasi istantaneo □

JIRA

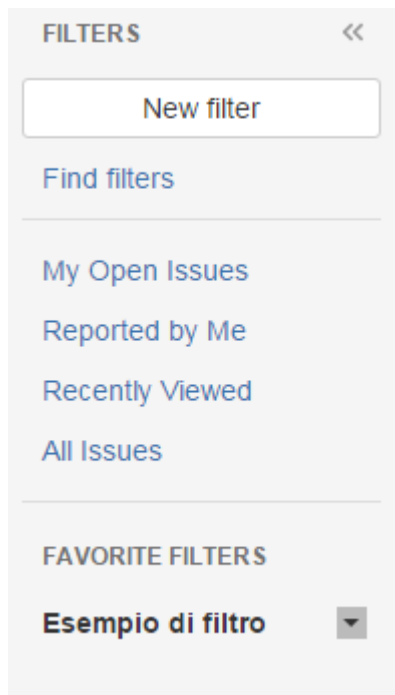
Su JIRA semplicemente andiamo ad accedere alla sezione delle Issue, dove andiamo ad impostare sia i filtri che ad eseguire tutte le interrogazioni che ci servono.



Una volta identificati i filtri che servono, possiamo semplicemente andare a salvare le query come dei filtri. Semplicemente selezioniamo il tasto **Save as** posto in alto (vedi precedente figura):

The image shows a 'Save Filter' dialog box. It has a title bar 'Save Filter'. Below it, there's a label 'Filter Name *' and a text input field containing 'Esempio di filtro'. Below the input field, there's a hint 'Enter a name for this Filter'. At the bottom right, there are two buttons: 'Submit' and 'Cancel'.

Una volta salvato, possiamo referenziare il filtro semplicemente dalla toolbar laterale:



Conclusioni

Abbiamo visto alcuni esempi di applicazione di JQL. Nei prossimi post andremo a vedere come possiamo estenderlo in maniera opportuna.

JIRA 7 a Bologna

JIRA 7 a Bologna

Il 27 Ottobre si è svolto a Bologna un evento dedicato a JIRA 7. Ho avuto la fortuna di poter partecipare all'evento e di poter vedere le ultime novità di JIRA 7. In questo post, e nei seguenti, cercherò di riassumere quanto è stato presentato in

questo evento □



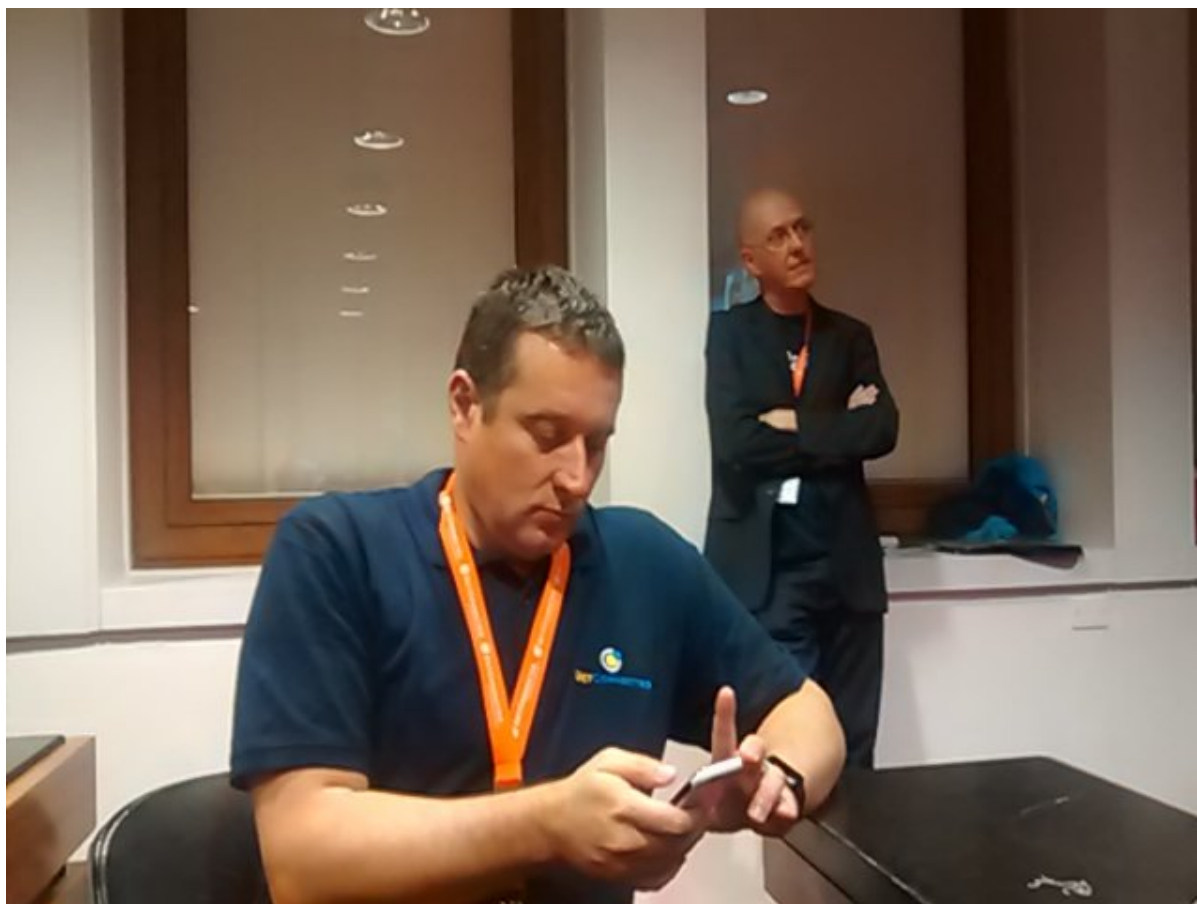
Team più performanti con JIRA

Il core della presentazione verteva sulla presentazione delle novità di ***JIRA 7***, presentate da un ospite di eccezione: ***Vladimir Cavalcanti***; EMEA Experts Manager Atlassian.



Dopo una breve introduzione, da parte di **Alessandro Rizzoli** di GetConnected, Vlad Cavalcanti ha iniziato una presentazione, tutta dedicata alla Atlassian, fornendo una panoramica sui vari prodotti e sui clienti (ben [50.000](#)), tra cui anche la

[NASA](#), dove la Atlassian ha fornito il suo contributo per la missione Rover Mars.



Vlad ha poi subito introdotto le ultime novità su JIRA 7, descrivendo le nuove pacchettizzazioni:

- [JIRA CORE](#)
- [JIRA SOFTWARE](#)
- JIRA Service Desk

Durante la presentazione ha subito evidenziato una delle domande più comuni, tra quelle che solitamente vengono poste durante le presentazioni: ***Gli Addons rimangono compatibili?*** La risposta: ***Rimangono compatibili con la nuova versione.***

Ha quindi evidenziato le indicazioni per le licenze: Nel caso dei vari prodotti, vince la licenza con il taglio più alto.

Vlad Cavalcanti ha quindi concluso il suo intervento mostrando un semplice caso d'uso di tutti i giorni, un esempio di situazione che si è presentata nel suo lavoro e di come, i

prodotti della Atlassian, lo hanno aiutato , soffermandosi come questi strumenti aiutano i team non IT



Si è quindi soffermato anche sulle ultime novità di Hipchat, segnalando che sarà presto rilasciata la versione Server per questo prodotto.



Ha concluso il suo intervento parlando degli [Ship-It Days in Atlassian](#), dove tutti aiutano con nuove idee. Ripeto: TUTTI. Una piccola curiosità: JIRA Service desk è nato a seguito di uno Ship-It :-).

Segue una presentazione, a cura di Cecilia Barbardi, sulla Continuous integration e sulla esperienza in azienda GetConnected.



Cecilia ha fornito una ottima spiegazione dei processi aziendali, sulla realtà e su come i prodotti della Atlassian, attraverso la loro integrazione, sono un valido supporto. :-). Passare da un sistema di scambio di file ad un sistema di integrazione, condivisione, scambio, ma soprattutto UNICO :-).

Questo è il valore aggiunto dei prodotti Atlassian.



Seguita una ottima presentazione di Luca, su come JIRA Service Desk aiuta i gruppi di lavoro nello svolgere il proprio lavoro, mettendo a disposizione sistemi semplici e veloci per la gestione delle anomalie e delle informazioni.



L'evento è stato chiuso da un intervento su **JIRA Portfolio**: Il prodotto più giovane della famiglia Atlassian. **Federico Sita** ha mostrato, con una demo molto completa e chiara, come questo prodotto può essere di aiuto nelle simulazioni e delle pianificazioni trasversali i vari gruppi di sviluppo.

Conclusioni

Un grande evento per Bologna, una occasione di potersi confrontare con esperti del settore e con varie esperienze sull'argomento. Sicuramente da ripetere

Un semplice esempio di Release Notes

Altro esempio di uso

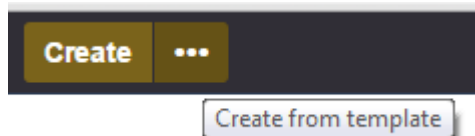
In questo post andremo ad esaminare quanto descritto nel post del blog ufficiale della Atlassian, cercando di spiegarlo meglio e di fornire quanti più dettagli possibili.



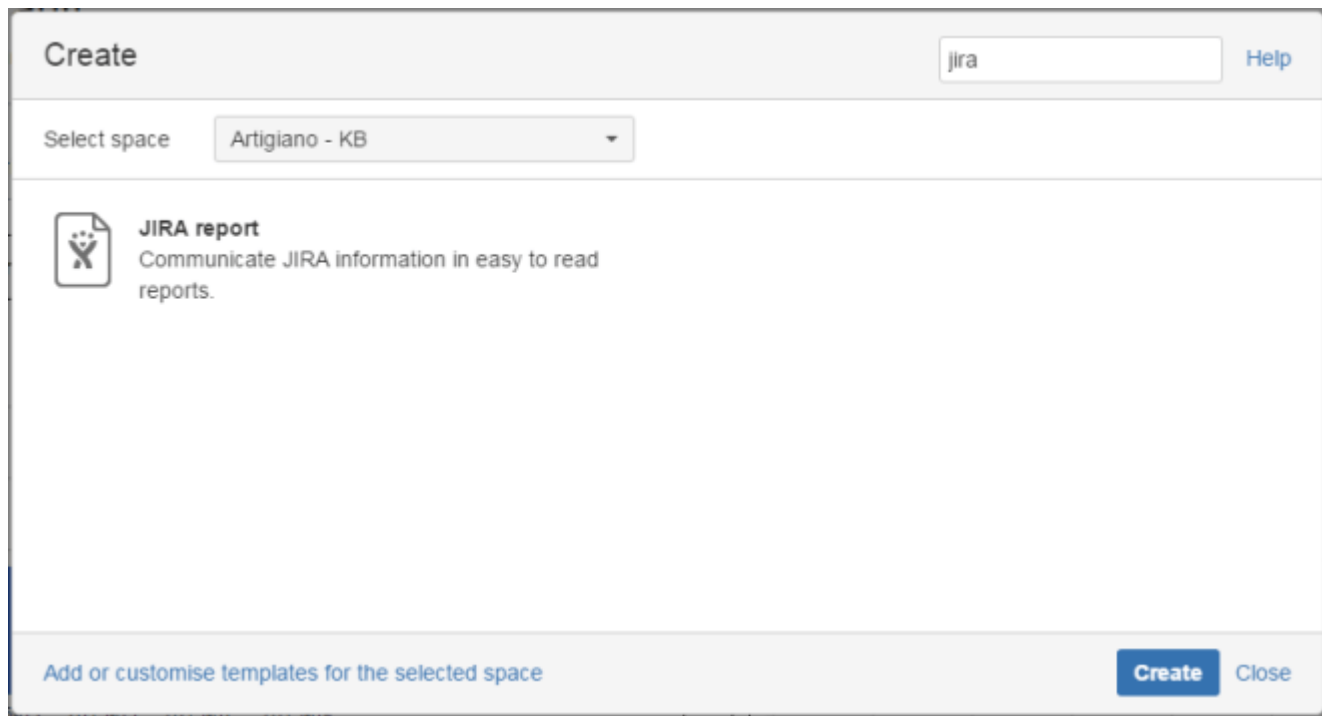
Procediamo...

Andiamo ad esaminare una funzionalità di Confluence, che ci consente di interfacciarci a JIRA, in modo da poter andare a leggere le informazioni delle issue, relative ad un rilascio, e pubblicare il risultato su di una pagina , che raccoglierà tutte queste informazioni.

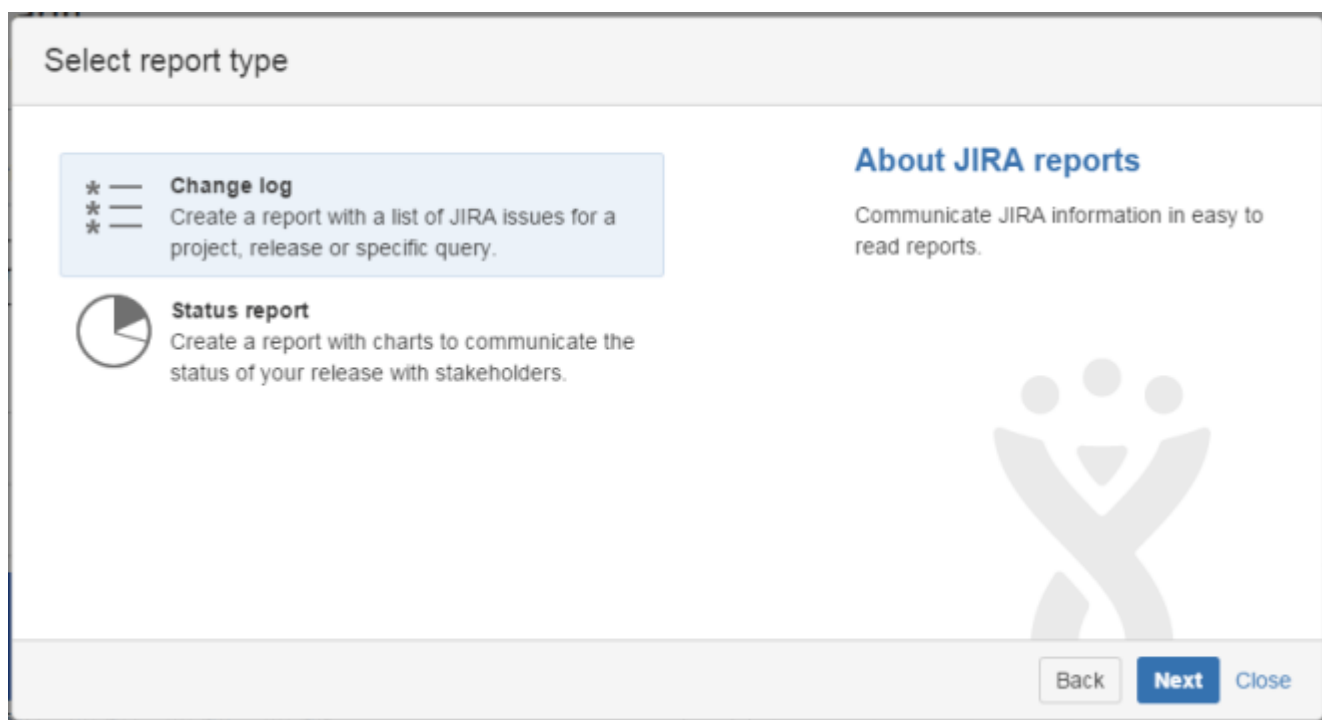
Andiamo a selezionare dal tasto:



e selezioniamo il template **JIRA report**.



quindi selezioniamo **Change log**:



A questo punto, attendiamo che l'autocomposizione di Confluence vada a leggere le informazioni dei progetti JIRA e ci chieda quali parametri passare:

Create a change log

Project

Fix version(s)

[Switch to advanced](#)

Title

About change logs
Keep a log of your teams progress or communicate deliverables. Generate a static or dynamic list of JIRA issues from a saved search, JIRA URL or JQL query.

Possiamo andare a imputare i parametri sfruttando la modalità semplice della autocomposizione, oppure, come mostrato nella seguente figura:

Create a change log

JIRA Query

[Switch to simple](#)

Title

Automatically update issue status and summaries from JIRA

About change logs
Keep a log of your teams progress or communicate deliverables. Generate a static or dynamic list of JIRA issues from a saved search, JIRA URL or JQL query.

possiamo impostare una opportuna query JQL, che ci consente di poter andare a selezionare le issue che ci servono :-). Il risultato è la seguente pagina:



Possiamo identificare le seguenti sezioni:

- **Page properties**, dove andiamo ad inserire dei metadati che possono risultare utili nel caso di report o recupero delle informazioni;
- **Riassunto**, dove riportiamo un riassunto delle informazioni;
- **Punti chiave del rilascio**, dove riportiamo che cosa si è voluto rilasciare
- **JIRA Issue**, dove è riportato l'elenco delle JIRA ISSUE coinvolte.

Come si vede, abbiamo a disposizione un semplice template che ci consente, attraverso l'uso di pochi passi e delle macro di connessione ai dati JIRA, di poter reperire le varie informazioni, in modo semplice e veloce.

Conclusioni

Abbiamo visto fino a qui delle indicazioni su come realizzare una Release Notes. Una cosa vorrei far notare: Questo funziona anche se non si tratta di un software, ma del rilascio di una nuova macchina, che presenta dei miglioramenti rispetto alla precedente edizione. Analogo discorso per un manuale. Ricordate quindi che quanto dico si può tranquillamente applicare anche ai progetti ed alle realtà NON IT, non di informatica :-).

Reference

- [Articolo del Blog della Atlassian.](#)
-

JQL – First look

JQL aka JIRA Query Language

In questo post andiamo ad esaminare questo semplice linguaggio, che consente di poter eseguire delle interrogazioni sulle ISSUE JIRA, come se si trattasse di usare una sintassi simile-SQL.

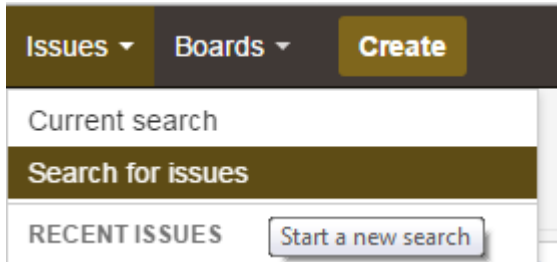


Che cosa è?

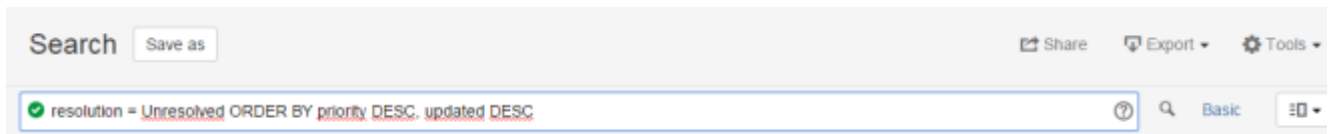
La prima cosa che facciamo è rispondere ad una semplice domanda: Che cosa è JQL? Come indicato nel titolo, JQL sta per JIRA Query Language ed è un linguaggio simil-sql che la Atlassian mette a disposizione per ricercare issue su JIRA.

JQL è stato studiato per TUTTI. Non è inteso come uno strumento per pochi, ma per tutti coloro che ne abbisognano. Consente di poter cercare facilmente ed agevolmente le Issue che interessano o abbisognano.

Possiamo agevolmente accedervi dal menù: **ISSUE** -> **Search for issue** come mostrato in figura.



Viene quindi proposta la form JIRA per le ricerche:



Dalla precedente immagine già abbiamo una prima indicazione di come si scrivono queste simil-sql query. In particolare:



Semplicemente andiamo a scrivere delle sequenze di:

- Campo
- operatore
- valore



JIRA ci aiuta con dei suggerimenti (vedi immagine precedente), fornendo dei suggerimenti nella composizione della query **JQL**. Se torniamo all'esempio della precedente immagine, notiamo che stiamo interrogando JIRA per capire quali sono le issue che risultano **NON RISOLTE**, ovvero **resolution = Unresolved**, ordinate per priorit  ed aggiornamento.

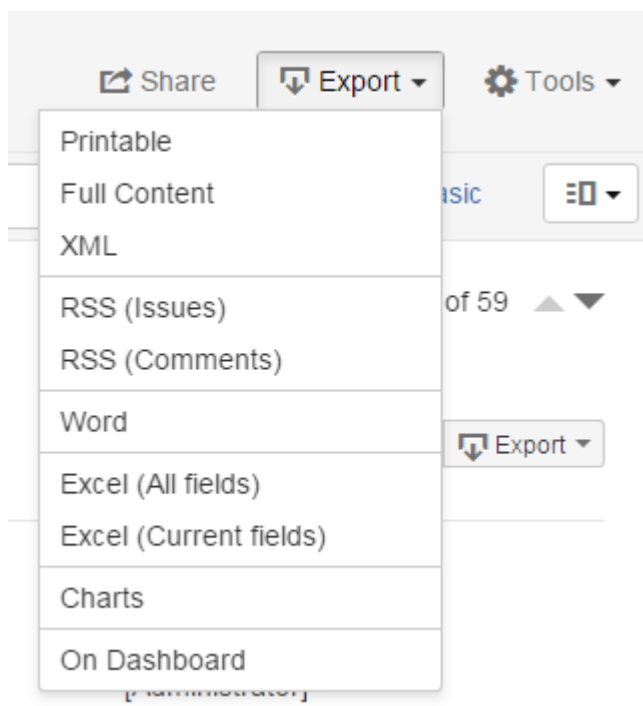
Notiamo anche che JIRA stesso ci indica se la query JQL   corretta o meno



aggiungendo una icona che indica se OK o meno. Il risultato è sicuramente ottimo.

Grande risultato, ma possiamo esportare i dati?

Certo che sì. La risposta è positiva. Una volta che abbiamo selezionato le informazioni che ci interessano, le issue che vogliamo, possiamo esportarle su Excel :-).



Attraverso l'apposito menù **Export**, visualizziamo le possibili alternative. Possiamo esportare tutte le informazioni che vogliamo. In questo modo possiamo eseguire ulteriori indagini sulle informazioni presenti in JIRA ☐

Conclusioni

Abbiamo iniziato a scoprire questa piccola perla. Si tratta di una funzionalità che consente di poter reperire issue da JIRA, in base alle nostre esigenze.

Reference

- [Manuale tecnico di JQL](#)
- [Articolo del blog ufficiale Atlassian](#)

Organizziamo una roadmap in Confluence

Organizziamo una roadmap

In questo post andremo ad esaminare come poter realizzare una [roadmap](#) sfruttando le potenzialità di Confluence. Riprendiamo quanto riportato nel [post ufficiale della Atlassian](#) e proviamo a sviluppare l'argomento.



Procediamo

Andiamo subito al dunque. L'obbiettivo è quello di sfruttare le funzioni, di cui dispone Confluence. La prima cosa da fare è quella di avere subito chiari i seguenti punti:

- Identifichiamo i punti cardini della nostra roadmap
- Inseriamo tutte le informazioni nelle pagine, organizzandole nel modo opportuno
- indirizziamo le persone verso le informazioni che le riguardano.

Possiamo iniziare a riassumere il tutto attraverso la seguente pagina:



Andiamo ad esaminarla ;-), analizzandola in ogni singolo punto.

Come prima cosa vediamo che le informazioni sono organizzate in due colonne, consentendo una migliore organizzazione della pagina. Un suggerimento che sempre ai vari utenti, è di fare un sapiente uso delle sezioni. Dalla versione 4 di Confluence, l'editor mette a disposizione la possibilità di poter organizzare varie sezioni già in colonne. A loro volta, è possibile aggiungere ulteriori sezioni (con il tempo e l'avanzare delle versioni, Confluence si è notevolmente evoluto □):

In questa pagina notiamo anche l'uso del componente standard Userprofile, che abbiamo ampiamente usato e spiegato in vari [post](#) e della [Roadmap](#), anche questa ampiamente spiegato :-). Infine, abbiamo le [Page Properties](#), che ci aiutano tantissimo.

La prima cosa che notiamo è che in questa pagina riusciamo a riassumere tantissime informazioni. Abbiamo subito le

indicazioni del gruppo di lavoro (lo Userprofile è uno strumento molto versatile ☐) ed anche il piano di battaglia relativo al lavoro da svolgere.



Questa è una delle pagine principali del progetto, dove abbiamo la prima impressione. E' una buona abitudine creare sempre una pagina principale che dia subito le informazioni principali. Poi da li, occorre inserire tutti i riferimenti alle altre informazioni.



L'uso della macro **Status** è utilissimo per arrivare ad identificare uno stato dello sviluppo, di un documento, di una procedura collegata o di qualsiasi altra indicazione relativa al progetto. Già in altri post, quali ad esempio dove abbiamo spiegato come realizzare una [scheda contribuente](#).



Aggiungiamo anche l'uso dei [task](#), che aiutano nella gestione dei compiti e, sfruttando la parte di messaggistica, consentono una rapida interazione con i vari membri del gruppo, anche se le persone sono distanti km e km.

Conclusioni

Abbiamo visto un esempio di come Confluence sia un valid strumento per aiutarci nel lavoro di tutti i giorni. Da evidenziare è il fatto che, nonostante sia nato per gli informatici, questo strumento ci aiuta notevolmente anche per progetti NON informatici.

Reference

- [Articolo del Blog Atlassian](#)
-

JIRA 7 – JIRA SOFTWARE

Continuiamo l'approfondimento

Proseguiamo il nostro viaggio verso le novità di JIRA. Iniziamo a parlare di JIRA SOFTWARE.



JIRA SOFTWARE

Come anticipato in questo post, abbiamo una nuova specializzazione di JIRA, espressamente dedicata per lo sviluppo software. Mentre abbiamo [JIRA CORE](#), dedicato ai progetti non IT, questa pacchettizzazione è espressamente pensata e rilasciata per il software.



JIRA SOFTWARE non è altro che JIRA, cui siamo già abituati, con già integrate le funzionalità di AGILE programming. Nulla di più. Il tutto espressamente dedicato allo sviluppo

software.



In aggiunta, sono presenti le ultime realizzazioni che, ad oggi, la Atlassian ci ha costantemente abituati, comprensive di migliorie che ci aiutano costantemente nella realizzazione del software. Segnaliamo anche ulteriori miglioramenti sulla grafica e sulla interfaccia web, come mostrato in figura.



Conclusioni

Abbiamo una nuova pacchettizzazione di JIRA che continua ad offrire molto agli sviluppatori. Con già inclusi i tools per lo sviluppo AGILE, ne fa uno tra i tool più potenti ☐