

JIRA CORE – Prova su strada 2

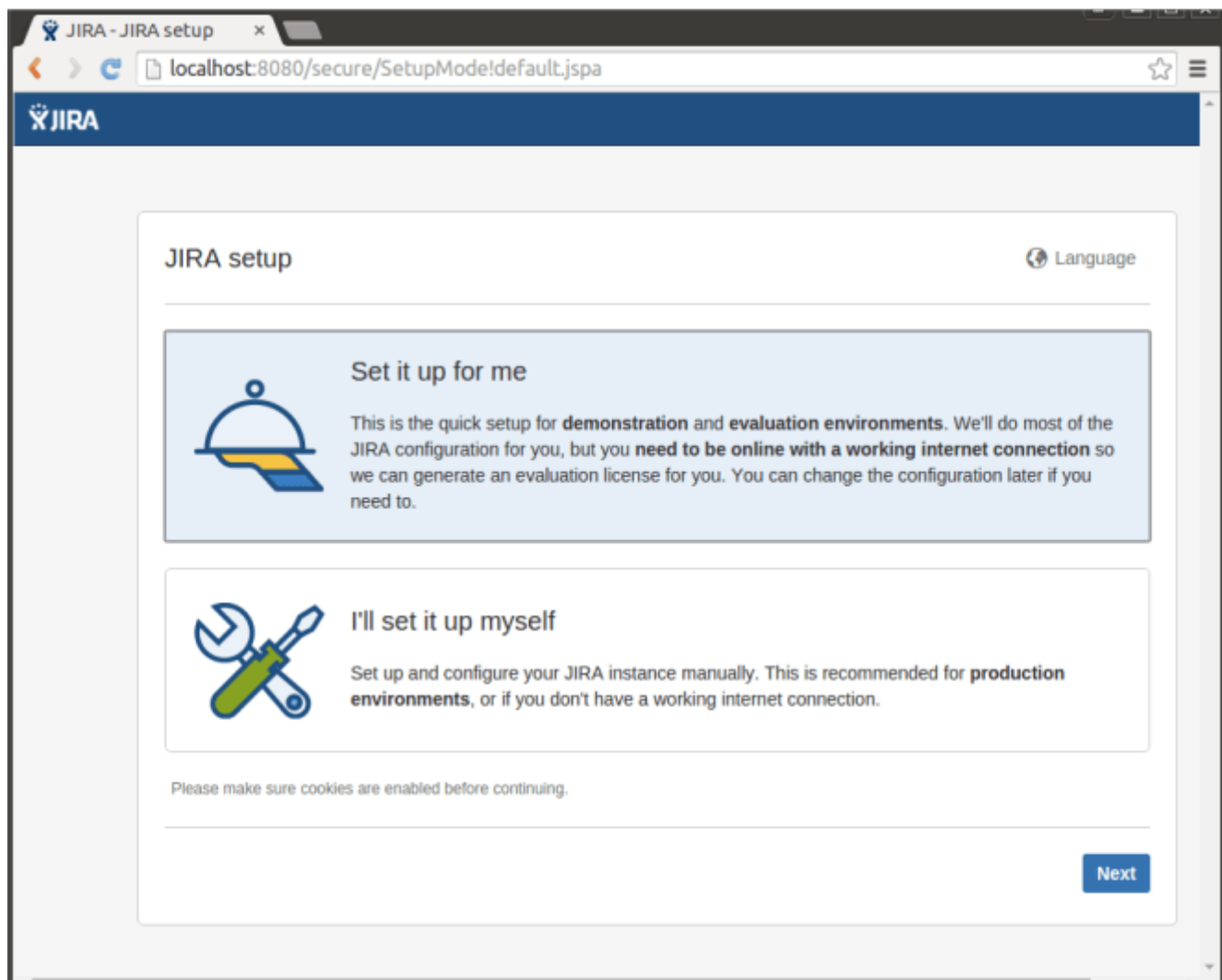
JIRA CORE – proseguiamo la prova su strada

In questo post andremo a proseguire la prova su strada. Passiamo a visionare l'installazione web.



Dove eravamo rimasti

Eravamo rimasti alla pagina iniziale della installazione web, come mostrato nella seguente immagine:



A questo punto, possiamo scegliere due possibili opzioni:

- La prima consiste nel lasciare al setup di fare tutto, fornendo solo alcuni semplici parametri e le indicazioni per staccare la licenza.
- La seconda consiste nel installare manualmente (opzione che andremo a scegliere ☐)

Il motivo della scelta è che in questo post voglio spiegare alcuni punti e, se lasciamo l'installazione fare tutto, allora non riesco a mostrarle ☐

Una piccola precisazione

Prima di procedere, abbiamo installato il driver MySQL sul JIRA al termine della installazione. Prima di procedere si è reso necessario un riavvio del servizio. Tutte le istruzioni sono presenti in questa [pagina](#) della documentazione di JIRA. Occorre riportare nella **<JIRA_HOME>/libs** il driver.

Una volta installato, per eseguire il restart del servizio, collegandosi come root, eseguire i comandi **stop-jira.sh** e **start-jira.sh** presenti nella **<JIRA_HOME>/bin**. Se non viene fatto, il driver non sarà disponibile.

Quindi procedere con la creazione del database vero e proprio, usando i seguenti comandi:

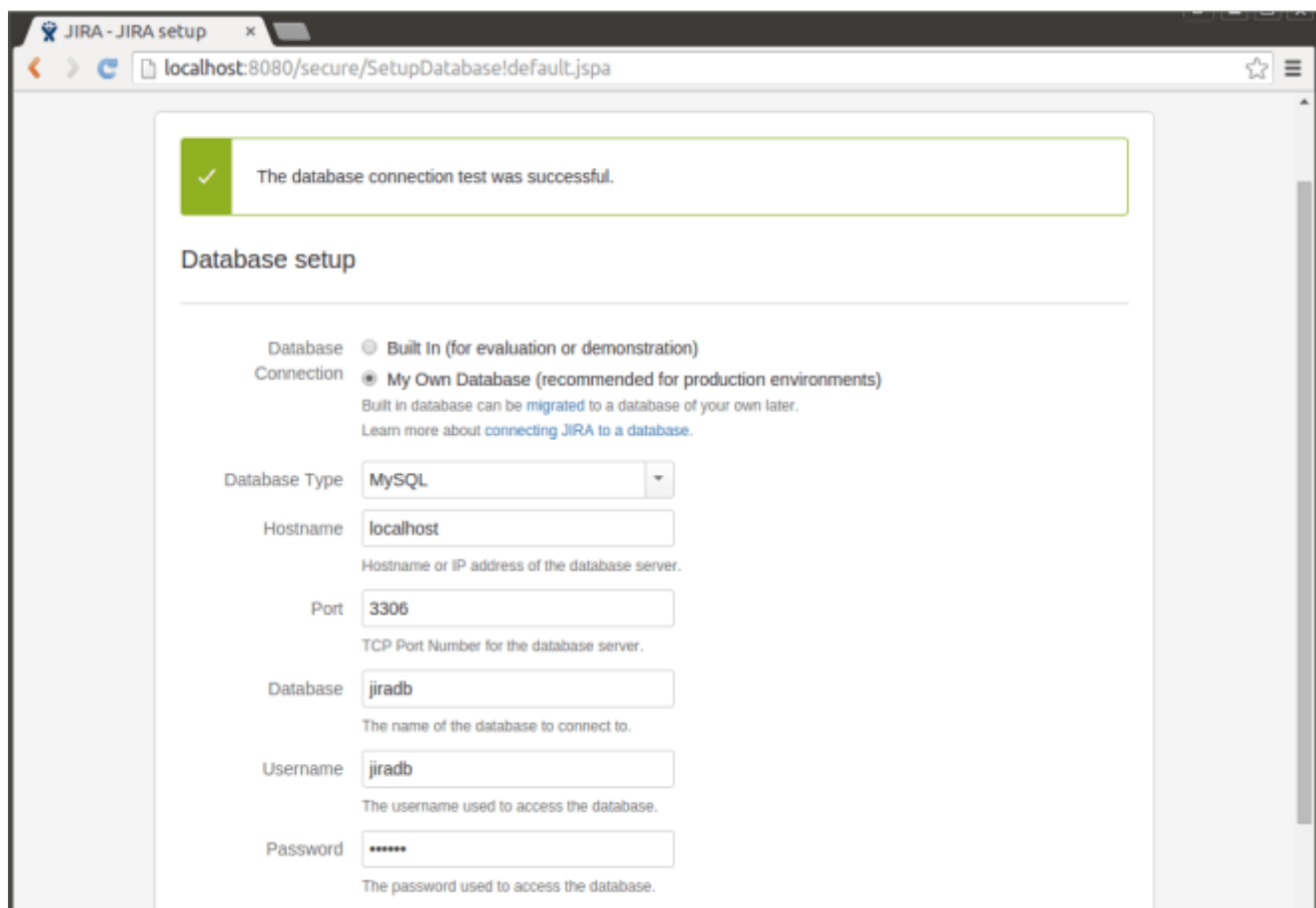
```
▪ CREATE USER '<USERNAME>'@'<JIRA_SERVER_HOSTNAME>'
  IDENTIFIED BY '<PASSWORD>';
▪ CREATE DATABASE jiradb CHARACTER SET utf8 COLLATE
  utf8_bin;
▪ GRANT
  SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,INDEX on
  <JIRADB>.* TO '<USERNAME>'@'<JIRA_SERVER_HOSTNAME>'
  IDENTIFIED BY '<PASSWORD>';
  flush privileges;
```

come potete vedere nella pagina che ho indicato prima :-).

Proseguiamo

Andiamo quindi avanti con il setup. Il primo passo che viene richiesto è quello di scegliere il database. Come abbiamo già scritto, scegliamo MySQL e forniamo tutti i parametri per la

connessione.



The screenshot shows the 'JIRA - JIRA setup' window in a browser. The address bar indicates the URL is `localhost:8080/secure/SetupDatabase!default.jspa`. A green success message at the top states: 'The database connection test was successful.' Below this, the 'Database setup' section contains two radio buttons for 'Database Connection': 'Built In (for evaluation or demonstration)' and 'My Own Database (recommended for production environments)'. The 'My Own Database' option is selected. Below the radio buttons, there are input fields for 'Database Type' (set to 'MySQL'), 'Hostname' (set to 'localhost'), 'Port' (set to '3306'), 'Database' (set to 'jiradb'), 'Username' (set to 'jiradb'), and 'Password' (masked with asterisks). Each input field has a small text description below it.

La pagina ci consente di poter eseguire il test di configurazione, verificando se abbiamo inserito tutti i parametri corretti. Quindi procediamo con la fase di installazione, dove sarà creata l'intera struttura di tutte le tabelle. Questa operazione richiede del tempo. Lasciamo che lavori senza alcuna fretta e attendiamo che si concluda :-).

Al termine, ci sarà proposta la fase successiva, come mostrato dalla figura successiva:

Set up application properties

Existing data? You can [import your data](#) from another installed or hosted JIRA server instead of completing this setup process.

Application Title
The name of this installation.

Mode ☒ Private
Only administrators can create new users.

☐ Public
Anyone can sign up to create issues.

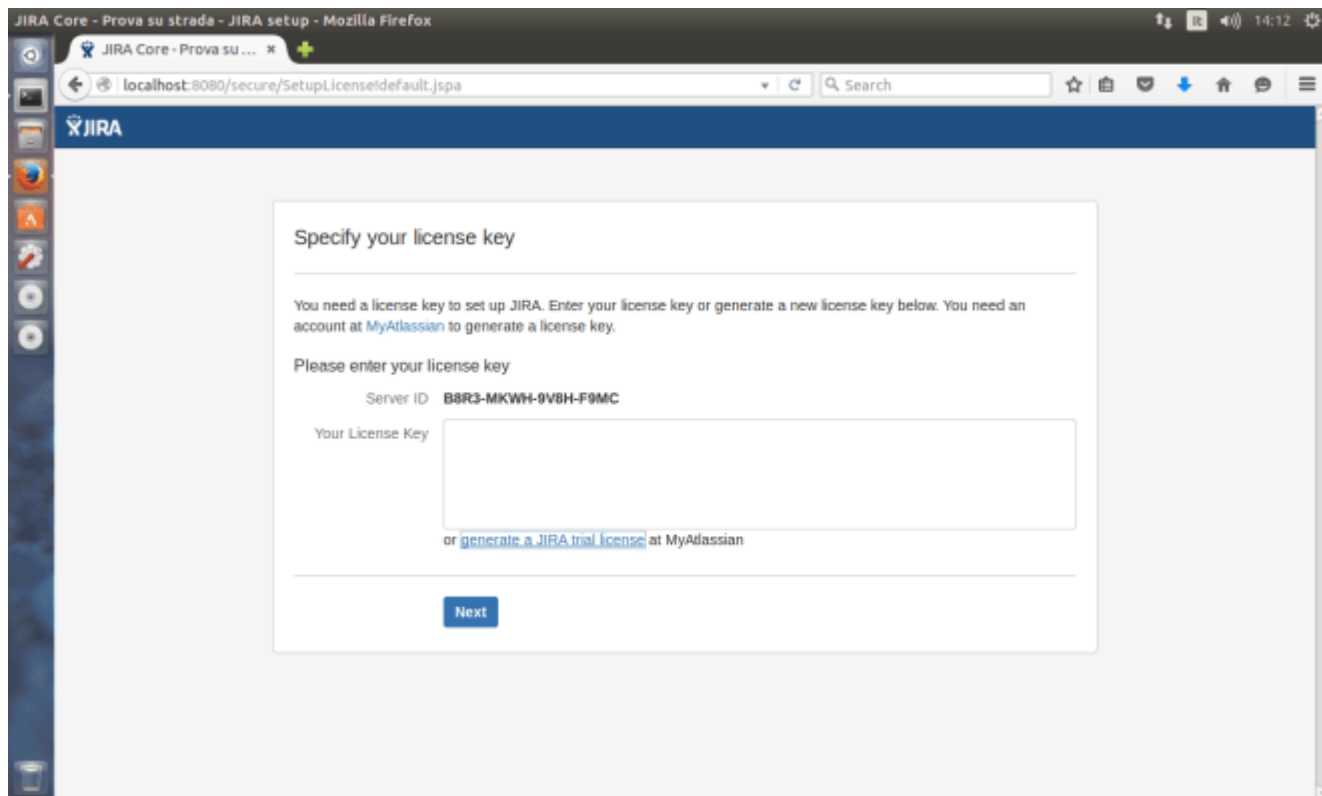
Base URL
The base URL for this installation of JIRA.
All links created will be prefixed by this URL.

[Next](#)

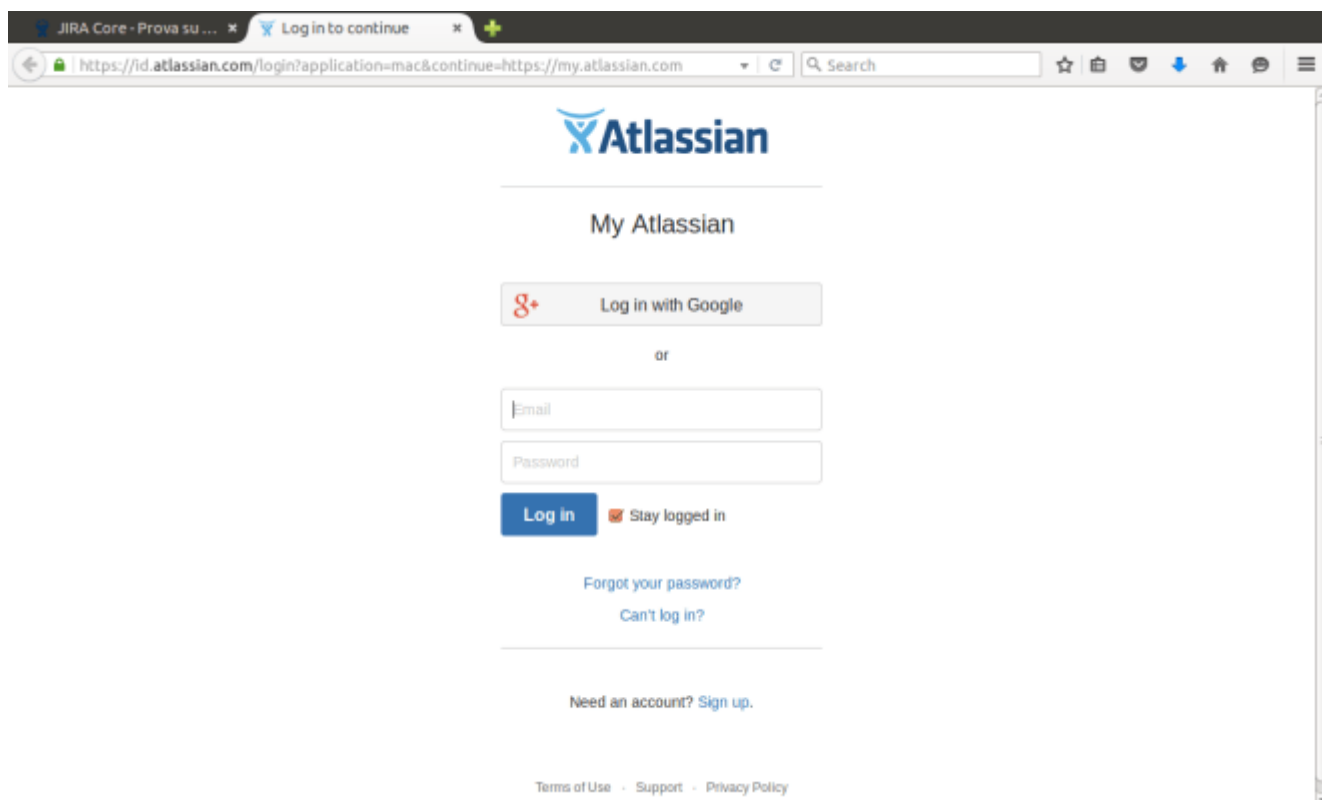
Son richiesti i seguenti parametri:

- Il **titolo** del JIRA CORE: viene richiesto che titolo mostrare quando accediamo al nostro JIRA;
- **Modalità di accesso:**
 - **Pubblico:** Gli utenti si possono registrare in autonomia
 - **Privato:** Solo gli amministratori possono creare nuovi utenti
- Definire la **Base Url**. Si tratta della URL di base che sarà utilizzata per la gestione del nostro JIRA.

Nella fase successiva viene richiesta la **Licence Key**. Possiamo, in automagico, ricavarla usando il link **generate a JIRA Trial licence**, come mostrato dalla figura successiva:



che ci reindirizza al portale **my.atlassian.com**:



attraverso il quale ricavare subito, sfruttando le nostre credenziali, la licenza di uso:

JIRA Core - Prova su ... x Generate Evaluation... x

https://my.atlassian.com/license/evaluation?utm_nooverride=1&ref=prod&product=jir

MyAtlassian Licenses Orders Downloads Buy

New Evaluation License

Product: JIRA Core (Server)

Licensed To: Fabio Genovese

Your instance is: ☒ up and running ☐ not installed yet

Server ID: B8R3-MKWH-9V8H-F9MC

Please note we only provide evaluation support for 90 days per product.

☐ I accept the [Atlassian Customer Agreement](#).

[Generate License](#) [Cancel](#)

Leave Feedback

e quindi riportarla subito nel nostro server e procedere nelle ultime fasi di installazione.

JIRA Core - Prova su ... x JIRA Core - Prova su ... x

localhost:8080/secure/SetupLicense/default.jspx

JIRA

Specify your license key

You need a license key to set up JIRA. Enter your license key or generate a new license key below. You need an account at [MyAtlassian](#) to generate a license key.

Please enter your license key

Server ID: B8R3-MKWH-9V8H-F9MC

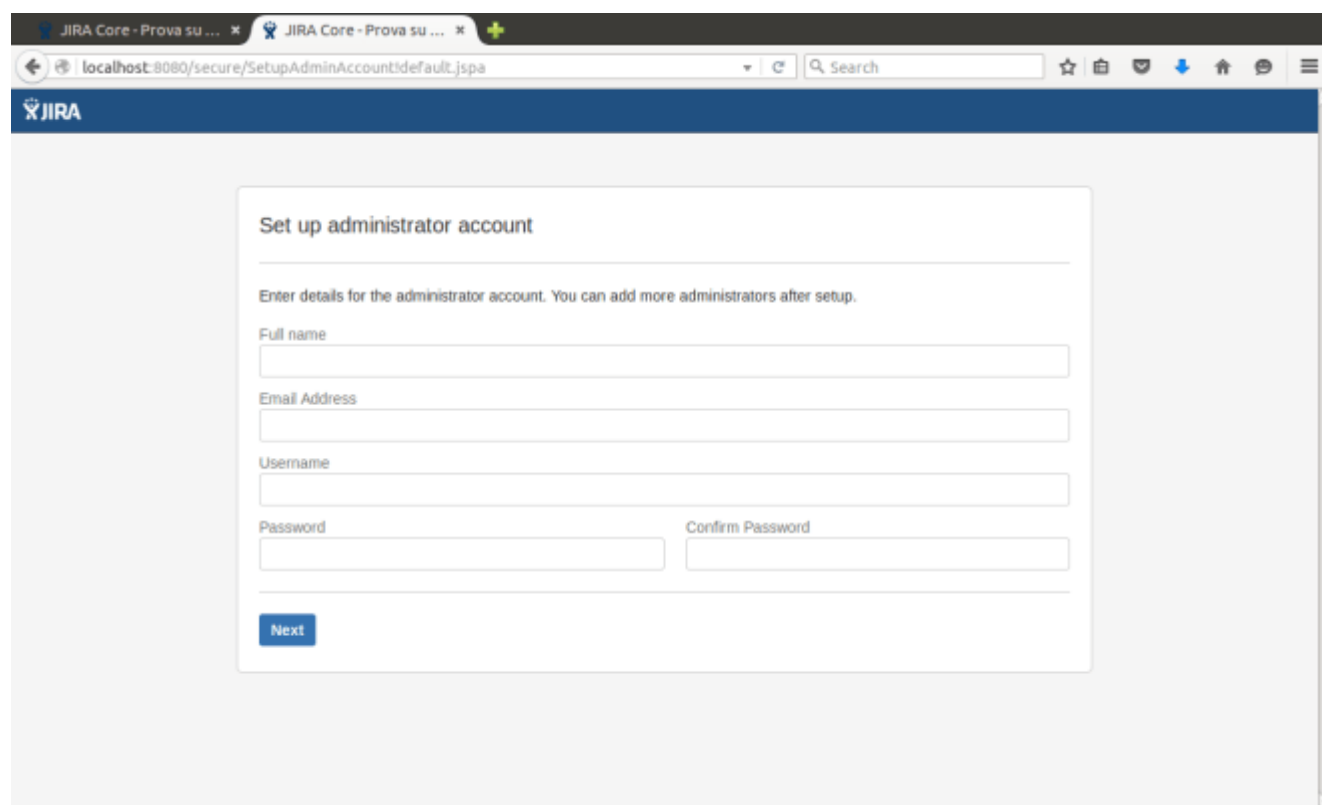
Your License Key: `AAAY000DApWp1u13PukRhu/SF2188cR25ZG2YR2yJVCgRv4R21gRt1u8bvy711a1shuK03mfoqfahc000uP/MAB8B923ruK2vN1v9EDWb8Fv23p13m11q1kxawfY9YCV0EY7/n0nH3+130u0C84j1a0B435AULD1rC1B28pL1R9a70y4U122c7Y9sPn2c78uagfSaFh8113e4Kx23nAp2ny5G/38F4nCl10yefy9Q23vRy5n17aw0Tpxj3178u4ATL3K1T8uKAG8u7ap1PhmcdFmely5q16N0U1u7u499uLJ9qAkg2H403uq0WNTFE7/vet2N5u4v1u417RQ143096082Hed8P6007/3j1v52hefYc9ueh8p1yc2p08WY3860u9u8d1x10rusk11v08R1hehvc23u/A0r9c5PRKwCFFukq+678FpJ7Zf0Q3F7ka373gR6Rv2y0VW0U6/0511-966u0344qau082hd`

[or generate a JIRA trial license at MyAtlassian](#)

[Next](#)

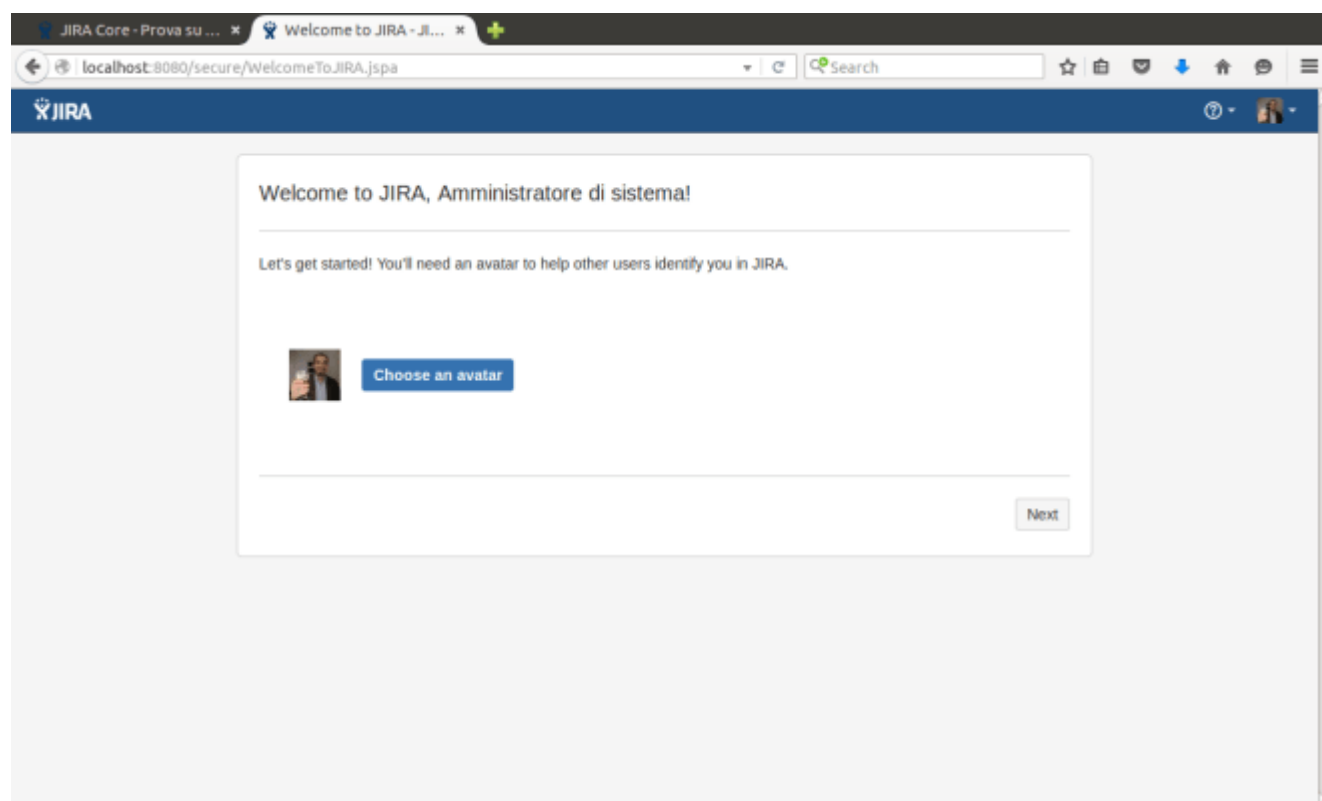
Si passa quindi alla generazione della utenza di amministrazione, molto importante. Si tratta di una utenza

attraverso la quale riusciremo ad eseguire l'amministrazione dell'intero JIRA. Come mostrato dalla seguente immagine:



The screenshot shows a web browser window with the URL `localhost:8080/secure/SetupAdminAccountDefault.jspa`. The page title is "Set up administrator account". Below the title, there is a text input field for "Full name", followed by "Email Address", "Username", "Password", and "Confirm Password" fields. A blue "Next" button is located at the bottom left of the form area.

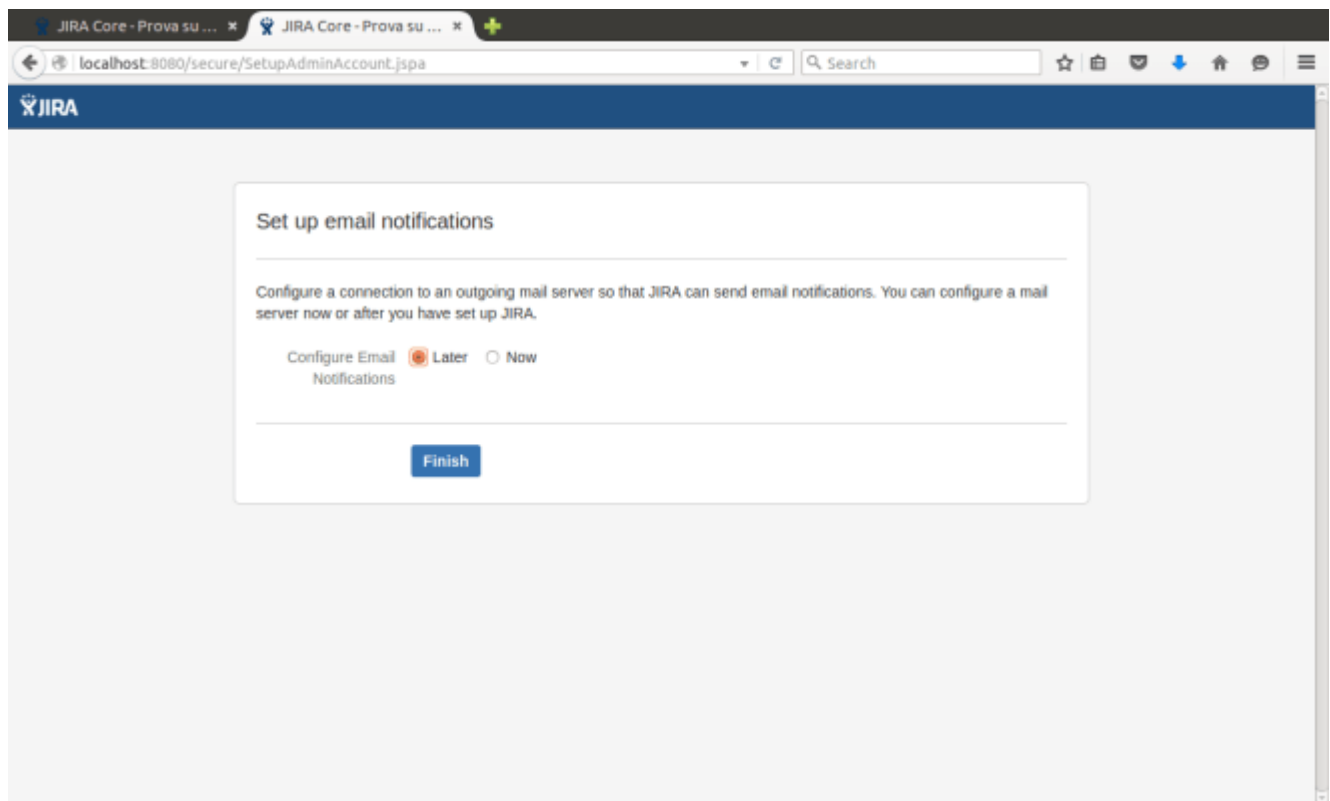
specifichiamo tutti i parametri e andiamo a generare la nostra utenza.



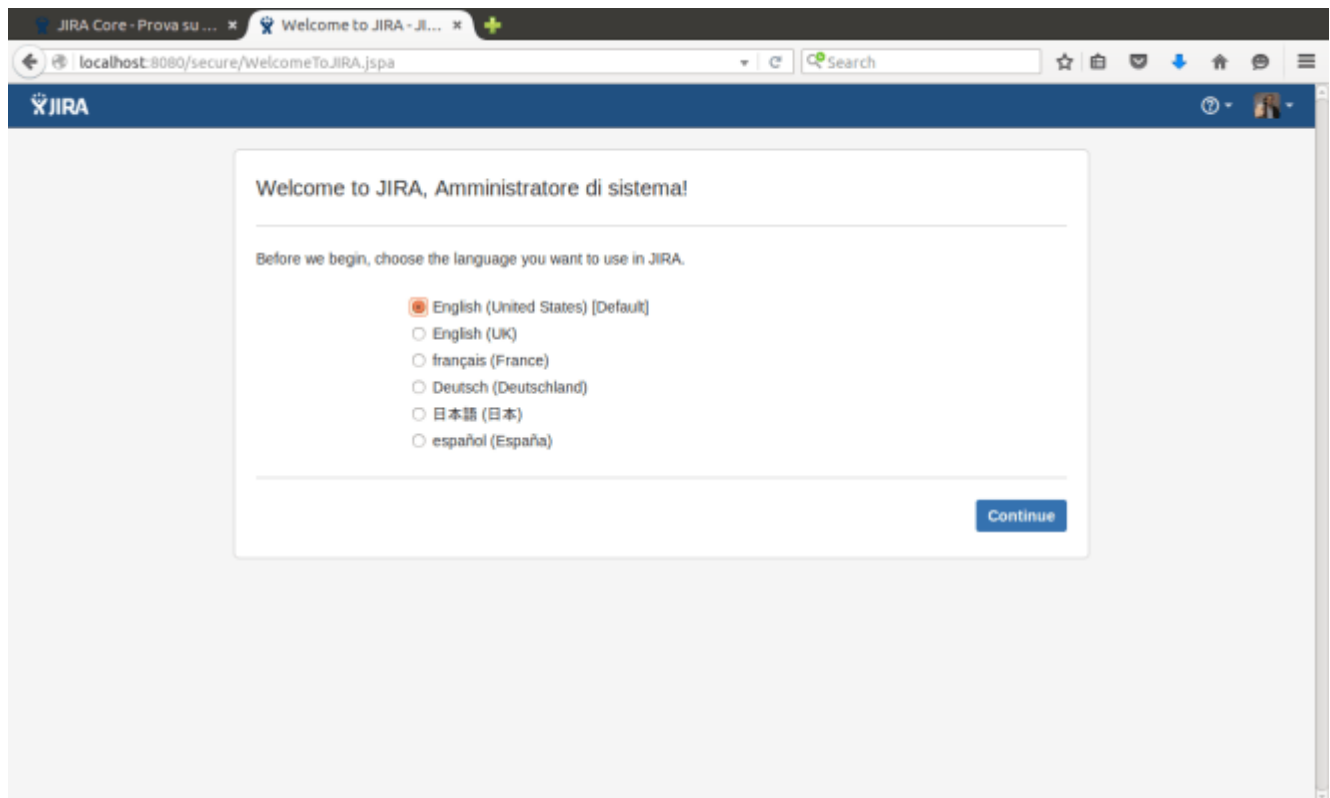
The screenshot shows a web browser window with the URL `localhost:8080/secure/WelcomeToJIRA.jspa`. The page title is "Welcome to JIRA, Amministratore di sistema!". Below the title, there is a text input field for "Full name", followed by "Email Address", "Username", "Password", and "Confirm Password" fields. A blue "Next" button is located at the bottom right of the form area.

Viene quindi chiesto di specificare un avatar. Questo rimarca molto l'aspetto social di JIRA e dei prodotti Atlassian. Come potete vedere, viene specificato l'avatar dell'utente del portale **my.atlassian.com**.

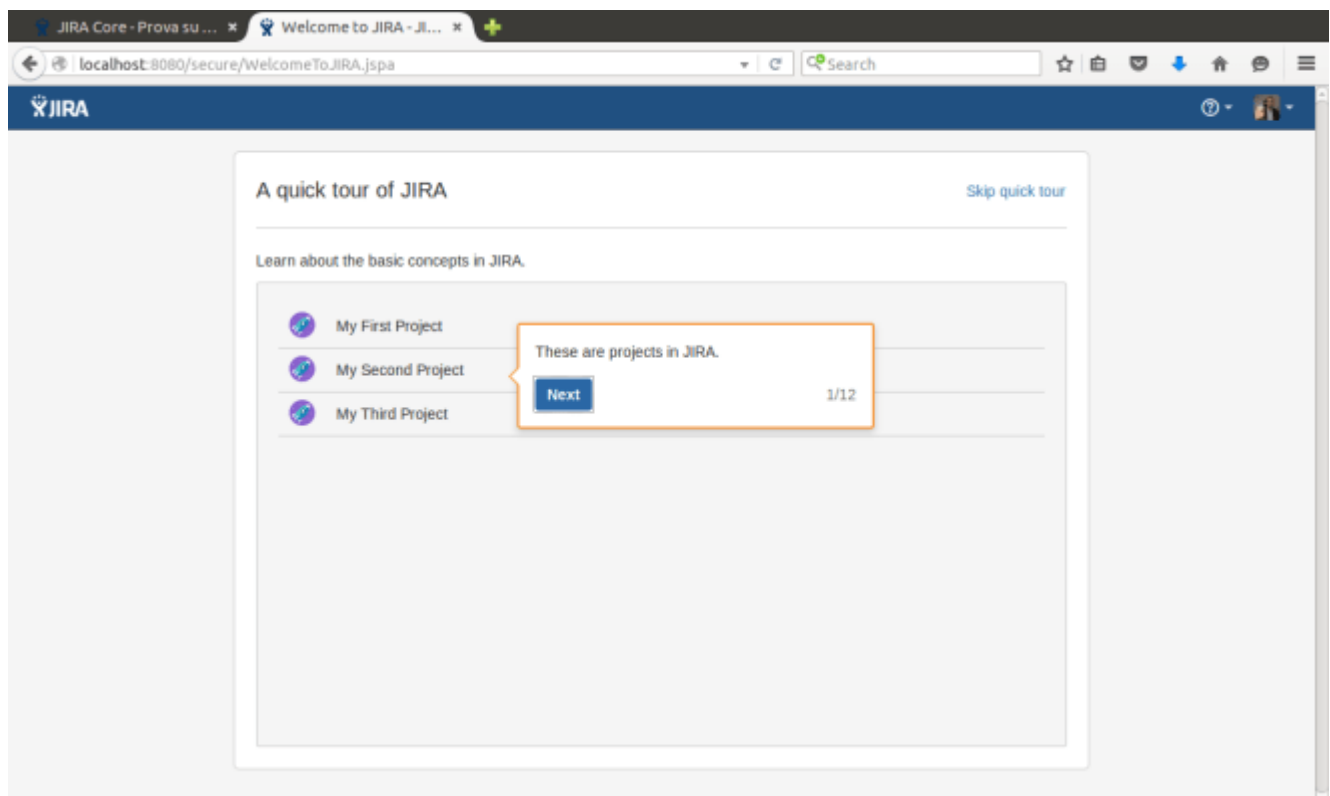
Quindi, viene richiesto di settare i parametri per gestire la notifica delle email.



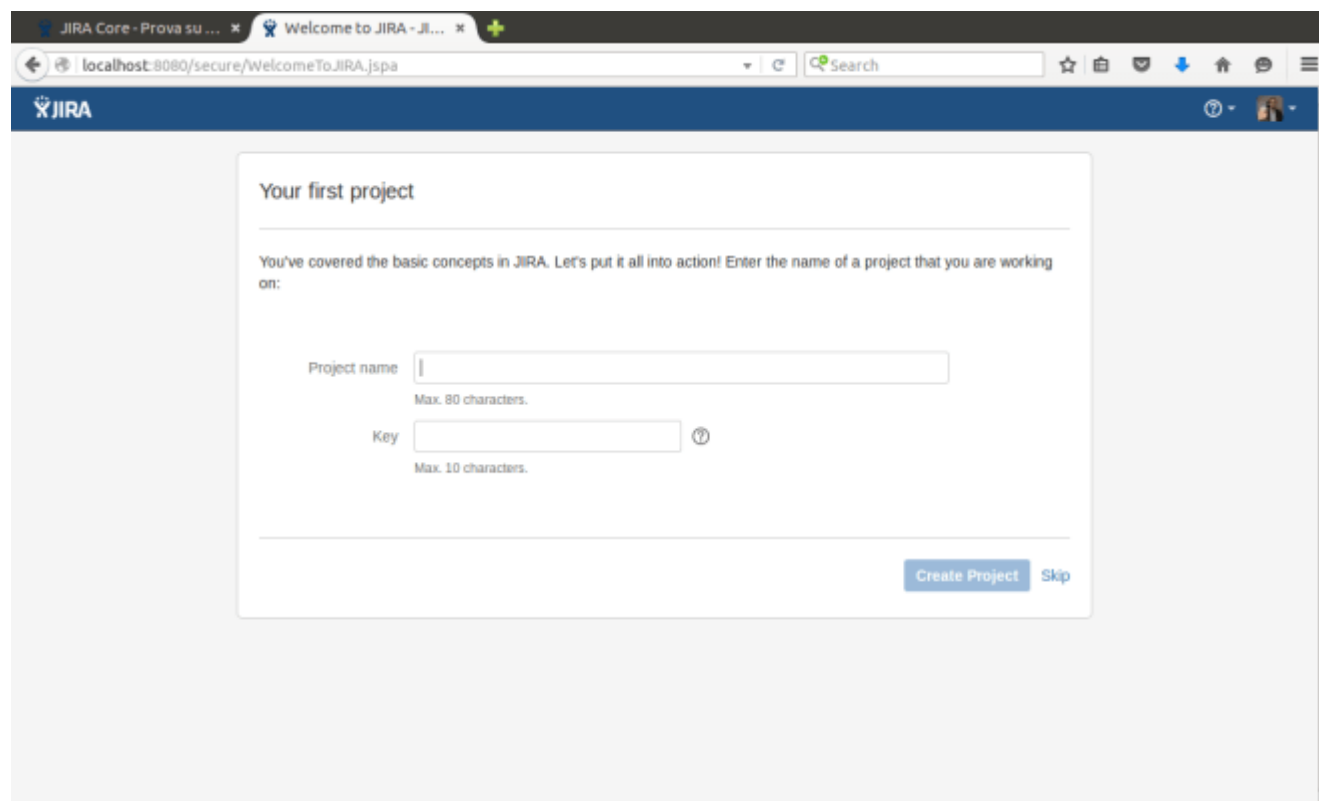
Come possiamo vedere dalla immagine, vediamo che viene suggerito di eseguire dopo questa operazione. Suggerisco di eseguirla in una fase successiva. Quindi passiamo a settare il linguaggio che sarà utilizzato su JIRA, come mostrato nella seguente immagine.



Eseguita tale selezione, l'installazione è formalmente terminata e si passa ad una fase di **quick tour** del prodotto, dove sono spiegate alcune informazioni. Salteremo queste e procederemo oltre.

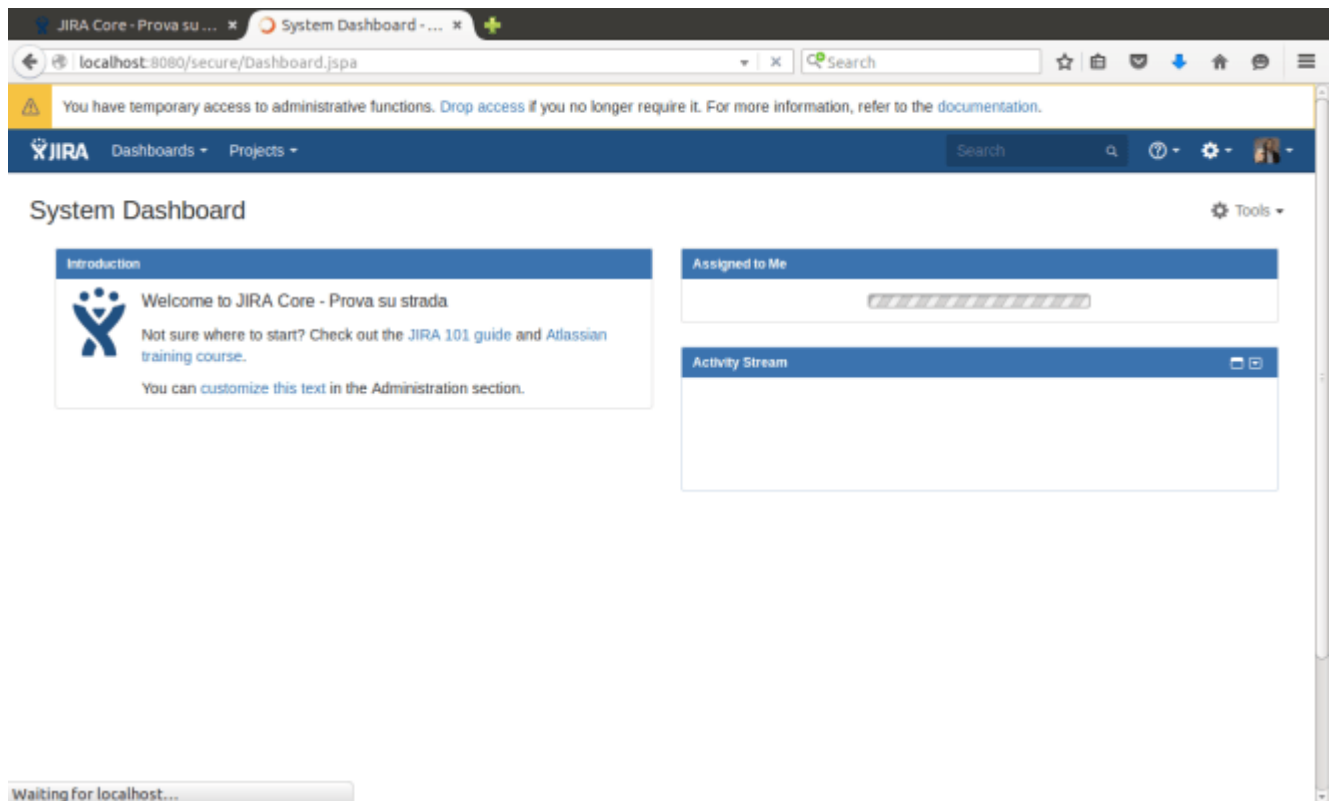


Fatto ciò, possiamo subito iniziare. Dato che si tratta della prima installazione, viene proposto di creare il primo progetto. Anche in questo caso, possiamo eseguire l'operazione in una fase successiva, come mostrato nella seguente immagine:



The screenshot shows a web browser window with the JIRA logo in the top left corner. The main content area is titled "Your first project". Below the title, there is a message: "You've covered the basic concepts in JIRA. Let's put it all into action! Enter the name of a project that you are working on:". There are two input fields: "Project name" with a maximum length of 80 characters, and "Key" with a maximum length of 10 characters. A "Create Project" button and a "Skip" button are located at the bottom right of the form.

Fatto ciò, abbiamo la nostra installazione di JIRA. La Dashboard viene proposta e possiamo iniziare nelle fasi successive:



Conclusioni

Terminiamo qui questo post, chiudendo l'installazione. Proseguiremo nel prossimo post con la prova su strada, verificando le funzionalità, confrontandole con le precedenti versioni e cercando di capire come si comporta questa nuova versione 7 ☐

Traduzione Italiana libro Dan Radigan e Dan Horsfall

Un primo lavoro dell'artigiano

In questo post è adesso disponibile un primo lavoro dell'artigiano. Si tratta della traduzione italiana del libro *Guide to Asset Management* di *Dan Radigan* e di *Dan Horsfall*.



Il libro è reperibile al seguente [link](#) nella versione originale in inglese. In questo post metto a disposizione la versione italiana, dove gli autori spiegano come sia possibile utilizzare JIRA e JIRA Service Desk per realizzare un Asset Management.

Troverete la traduzione qui. [AssetDOC-IT](#).

Una piccola precisazione

Appartengo alla razza umana e non sono perfetto. Di conseguenza, segnalatemi tutti gli errori, anche di battitura, che trovate. Sarà mia cura correggere la traduzione e metterla nuovamente a disposizione.

Exporter per JIRA – Prova su strada

Prova su strada

In questo post andremo a vedere la prova dell'addon descritto in questo [post](#). Questo post sarà propedeutico per un altro, completamente dedicato ad una migrazione vera e propria.

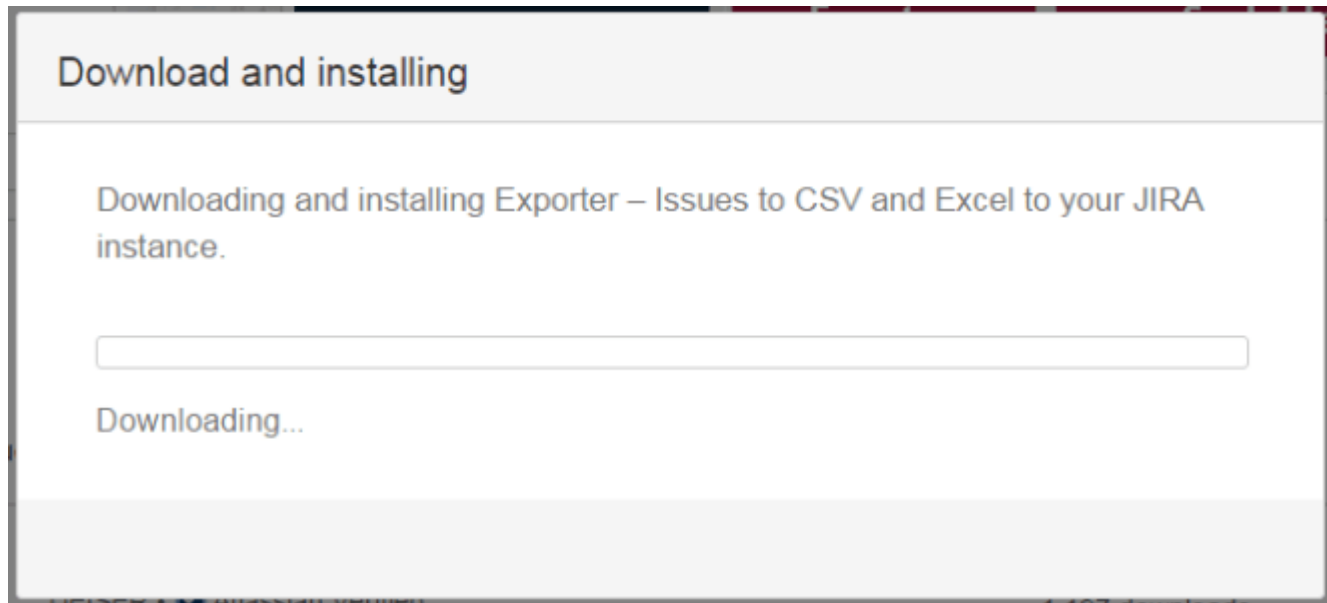


Installiamo

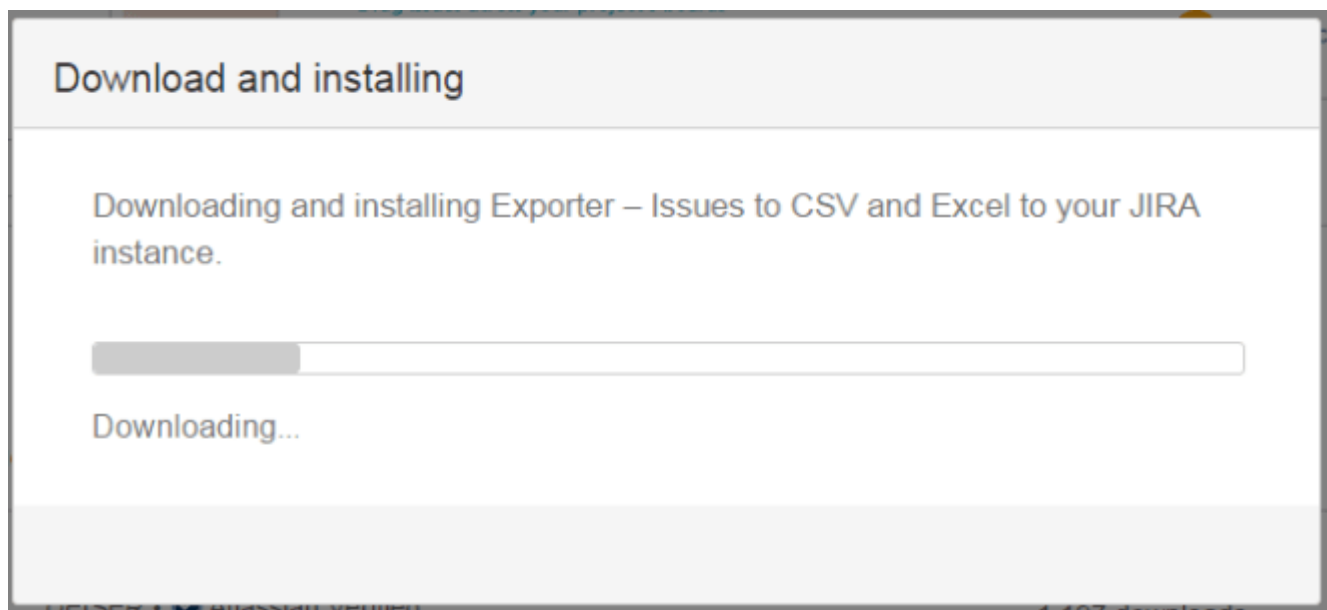
La prima operazione è sempre l'installazione. Guardiamo sempre come viene eseguita, per tenere sempre a mente tutte le operazioni che sono eseguite.

Selezioniamo una licenza Trial e procediamo :-). La prima

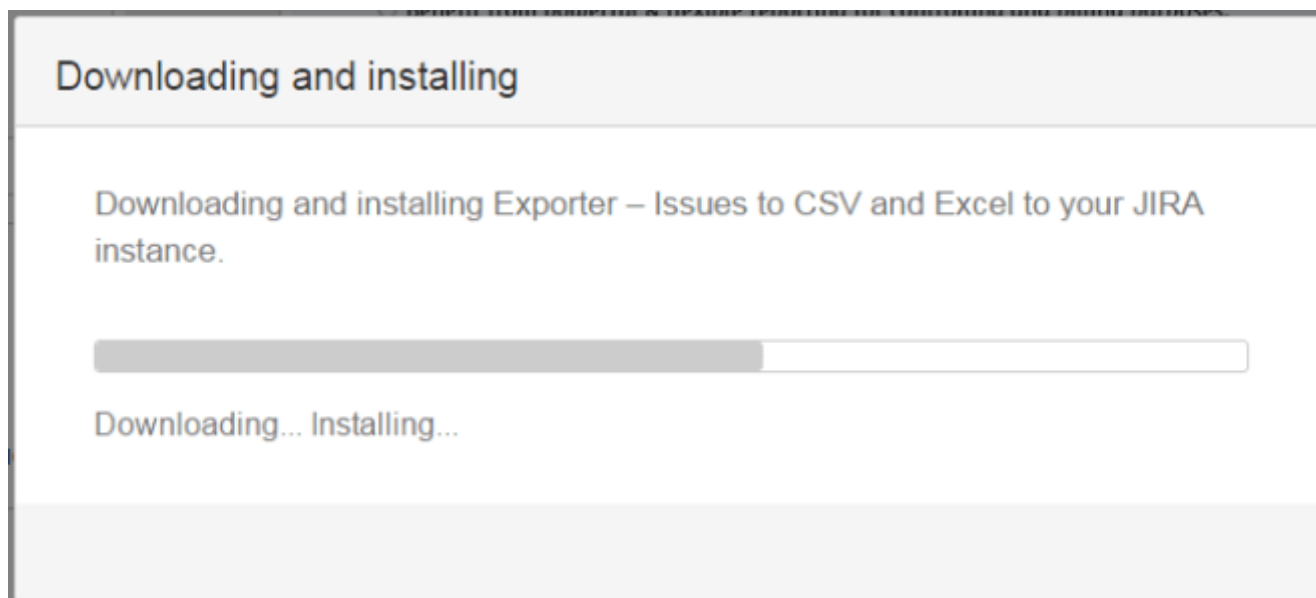
operazione è quella di download



La barra di avanzamento ci avvisa sulle operazioni in corso



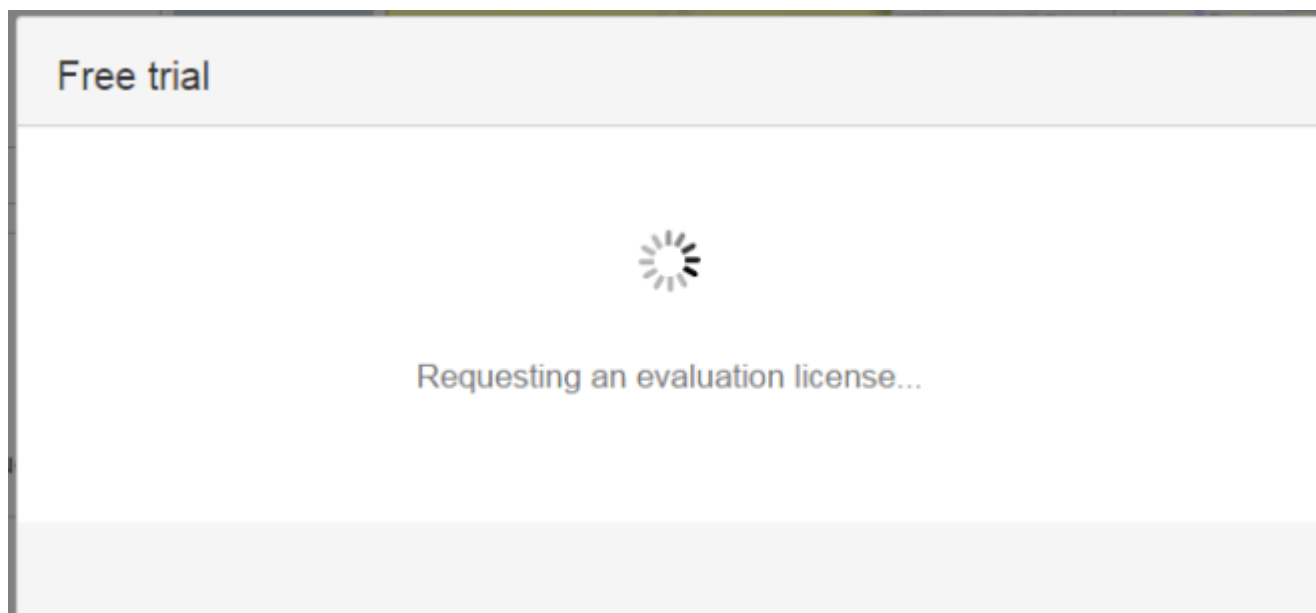
Quindi si procede con la fase di installazione.



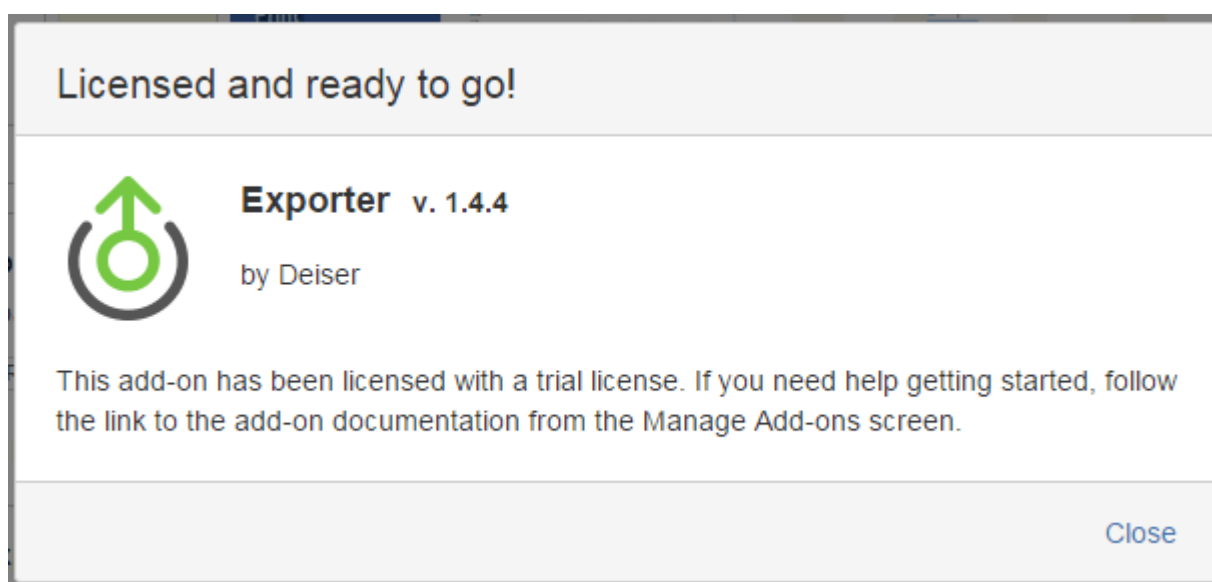
Al termine della installazione, come sempre, si deve procedere al rilascio della licenza ***Trial***. Basta fornire le credenziali e abbiamo il rilascio della licenza.

A screenshot of the Atlassian ID login form. The header says "Atlassian ID". Below it, the text reads "Log in with your Atlassian ID or MyAtlassian account to obtain an evaluation license key." There are two input fields: "Email *" and "Password *". To the right of the password field is a link that says "Lost password". Below the password field is a checkbox labeled "Remember me". At the bottom right, there are two buttons: "Log in" and "Cancel".

Si comunica con il sito della Atlassian e...



... la licenza è pronta.



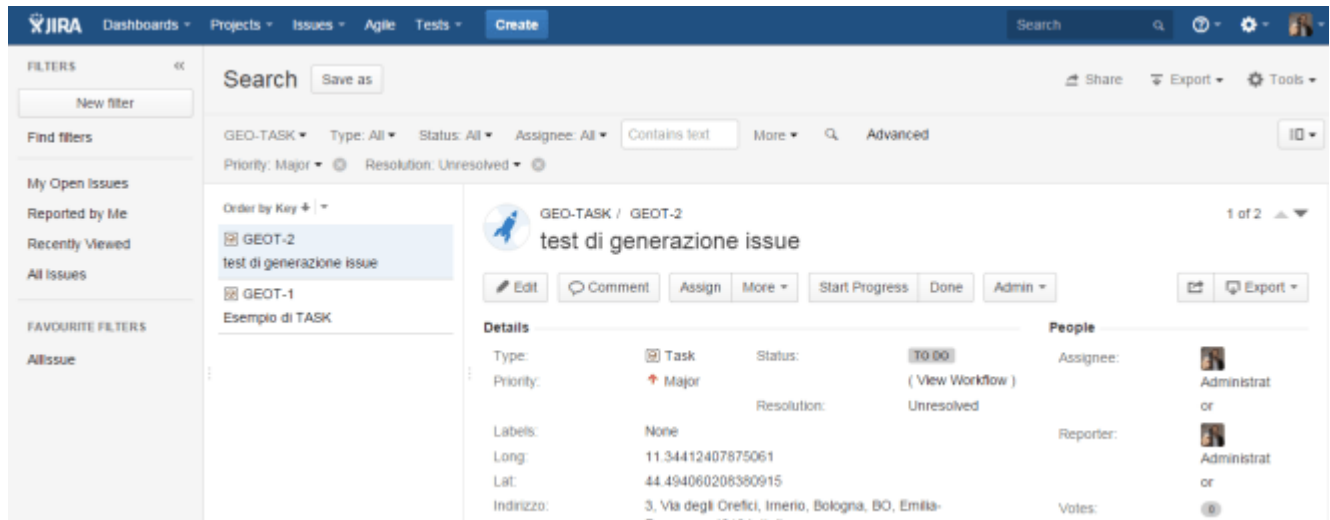
Test Test Test Test Test

Procediamo con la nostra prova. La prima cosa da fare è sempre controllare la configurazione. Deve diventare una abitudine, un modo di procedere semplice che ci aiuta sempre ogni giorno.

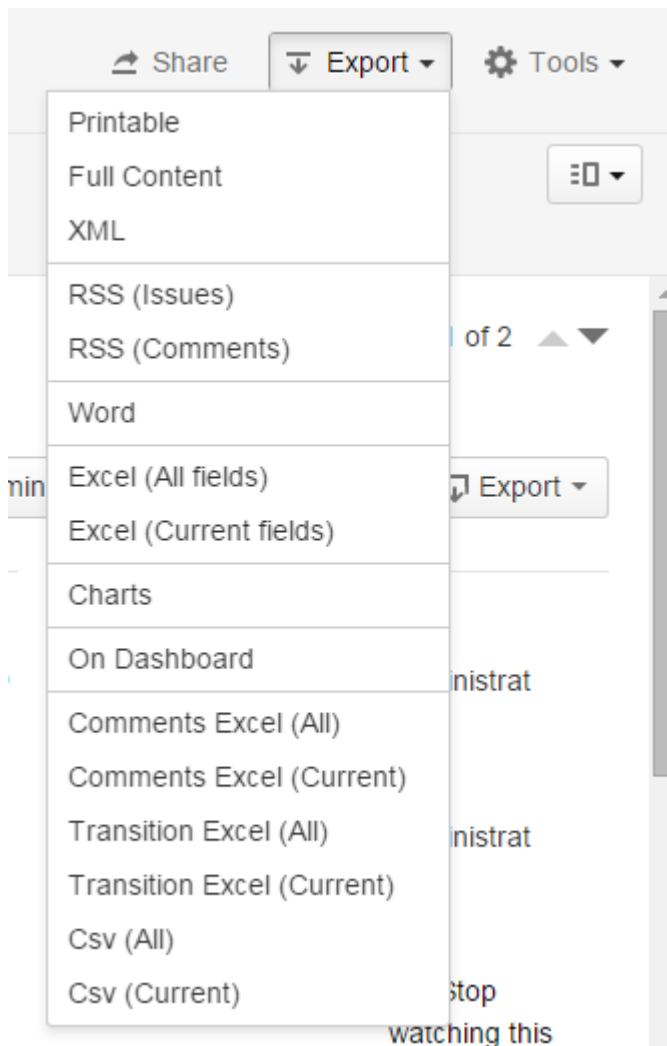
Nel nostro caso caso, non abbiamo alcuna sezione di

configurazione. Questo lo rende molto più semplice.

Passiamo quindi subito all'utilizzo, andando subito al sodo. Per poter usare il tutto, ci posizioniamo sulla schermata delle Issue, come indicato nella figura successiva:



Se andiamo a vedere le opzioni **Export** (si intende il menù Export in alto, a fianco di **Share** e **Tools**).



Le opzioni che sono messe a disposizione sono le seguenti:

- Estrazione commenti, in formato Excel
- Estrazione transazioni, in formato Excel
- Estrazione di TUTTO, in formato CVS.

Possiamo sia eseguire l'estrazione di tutto che del singolo al momento evidenziato.

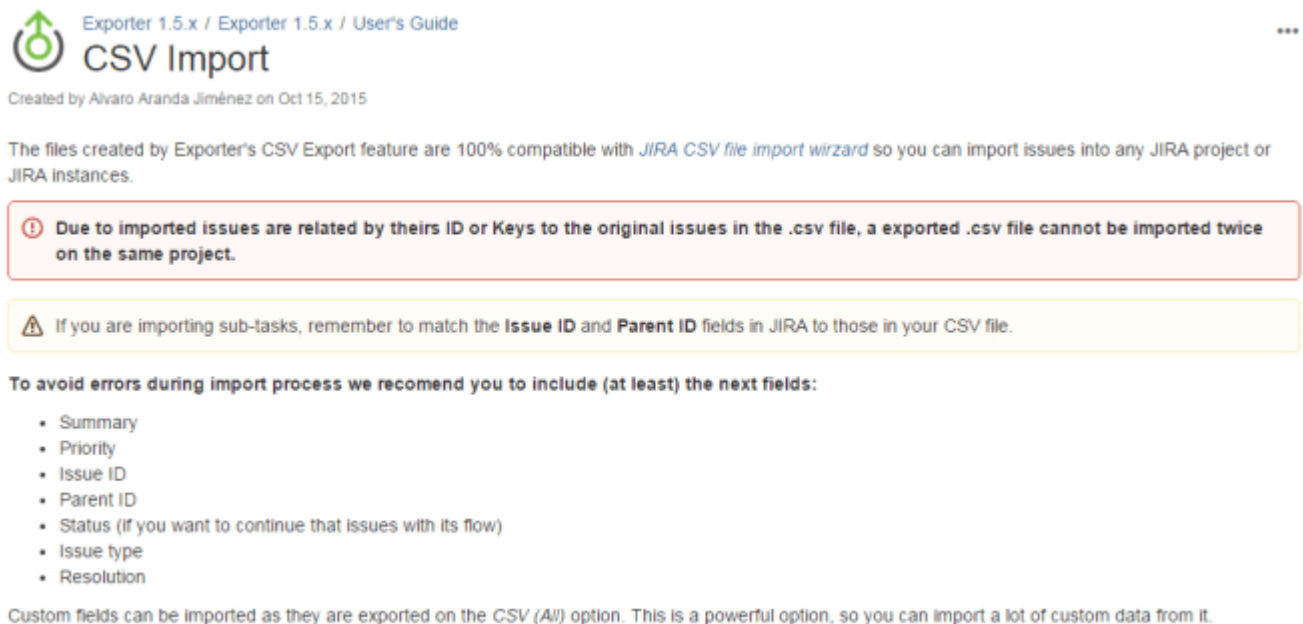
Applico il tutto ad un progetto di test, che ho utilizzato quando ho trattato l'addon [MapIt](#). I risultati sono i seguenti:

Su [Transazioni](#) e su [Commenti](#) trovate gli esempi, mentre su [ALL](#) trovate l'esempio (in formato Excel) di quanto esportato.

Come si può osservare, quello che otteniamo è un file che possiamo poi usare per i nostri scopi. In aggiunta possiamo

anche ... importare :-). Vediamo come reperire l'informazione.

Ma prima, alcune considerazioni prima di procedere:



The screenshot shows the 'CSV Import' section of the 'Exporter 1.5.x / Exporter 1.5.x / User's Guide'. It includes a warning box stating that a CSV file cannot be imported twice on the same project due to ID or Key conflicts. Another note mentions matching 'Issue ID' and 'Parent ID' fields for sub-tasks. A list of recommended fields for import is provided, including Summary, Priority, Issue ID, Parent ID, Status, Issue type, and Resolution. A final note mentions that custom fields can be imported using the 'CSV (All)' option.

Exporter 1.5.x / Exporter 1.5.x / User's Guide
Created by Alvaro Aranda Jiménez on Oct 15, 2015

The files created by Exporter's CSV Export feature are 100% compatible with *JIRA CSV file import wizard* so you can import issues into any JIRA project or JIRA instances.

⚠ Due to imported issues are related by their ID or Keys to the original issues in the .csv file, a exported .csv file cannot be imported twice on the same project.

⚠ If you are importing sub-tasks, remember to match the **Issue ID and **Parent ID** fields in JIRA to those in your CSV file.**

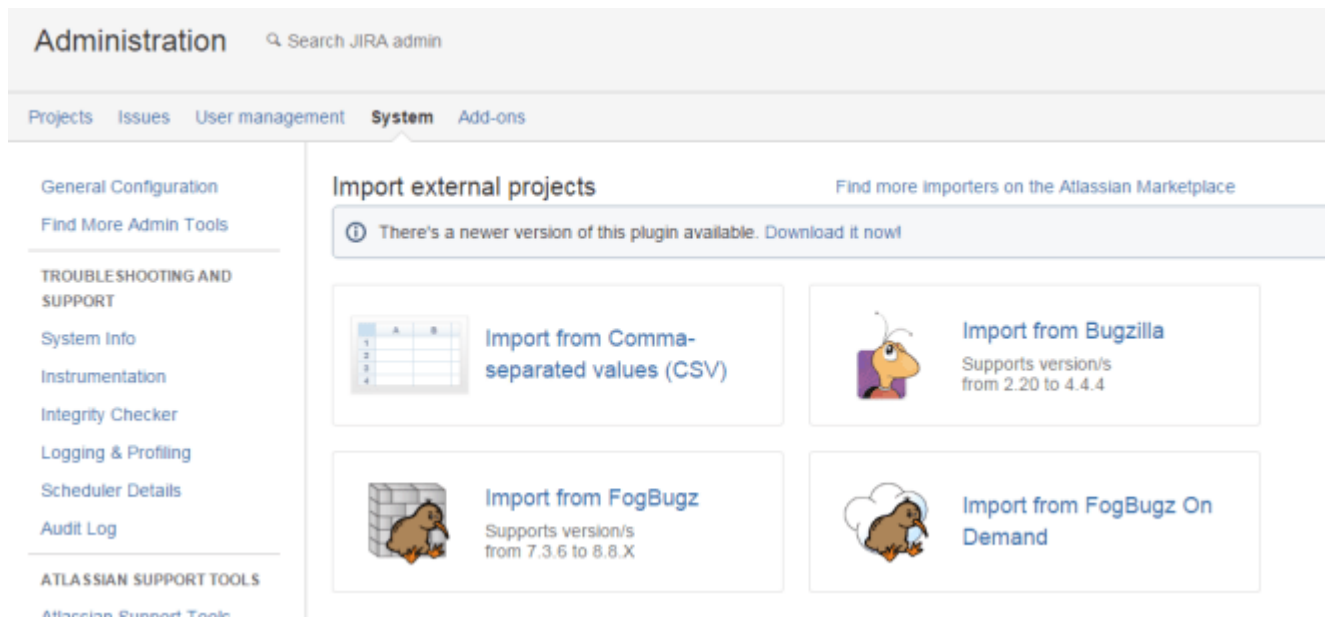
To avoid errors during import process we recomend you to include (at least) the next fields:

- Summary
- Priority
- Issue ID
- Parent ID
- Status (if you want to continue that issues with its flow)
- Issue type
- Resolution

Custom fields can be imported as they are exported on the *CSV (All)* option. This is a powerful option, so you can import a lot of custom data from it.

- Non cerchiamo di importare due volte lo stesso CSV. Ricordarsi sempre che gli ID non possono essere duplicati e, come conseguenza, prestare massima attenzione.
- Includere i campi sopra elencati, per evitare gli errori in fase di importazione.

Come indicato nella immagine precedente, possiamo sfruttare la funzione standard ***JIRA CSV file import wizard***, per eseguire l'importazione. Questo perché l'addon mette a disposizione un formato completamente compatibile con la [funzionalità standard](#).



Seguendo sempre le indicazioni della documentazione.

Conclusioni

Abbiamo visto quali funzionalità offre questo addon. Possiamo fare tantissime cose interessanti. Nei prossimi post andremo a sfruttare questa funzionalità per arrivare a realizzare una migrazione verso JIRA 7.

Precisazioni

Per eseguire il test, ho utilizzato JIRA versione 6.3.11#6341. Di conseguenza potremmo avere delle schermate video che potrebbero risultare strane. Semplicemente si tratta di una versione un più vecchia :-), che mi serve anche per dei test di migrazione.

JQL – Estendiamolo

Estendiamo il JQL

In questo post andremo a dare una occhiata su alcuni Addon per JIRA, per capire come possiamo estendere le funzionalità di JQL.



JQL Tricks Plugin

Si tratta di un addon a pagamento, disponibile per le installazioni server, (al momento in cui viene redatto il post), che estende JQL con nuove funzioni di ricerca, che ci aiutano notevolmente.



Abbiamo la possibilità di poter restringere/autorizzare l'uso di tali funzioni in base ai gruppi di appartenenza degli utenti o in base ai progetti.



In questo modo si possono avere funzioni opportune solo per determinati progetti.

Craftware Search Attachments for JIRA

Si tratta di un addon a pagamento, disponibile per le installazioni server, (al momento in cui viene redatto il post), che estende JQL dando la possibilità di poter eseguire delle ricerche del testo contenuto negli allegati alla Issue, sul nome degli allegati o sul tipo.



Un apposito motore si preoccupa di indicizzare gli allegati,
in modo da poter facilitare la ricerca



permettendo di poter eseguire le ricerche in maniera opportuna



Craftware Search Linked Issues for JIRA

Questo addon, gratuito (al momento in cui viene scritto il post), consente di poter estendere la ricerca permettendo di identificare i subtask in maniera molto più agevole.



L'addon consente di poter impostare delle condizioni in modo da meglio rintracciare le issue che ci interessano



permettendo di poter identificare le issue che bloccano altre issue particolari.



Conclusioni

Abbiamo visto un insieme di addon il cui obiettivo è quello di estendere JQL al fine di aiutarci. Abbiamo visto come tali strumenti ci possono aiutare nella vita lavorativa, per migliorare le ricerche. Nei prossimi post andremo ad esaminarli in dettaglio.

Reference

[Articolo del blog di Atlassian](#)

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.

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

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.

5 JIRA links

JIRA links

Epics

Mobile

☐ 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

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.



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](#)

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 ☐



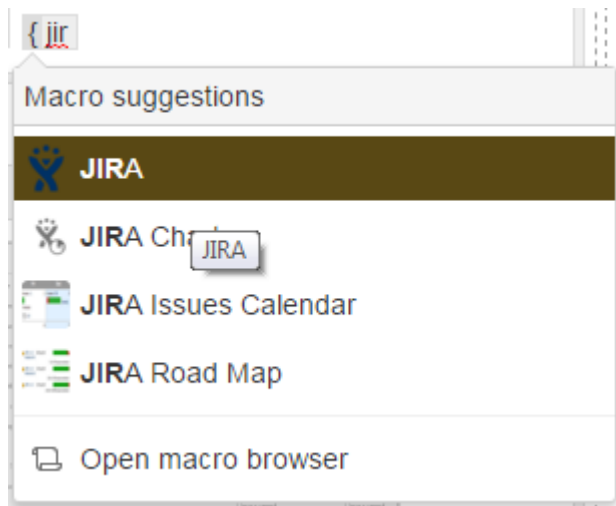
Andiamo al dunque

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:

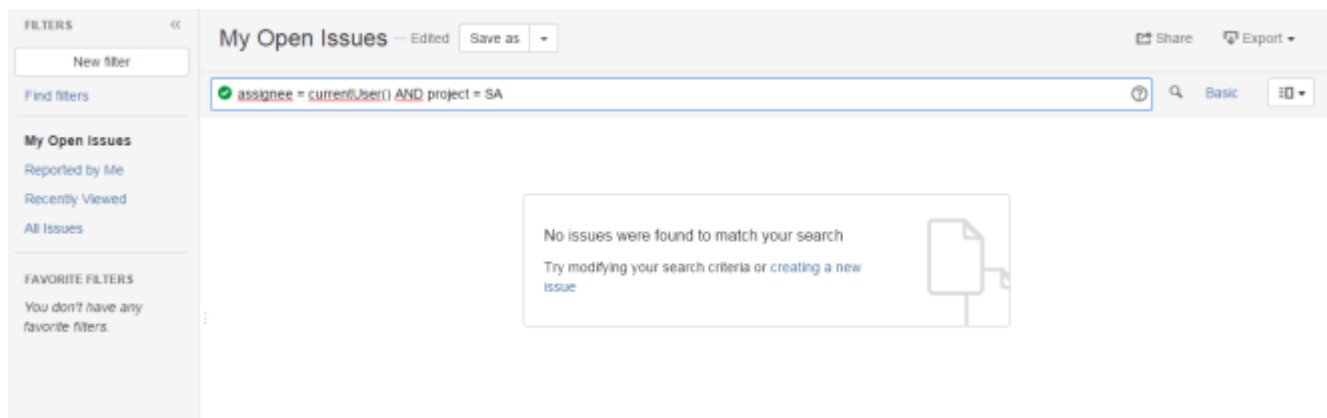


Quindi, sfruttiamo il JQL per andare a selezionare le Issue o la Issue che ci interessa.

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

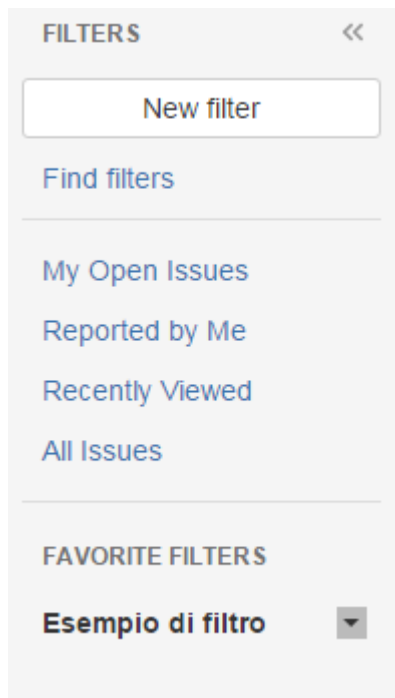
Su JIRA semplicemente andiamo ad accedere alla sezione delle Issue, dove andiamo ad impostare sia i filtri che ad eseguire tutte le interrogazioni che ci servono.



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):

A screenshot of the 'Save Filter' dialog box in JIRA. The dialog has a title bar 'Save Filter'. Inside, there is a label 'Filter Name *' followed by a text input field containing 'Esempio di filtro'. Below the input field is a hint text 'Enter a name for this Filter'. At the bottom right of the dialog 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.

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

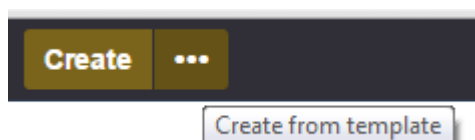
meglio e di fornire quanti più dettagli possibili.



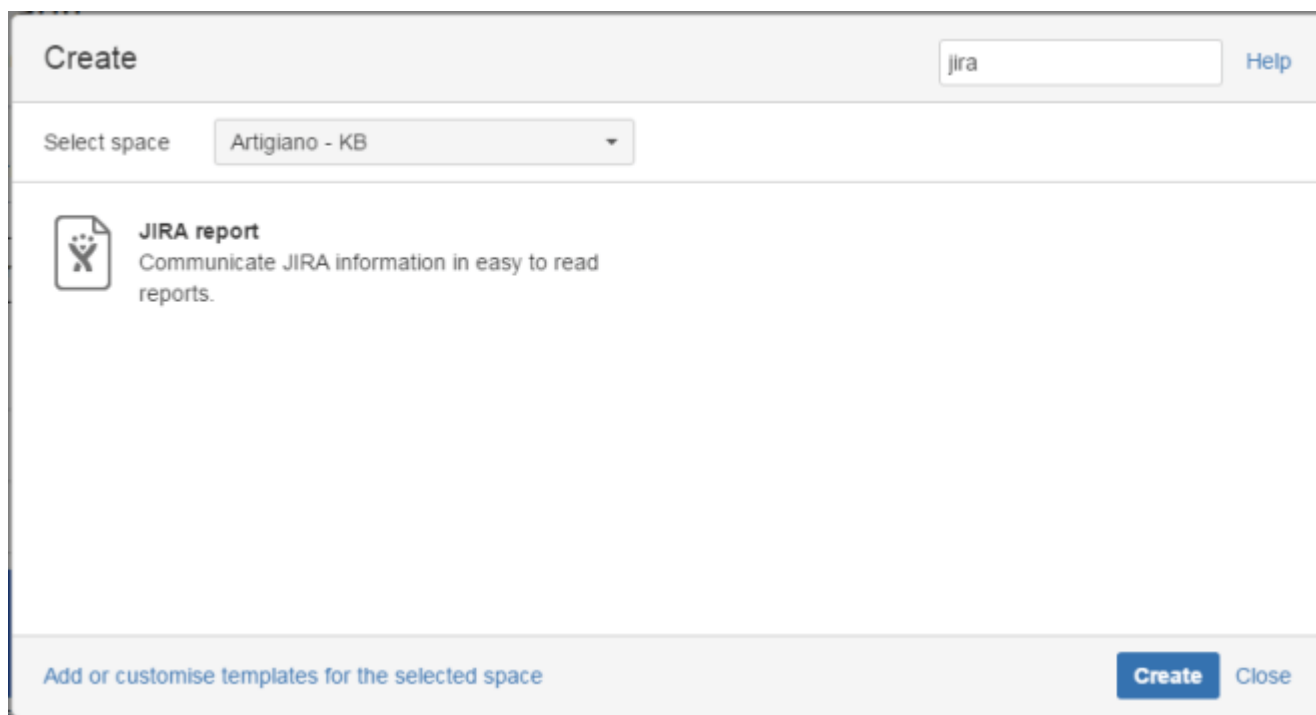
Procediamo...

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**:

Select report type

Change log
Create a report with a list of JIRA issues for a project, release or specific query.

Status report
Create a report with charts to communicate the status of your release with stakeholders.

About JIRA reports

Communicate JIRA information in easy to read reports.

Back Next Close

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 TAG

Fix version(s)

[Switch to advanced](#)

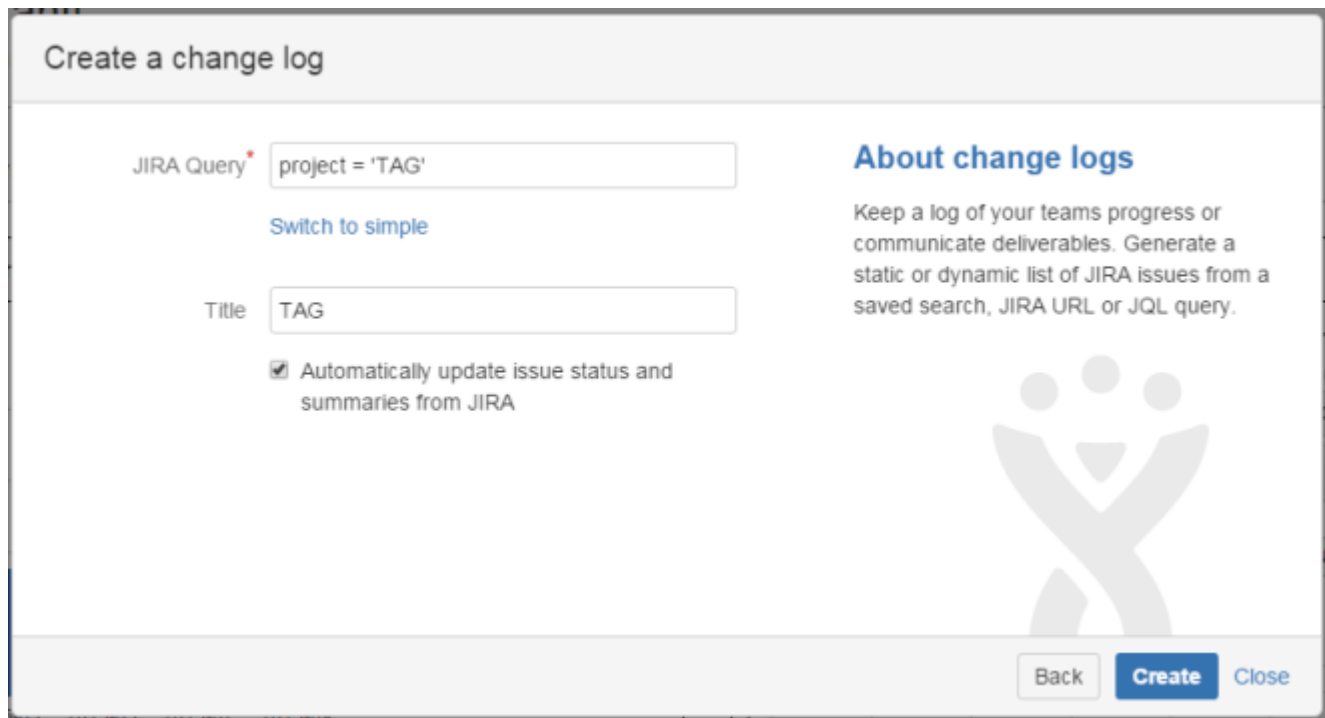
Title TAG

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.

Back Create Close

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.



[Back](#) [Create](#) [Close](#)

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.

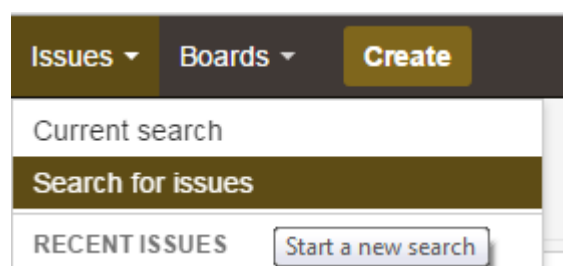


Che cosa è?

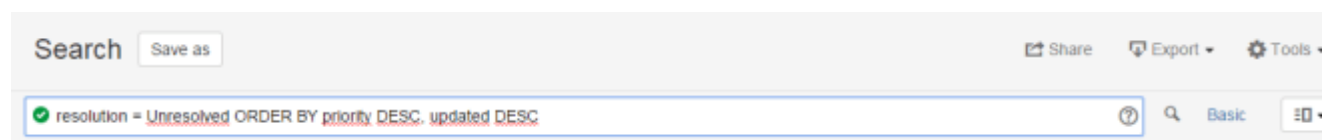
La prima cosa che facciamo è rispondere ad una semplice domanda: Che cosa è JQL? Come indicato nel titolo, JQL sta per JIRA Query Language ed è un linguaggio simil-sql che la Atlassian mette a disposizione per ricercare issue su JIRA.

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 priorita ed aggiornamento.

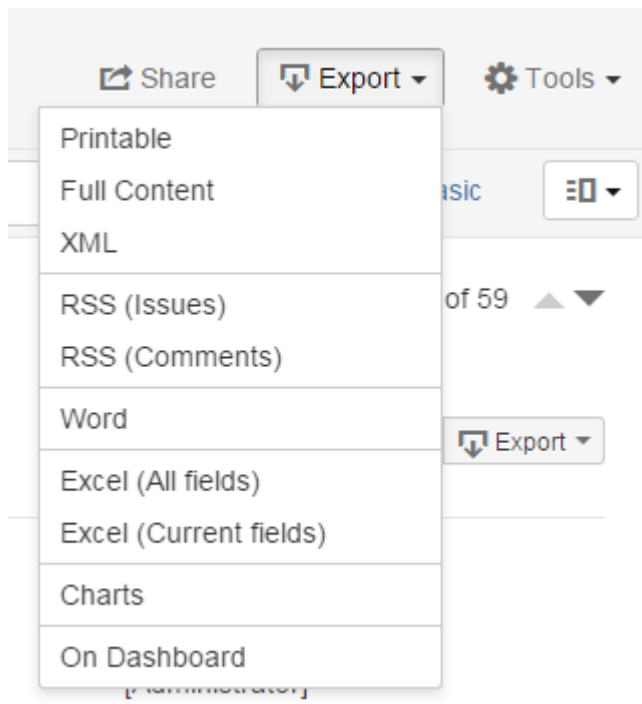
Notiamo anche che JIRA stesso ci indica se la query JQL è corretta o meno



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'obiettivo 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](#)

ConfForms per Confluence – Ultime novità

News news news

Questo post è dedicato alle ultime novità dell'Addon [ConfiForms](#) per Confluence. Avevamo già parlato di questo addon nel post, dove avevamo descritto come aggiungere delle Form a Confluence.



Andiamo in dettaglio

Vediamo nel dettaglio quali sono le novità che annunciano dalla [Vertuna](#) :-).

- ***Gestione di più di 35 tipi di campi custom.*** Questo sicuramente aiuta nella gestione di form abbastanza complesse e consente di poter meglio configurare le

ATLASSIAN

form

▪ *Console di amministrazione potenziata*



ATLASSIAN

▪ *Gestione di form Master/Detail e filtri dinamici*

ATLASSIAN

▪ *Gestione di Tabbed form*

ATLASSIAN

- **Gestione di Conditional Fields**
- **CalendarView**, che consente di mostrare i dati della form in un Calendario

ATLASSIAN

- **Integrazione con JIRA**: Possibilità di poter creare delle JIRA ISSUE ☐
- **Inviare email e notifiche** via confiForms ed usare delle form per creare delle pagine
- e tanto altro ancora

Conclusioni

Abbiamo visto solo la punta di un iceberg. Nei prossimi post andremo a testare sul campo le potenzialità di queste nuove features.