

# FishEye – 2

## *Approfondiamo Fisheye*

In questo post andremo a vedere il funzionamento di Fisheye, in modo da gestire un semplice repository di codice sorgente, proseguendo quanto già discusso nel precedente [post](#).



## *Di che cosa abbiamo bisogno?*

Sostanzialmente, abbiamo bisogno delle seguenti componenti:

- Repository SVN
- FishEye
- JIRA



# Iniziamo

Supponiamo che il sistema sia stato già configurato in modo da avere a disposizione **JIRA** come repository degli utenti di **FishEye**.

La prima operazione da eseguire, una volta che **FishEye** è stato installato, è di eseguire le operazioni di Login come utente amministratore e di configurare un accesso ad un repository. Una volta eseguito l'accesso:

- Tramite **COG** menù, accedere alla sezione di amministrazione. Quindi selezionare, da **Repository Settings, Repositories**:

The screenshot shows the FishEye administration interface. At the top, there's a navigation bar with links for 'Repositories', 'Projects', 'People', 'Reviews', and a 'Create review' button. Below the navigation bar, the user is identified as 'Admin'. On the left, there's a sidebar with collapsed sections for 'Repository Settings' (containing 'Repositories', 'Defaults', 'Smart Commits', and 'Web Hooks'), 'Project Settings' (containing 'Projects'), and 'User Settings' (containing 'Users', 'User Mappings', and 'Avatars'). The main content area is titled 'Repositories' and displays a table of existing repositories. The table has columns for 'Name', 'Type', and 'Description'. One repository, 'Dragons', is listed as a Mercurial repository with the description 'Dragons repository'. At the bottom of the table, there are buttons for 'Add repository', 'Refresh', and 'Filter repositories'.

- Selezionare quindi **Add repository**

A questo punto, seguire il wizard, in modo da configurare l'accesso di FishEye al repository. Vediamolo nel dettaglio:

## Add Repository - Page 1 of 3

### Basic Details

Repository Type:  ▾

Name: \*  ?

Description:  ?

Specificare la tipologia di repository. Fisheye consente di poter selezionare tra più tipologie (Es. Mercurial, CVS, Subversion, etc).

## Add Repository - Page 2 of 3

### SVN Connection Details

SVN URL: \*  ?

Path:  ?

Username:  ?

Password:  ?

Passare quindi a fornire i parametri di connessione. Questa sezione del wizard varia a seconda della tipologia di

repository. Viene quindi data la possibilità di eseguire il test di connessione, prima di andare al passo successivo.

Add Repository - Page 3 of 3

---

### Final Settings

Store Diff Info:  [?](#)

Enable Repository  [?](#)

After Adding:

### Checkout URL

Users can see a URL for checking out the repository's code.

Use the system default settings for showing checkout urls

Show URL

URL

Vengono quindi richieste le ultime configurazioni. Confermate anche queste ultime, si è configurato il Repository. Si può procedere alla fase di sincronizzazione. Al termine, si dispone di tutte le informazioni necessarie memorizzate sul DB di FishEye. Questa operazione potrebbe richiedere del tempo e, di conseguenza, occorre attendere che sia terminata prima di eseguire altre operazioni.

Questo è il risultato:



Come si può osservare, direttamente da FishEye andiamo a spulciare il codice. Ma questo è solo uno degli aspetti. Una volta collegato a JIRA, possiamo anche a referenziare direttamente il codice sorgente e le variazioni dello stesso da JIRA. Semplicemente, andando a vedere la sezione **source**, possiamo andare a leggere quali interventi sono stati eseguiti



## Activity

All Comments Work Log History Activity Source Reviews Builds

### Source changes - FishEye

atlassian\_dragons committed 1b7d467... (1 file) 11/May/11 12:20 AM

Create review for all 2 commits

Reviews: none

Create review

#### DRA-1 Repository Update

Dragons

REMOVED target/classes/com/atlassian/dragonslayers/App.class

atlassian\_dragons committed 922fd93... (4 files) 11/May/11 12:08 AM

Reviews: UNDER REVIEW

Create review

#### DRA-1 Repository Update

Dragons

ADDED pom.xml  
ADDED src/main/java/com/atlassian/dragonslayers/App.java  
ADDED src/test/java/com/atlassian/dragonslayers/AppTest.java  
ADDED target/classes/com/atlassian/dragonslayers/App.class

l tutto semplicemente andando a inserire, nel testo delle note a fronte del Commit, la chiave della issue JIRA. Infatti, usando questa chiave, Fisheye riesce a eseguire il link con la issue JIRA e attiva il TAB **source**.

i

## Conclusioni

Abbiamo visto, in questo post, che cosa consente di fare FishEye nel dettaglio. Nei prossimi post approfondiremo l'argomento con la Code Review (Crucible) e con altre operazioni che sono possibili con Fisheye.

# Fisheye – First Look

## *First Look*

In questo post iniziamo ad esaminare un componente della Suite di Atlassian. [FishEye](#).



## **Quale funzione svolge?**

Fondamentalmente è il componente della suite che si interfaccia con il sistema di controllo versione del codice.

*FishEye provides a read-only window into your Subversion, Perforce, CVS, Git, and Mercurial repositories, all in one place. Keep a pulse on everything about your code: Visualize and report on activity, integrate source with JIRA issues, and search for commits, files, revisions, or people*

Si tratta di un applicativo WEB che si interfaccia al sistema di controllo versione del codice e, agganciandosi agli altri componenti della suite, consente di poter arrivare ad un controllo completo dello sviluppo.



Attraverso FiehEye è possibile tracciare, usando una unica interfaccia, tutte le variazioni al codice di un software. Non solo. Integrandosi con JIRA e eseguendo opportune azioni, è possibile agganciare le variazioni che sono state operate al codice alla Issue JIRA di pertinenza, senza fatiche estreme e senza dover eseguire chissà quale numero di azioni.

## ***Come lo realizza?***

Semplicemente inserendo il codice della issue JIRA nel testo della commit. Provo a chiarire meglio. FishEye, semplicemente, esegue un polling al database del controllo versione configurato. Quando identifica una azione di commit, verifica che cosa è stato riportato come descrizione. Se riconosce la issue JIRA, verifica se esiste nel JIRA collegato a FishEye ed esegue un aggiornamento affinché il codice modificato sia agganciato alla ISSUE.

Il risultato è la possibilità di poter avere il controllo completo dello sviluppo, semplicemente accedendo alla pagina di dettaglio della Issue di JIRA.



Infatti, l'operazione di aggiornamento che FishEye esegue, è quella di agganciare il codice alla issue e fai si che, accedendo alla pagina della Issue, si possa anche visionare il codice modificato e dove è stata eseguita la modifica.

# ***Conclusioni***

Vi lascio immaginare il risultato, ma quello che si ottiene, letteralmente, è il controllo completo dello sviluppo/manutenzione del codice. Gli sviluppatori e i project manager dispongono così di uno strumento avanzato per poter monitorare tutte le azioni che sono eseguite sul codice, consentendo anche di migliorare la produzione ma non solo.

FishEye non è solo questo, ma anche [Code Review](#), attraverso uno strumento che viene fornito: ***Crucible***.



# ***Prossimi passi***

Nei prossimi post, andremo a vedere nel dettaglio come realizzare questa integrazione, come poter eseguire una code review e come poter integrare tutti questi strumenti.