INTEGRAZIONE BONITA 5.10 - DOCWAY 4

Bonita 5.10

Attraverso questa pagina vengono descritte le linee guida per la creazione di flussi di Bonita 5.10 da integrare su DocWay4.

L'integrazione dei form del flusso su DocWay avviene attraverso il componente **TaskForm** per Bonita (progetto *JSF2WorkflowComponent*). Prima di procedere con l'implementazione del workflow sull'IDE di Bonita **occorre verificare se le funzionalità richieste e i componenti necessari alla gestione dei form del flusso sono tutti supportati dal modulo TaskForm** (in caso contrario si potrebbero avere malfunzionamenti a livello di visualizzazione del form o ancor peggio a livello di salvataggio di dati del flusso).

Per maggiori informazioni su TaskForm si rimanda alla pagina di descrizione del componente TaskForm

Documentazione di base

Documentazione Bonita 5.10 (richiede la registrazione)

Variabili globali da definire sul workflow

In fase di definizione di un workflow occorre definire le seguenti variabili globali (che saranno riempite da DocWay durante l'assegnazione di un workflow ad un documento):

- **dw__nrecord** [STRING]: nrecord del documento al quale viene agganciato il workflow
- **dw__cod_persona** [STRING]: matricola dell'utente che avvia il workflow (iniziatore del workflow)
- **dw__cod_uff** [STRING]: codice dell'ufficio di appartenenza dell'utente che avvia il workflow
- **dw__document** [STRING]: XML del documento al quale viene agganciato il workflow. E' comodo per ottenere qualsiasi informazione sul documento in fase di avvio del workflow (caricamento dei dati attraverso script Groovy o metodi di Bonita). **N.B.** Non fare affidamento a questa variabile in connettori caricati in task successivi all'avvio perchè il documento potrebbe aver subito variazioni!
- **dw__proprietario** [STRING]: Cognome e nome del proprietario del documento
- **dw__oggetto** [STRING]: Oggetto del documento

N.B.: Oltre alle variabili appena indicate occorre definire (se è necessario caricare files sul documento tramite il workflow - connettore **XwAddAttachment**) una o più variabili **globali** di tipo **Allegato**. Per poter operare sui files **Bonita necessita obbligatoriamente** che i file vengano gestiti tramite variabili globali e non associate ad uno specifico task.

Connettori implementati

I connettori di Bonita si dividono in queste categorie:

- Connettori
- Selettore di attori
- Filtro

Connettori implementati:

NOME	TIPO	DESCRIZIONE
ACLRoleResolver	Selettore di attori	Dato il codice di un ruolo restituisce tutte le persone che ne fanno parte
ACLOfficeResolver	Selettore di attori	Dato il codice dell'ufficio restituisce tutte le persone che ne fanno parte
ACLFatherOfficeResolver	Selettore di attori	Dato il codice di un ufficio restituisce il responsabile dell'ufficio padre
XwLoadDocument	Connettore	Caricamento di un documento da eXtraWay
XwAddXPathDoc	Connettore	Dato un documento, un xpath ed un valore, provvede ad inserire tale valore nell'elemento/attributo individuato dall'xpath nel documento (**N.B.** richiede il successivo salvataggio del documento tramite il connettore XwSaveDocument)
XwRemoveXPathDoc	Connettore	Dato un documento ed un xpath, provvede ad eliminare tutti i nodi del documento (**N.B.** richiede il successivo salvataggio del documento tramite il connettore XwSaveDocument)
XwSaveDocument	Connettore	Salvataggio di un documento su eXtraWay. **N.B.** se il documento è stato modificato tramite add/remove xpath (o altro connettore) e ne è stato compromesso lo stato su DocWay, per forzare comunque il salvataggio occorre impostare come parametro del DB un **alias** (non si deve utilizzare xdocwaydoc altrimenti viene restituito un errore dai 3DIWS)
XwGrantRight	Connettore	Assegnazione di un documento ad una persona/ufficio/ruolo
XwDenyRight	Connettore	Eliminazione di un assegnatario dalla lista CC o CDS di un documento
XwApplicaSegnatura	Connettore	Protocollazione di un documento in bozza
XwAddAttachment	Connettore	Allega un file ad un documento
XwAddToDocument	Connettore	Aggiunge o sovrascrive un determinato xpath al campo extra
XwSavePostIt	Connettore	Salvataggio di un post-it (annotazione) all'interno del documento

NOME	TIPO	DESCRIZIONE
XwLoadFileIdsByPos	Connettore	Caricamento dei fileId di file allegati al documento in base alla loro posizione sul documento
XwLoadFileIdsByTitle	Connettore	Caricamento dei fileId di file allegati al documento in base al loro titolo
XwLoadFileIdsByLibroFirma	Connettore	Caricamento dei fileId di file allegati al documento in base alla selezione di file da firmare del Libro Firma
XwInserisciInLibroFirma	Connettore	Marca come da firmare con libro firma il fileid passato in ingresso
XwInserisciInLibroFirmaMultiplo	Connettore	Marca come da firmare con libro firma la lista di fileid passati in ingresso
XwBonitaLoadExternalProperties	Connettore	Permette di caricare i valori di file di properties in una mappa per poter estrarre informazioni utili nei vari task del workflow. (**N.B.** i file di properties, separati da **;** possono essere specificati attraverso la property **externalProperties** nel bonita-connectors.properties, oppure durante la configurazione del connettore).

Utilizzo generico dei connettori

I connettori vengono agganciati ai task per eseguire azioni specifiche (ad esempio al submit del form). Di seguito è mostrata la prima schermata che ci si presenta in fase di aggiunta di un connettore ad un task:

Tramite questa sezione viene definito l'evento sul quale attivare del connettore e la gestione degli errori.

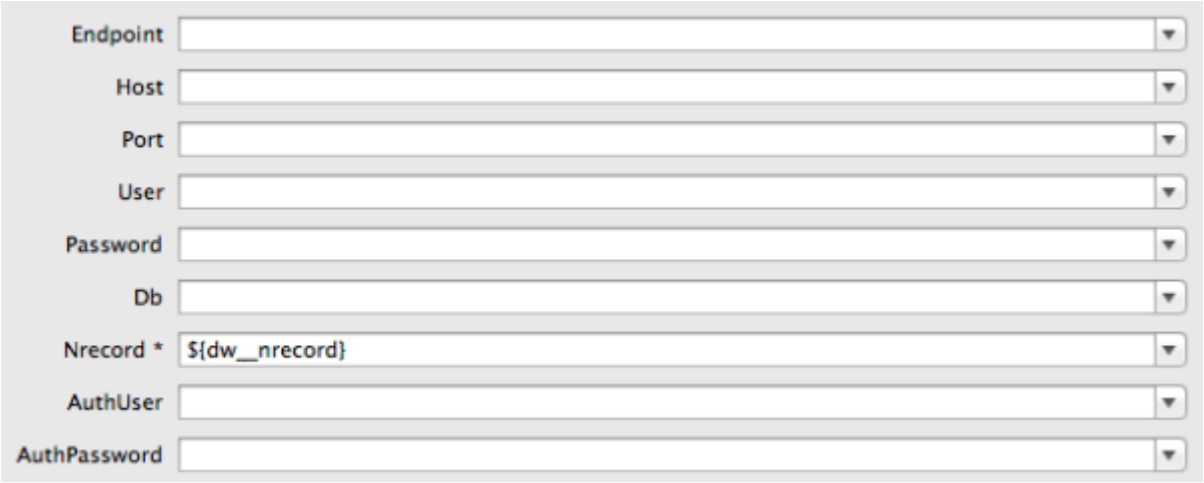
Eventi:

- **enter**: i connettori vengono avviati una volta che il task diventa attivo, quindi al completamento del task precedente, o all'avvio del workflow se si tratta del primo task
- **start**: i connettori vengono avviati quando l'utente avvia il task (prende in carico il task). Su DocWay questa attività corrisponde al submit del form del task (viene prima preso in carico il task ed immediatamente concluso), per questo motivo i connettori associati a start e finish vengono eseguiti nello stesso momento
- **suspend**: sospensione di un task da parte di un utente (attività non prevista da DocWay)
- **resume**: riapertura (post sospensione) di un task da parte di un utente (attività non prevista da DocWay)
- **finish**: i connettori vengono avviati al submit del form del task

Gestione Errori:

- **Solleva eccezione**: In caso di errore del connettore viene restituita l'eccezione. In questo caso il workflow rimarrà in uno bloccato sul task in errore. L'unica attività possibile sul workflow sarà quindi quella dell'annullamento
- **Ignora l'errore e continua il processo**: Ogni errore restituito dal connettore viene ignorato e il flusso avanza in ogni caso (es. l'errore in invio di una mail di notifica può non essere bloccante per il flusso)
- **Evento Throw Error**: L'errore viene catturato da Bonita e può essere gestito tramite gli eventi di “Cattura Errore”

Una volta definiti questi aspetti è possibile procedere con la configurazione specifica del connettore (immagine successiva).



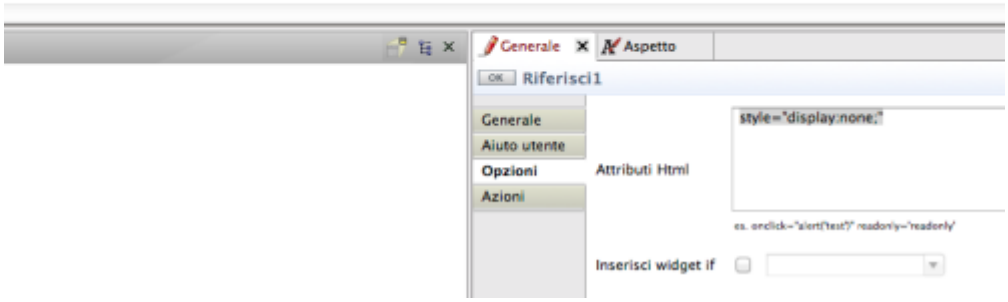
La maggior parte dei connettori 3DI presuppone l'utilizzo dei 3DIWS quindi molti dei parametri da configurare sui connettori riguardano i parametri di accesso ai webservices (endpoint, host, port, ecc.). Per semplificare l'attività di definizione dei flussi è stato implementato un meccanismo per il quale tutti questi parametri possono essere lasciati vuoti e, in fase di esecuzione del connettore, questi dati vengono letti da un file di properties dell'applicazione.

Firma digitale di files

Per definire un task di firma digitale di allegati di un documento di DocWay occorre procedere in questo modo:

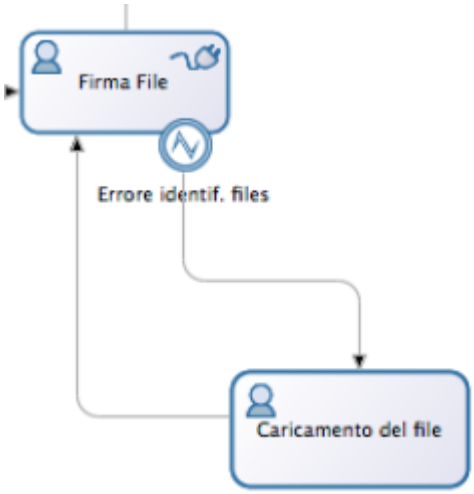
1. Aggiungere un task di tipo manuale (che richiede l'intervento di un utente)
2. Tab *Generale*, sezione *Attori*, identificare chi deve firmare il/i file (iniziatore o tramite un connettore di selezione attori)
3. Tab *Generale*, sezione *Dati*, definire una variabile con nome **dw__firmafiles_ids** di tipo stringa e lasciarla vuota. Questa variabile è obbligatoria per fare in modo che il task di firma sia completamente integrato con i servizi di firma digitale messi a disposizione su DocWay. In questo modo, quando l'utente firma il file direttamente dall'interfaccia di DocWay, viene verificata la presenza di task attivi e in attesa di firma e si controlla se è possibile fare avanzare o meno il flusso
4. Tab *Generale*, sezione *Connettori*, agganciare un connettore di recupero fileIds (*XwLoadFileIdsByPos*, *XwLoadFileIdsByTitle*, *XwLoadFileIdsByLibroFirma*) sull'enter del task. Il risultato del connettore deve essere associato alla variabile **dw__firmafiles_ids** definita sul task. In caso di errore (mancato riconoscimento dei file da firmare) potrebbe essere utile direzionare il flusso su un task di richiesta caricamento files, per poi tornare al task di firma
5. Tab *Applicazioni*, sezione *Inserimento del Pageflow*, definire un form senza alcun campo di inserimento, con un campo HTML widget che mostra un messaggio di richiesta firma all'utente e senza un pulsante di conferma (in modo che non possa essere possibile fare avanzare il flusso dal form del task)

Di seguito è mostrato un esempio di pageflow di firma:



Come si può notare dall'immagine, a differenza di quanto detto al punto 5 delle istruzioni di definizione del task, è presente nel pageflow il pulsante di submti del form (tipo 'Riferisci'). Questo per evitare che bonita mostri un errore (NON bloccante) sul flusso perchè è stato definito un form senza pulsante di submit. In questo caso per evitare che l'utente possa avanzare dal form è stato aggiunto l'attributo html *style*="*display:none;*" nella sezione *Opzioni* del pulsante del form

Esempio di task di firma con ridirezione ad un task di caricamento file in caso di errore in identificazione del/i file da firmare:



Gestione degli errori nella definizione del workflow

Questa sezione andrebbe approfondita meglio e integrata in base all'esperienza maturata in fase di definizione dei workflow

---

Per evitare blocchi sgradevoli in fase di avanzamento di un workflow, sarebbe necessario gestire eventuali errori che possono essere restituiti dai connettori in tutti i task che lo richiedono. In caso di errori non gestiti che provocano il blocco del flusso, infatti, l'unica attività possibile è l'annullamento dell'intero flusso.

Tutti i connettori che vengono agganciati ad un task dovrebbero gestire gli errori in uno di questi modi:

- Nel caso di errori non bloccanti, occorre definire il connettore in modo da ignorare l'errore e avanzare nel processo
- Nel caso di errori bloccanti, occorre definire delle tipologie di errori (**Evento Throw Error**) che dovranno poi essere gestite a livello di flusso

Di seguito è mostrato un esempio di gestione 'Throw Error':

Come si può notare dall'immagine, al task è stato aggiunto un elemento di **Cattura Errore** (associato all'errore restituito dal/i connettore/i del task) grazie al quale viene riattivato il task corrente in caso di errori. Attraverso questa gestione, qualsiasi errore del tipo gestito dall'elemento di cattura errore non provocherà il blocco del flusso ma riproporrà all'utente il task corrente. Su DocWay, nella storia del workflow, si avrà poi evidenza che il task è stato eseguito più volte a causa di errori emersi.

N.B.: La gestione degli errori descritta ora deve essere applicata SOLO a connettori attivati sullo start o finish di un task, **MAI sull'enter** (in questo caso infatti si provocherebbe un loop infinito sul task di Bonita).

Ovviamente questo è solo un esempio di come è possibile gestire gli errori tramite il **Cattura Errore**. Può essere utile in caso di errore attivare un altro task (es. errore in riconoscimento file da firmare di un task di firma) o addirittura attivare un sottoprocesso.

Esempio di flusso con gestione degli errori: