Produzione - call-centralized-flow: Flusso centralizzato Bandi di finanziamento

- Grafico del flusso
- Descrizione
- Etichette Stati
- Modello Dati
- Configurazioni
- Permessi
- Validazioni
- Logiche (action/start)

Grafico del flusso

HELPDESK
(HD)

DIVISIONE RICERCA
(DR)

ORGANI DIPARTIMENTALI
(OD)

RESPONSABILE O PROPRIETARIO
(RS)

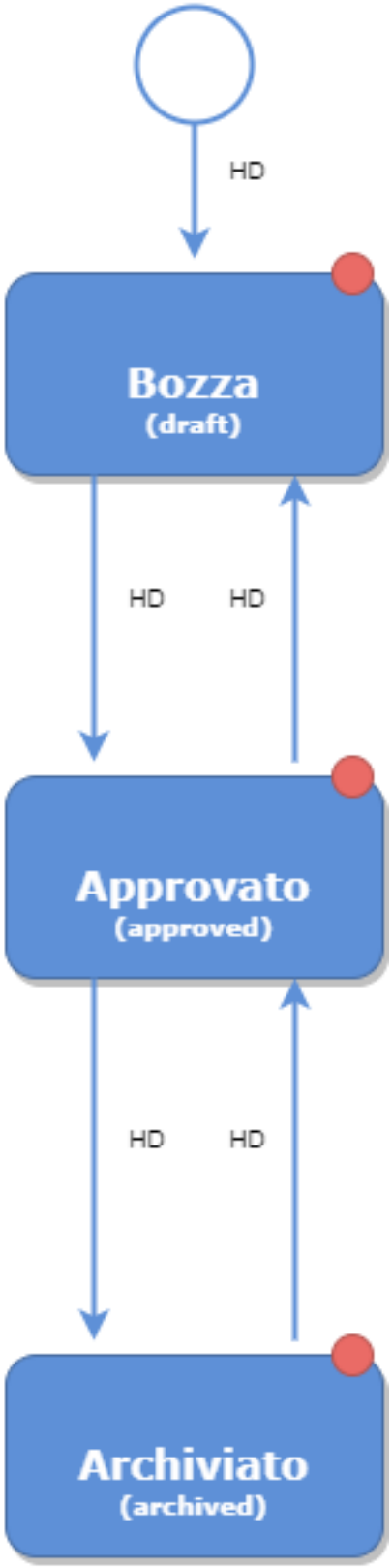
PARTECIPANTE O TUTOR
(PA/TU)

RESPONSABILE AMMINISTRATIVO
(RA)

DELEGATO O RAPPR. INTERNO
(RD)

CONTABILITÀ
(CO)

UFFICIO FORMAZIONE
(FO)



Descrizione

Il flusso call-flow è un **flusso centralizzato** che consente la gestione dei **bandi di finanziamento**.
Per informazioni sul **modello dati**, effettuare download del file excel disponibile al livello superiore.

Questo flusso prevede i seguenti attori nelle diverse visioni.

- Visione completa
  - **Helpdesk - HD**
    Team con profilo "Profilo Helpdesk per i Bandi (call)"

Trattandosi di un flusso centralizzato, la creazione di un nuovo bando è consentita SOLO all'HD, DR in visione completa.
L'HD avrà sempre accesso in scrittura al bando ed inoltre viene concessa qualunque transizione di stato anche senza seguire il flusso canonico: è quindi possibile anche effettuare "salti" di stato.
Nel grafico, per chiarezza, vengono riportate però solo le transizioni di stato canoniche.

Etichette Stati

I nomi degli stati possono essere personalizzati con la funzione di **Gestione etichette**.
Nella tabella sottostante vengono riportati tutti gli stati previsti dal flusso con relativa etichetta e valore di default.
Vengono anche fornite le etichette per i vari bottoni che consentono lo spostamento di stato.
Di default venegono utilizzate le etichette dei bottoni per lo stato successivo, è possibile configurare il flusso in maniera tale da utlizzare anche le etichette dei bottoni per lo stato precedente.
Esempio di utilizzo: l'oggetto si trova nello stato in attesa di essere validato, i possibili stati di cambio sono bozza e operativo. Il bottone per lo stato operativo utilizzerà l'etichetta per lo stato successivo, mentre il bottone per lo stato bozza utilizzerà l'etichetta per lo stato precedente.

Identificativo stato	Chiave etichetta stato	Valore default stato	Etichetta del bottone stato successivo	Etichetta del bottone stato precedente	Valore etichetta del bottone stato successivo	Valore etichetta del bottone stato precedente
draft	wfState.call.draft	Bozza	button.forward.to.call.draft	button.backward.to.call.draft	Salva e invia in "Bozza"	Salva e invia in "Bozza"
approved	wfState.call. approved	Approvato	button.forward.to.call.approved	button.backward.to.call.approved	Salva e invia in "Approvato"	Salva e invia in "Approvato"
archived	wfState.call.archived	Archiviato	button.forward.to.call.archived	button.backward.to.call.archived	Salva e invia in "Archiviato"	@button.backward.to.call.archived@

Modello Dati

Il dettaglio del modello dati associato a questo flusso è disponibile alla seguente pagina Produzione - Modello dati (call - CALL)

Configurazioni

Le configurazioni associate a questo flusso sono disponibili alla seguente pagina: Produzione - IRIS AP-RM Configurazioni: Bandi di finanziamento (call - CALL)

Permessi

Nella seguente sezione viene riportato il dettaglio dei permessi e delle transizioni di stato possibili per tutti gli attori del flusso.
Per quanto riguarda le transizioni di stato vengono riportati gli identificativi degli stati verso i quali è possibile effettuare la transizione.
Quando viene riportato il marcatore "__PREVIOUS_STATE__" significa che la transizione di stato è consentita verso lo stato precedente.
Di seguito la legenda dei permessi:

- c: create (disponibile solo per il primo stato del flusso)
- r: read
- w: write
- d: delete
- f: forward

Viene, inoltre, fornito dettaglio dei TAB disponibili (quelli in sola lettura presentano il suffisso readonly)

Stato	Attori	Permessi	Transizioni	Tab
Bozza (draft)	**Helpdesk (helpdesk)**	c r w d f	approved	• Dati Generali (call/form1) • Allegati (call/form2)
Approvato (approved)	**Helpdesk (helpdesk)**	r w d f	draft,archived	• Dati Generali (call/form1) • Allegati (call/form2)
Archiviato (archived)	**Helpdesk (helpdesk)**	r w d f	approved	• Dati Generali (call/form1) • Allegati (call/form2)

Validazioni

Nella seguente sezione viene riportato il dettaglio delle validazioni per tutte le coppie (attore, stato) del flusso.
Le validazioni sono distinte nei seguenti macrotipi e sono riferite, se non specificato altrimenti, all'oggetto radice.

- **enter**: validazione applicata in ingresso nello stato
  La transizione in ingresso viene NEGATA se anche solo una validazione NON viene superata con successo.
- **save**: validazione applicata ad ogni salvataggio e quindi anche per ogni spostamento di TAB
  Il salvataggio viene NEGATO se anche solo una validazione NON viene superata con successo.
- **delete**: validazione applicata in fase di eliminazione di un oggetto radice
- **element**: validazione applicata agli elementi figli di un oggetto radice
- **permissions**: logiche di generazione dinamica dei permessi (rwfd) sull'oggetto radice che sovrascrive i permessi di flusso (rwfd)

Le validazioni sono ulteriormente distinte nei seguenti tipi.

- **required**: validazione di obbligatorietà di un attributo sull'oggetto radice.
- **complex**: validazione complessa applicabile sia all'oggetto radice che agli elementi.
  Per avere maggiori dettagli sulla validazione cliccare sull'identificativo della validazione

Nel caso di validazioni di tipo **element**, oltre all'identificativo della validazione, viene riportato anche l'identificativo dell'elemento a cui è applicata e l'azione che l'ha scatenata:

- salvataggio (save)
- eliminazione (delete)

Ad esempio la seguente stringa
internalOrganizationUnit:delete *departmentDeleteValidator* indica che la validazione "departmentDeleteValidator" è applicata in eliminazione di un elemento di tipo internalOrganizationUnit dell'oggetto radice.

Per avere maggiori dettagli sui possibili elementi fare riferimento alla definizione del modello, disponibile nella sezione Modello Dati.

Infine è possibile applicare le validazioni, condizionalmente al soddisfacimento di determinate condizioni (opzionali).
Queste condizioni sono specificate nella colonna "Applicabilità": se è specificato **always**, la validazione è sempre attiva.
Per avere maggiori dettagli sulle possibili condizioni di applicabilità e dei relativi parametri, fare riferimento alla lista completa nella sezione Definizione Apply Logic condivise.

Stato	Attori	MacroTipo	Tipo	Attributo/Identificativo	Applicabilita'
Bozza (draft)	**all**	**enter**	**required**	wfItemTypeId	always
				dateMap[startDate]	always
				description	always
				booleanMap[isHierarchical]	configTrue (ap.call.createWithParent.enabled, false)
				booleanMap[isParent]	configTrue (ap.call.createWithParent.enabled, false)
			complex	checkCreationPermissionsValidator	always
				checkHierarchyValidator	configTrue (ap.call.createWithParent.enabled, false) booleanFieldTrue (isHierarchical, false)
		save	**required**	booleanMap[isHierarchical]	always
				booleanMap[isParent]	always
			complex	checkHierarchyValidator	configTrue (ap.call.createWithParent.enabled, false) booleanFieldTrue (isHierarchical, false)
		delete	**complex**	childAndParentDetectorDeleteValidator	always
Approvato (approved)	**all**	**enter**	**required**	wfItemTypeId	always
				description	always
				dateMap[startDate]	always
				dateMap[endDate]	booleanFieldFalse (openEnded, true)
				booleanMap[pnrr]	always
				stringMap[code]	always
				booleanMap[openEnded]	always
				booleanMap[competitive]	always
				booleanMap[isHierarchical]	always
				booleanMap[isParent]	always
				element:grantor	configTrue (ap.prj.create-by-call.enabled, false)
			complex	linkedProjectTypeValidator	configTrue (ap.prj.create-by-call.enabled, false)
				checkHierarchyValidator	configTrue (ap.call.createWithParent.enabled, false) booleanFieldTrue (isHierarchical, false)
				startDateAndEndDateValidator	always
				uniqueCodeValidator	always
			required	wfItemTypeId	always
				description	always
				dateMap[startDate]	always
				dateMap[endDate]	booleanFieldFalse (openEnded, true)
				stringMap[code]	always
				booleanMap[openEnded]	always
				booleanMap[competitive]	always

		save		booleanMap[isHierarchical]	always
				booleanMap[isParent]	always
				element:grantor	configTrue (ap.prj.create-by-call.enabled, false)
			complex	checkHierarchyValidator	configTrue (ap.call.createWithParent.enabled, false) booleanFieldTrue (isHierarchical, false)
				linkedProjectTypeValidator	configTrue (ap.prj.create-by-call.enabled, false)
				startDateAndEndDateValidator	always
				uniqueCodeValidator	always
		delete	**complex**	childAndParentDetectorDeleteValidator	always
Archiviato (archived)	**all**	**enter**	**required**	wfItemTypeId	always
				description	always
				dateMap[startDate]	always
				dateMap[endDate]	booleanFieldFalse (openEnded, true)
				stringMap[code]	always
				booleanMap[openEnded]	always
				booleanMap[competitive]	always
				booleanMap[isHierarchical]	always
				booleanMap[isParent]	always
				element:grantor	configTrue (ap.prj.create-by-call.enabled, false)
			complex	linkedProjectTypeValidator	configTrue (ap.prj.create-by-call.enabled, false)
				checkHierarchyValidator	configTrue (ap.call.createWithParent.enabled, false) booleanFieldTrue (isHierarchical, false)
				startDateAndEndDateValidator	always
				uniqueCodeValidator	always
		save	**required**	wfItemTypeId	always
				description	always
				dateMap[startDate]	always
				dateMap[endDate]	booleanFieldFalse (openEnded, true)
				stringMap[code]	always
				booleanMap[openEnded]	always
				booleanMap[competitive]	always
				booleanMap[isHierarchical]	always
				booleanMap[isParent]	always
				element:grantor	configTrue (ap.prj.create-by-call.enabled, false)
			complex	checkHierarchyValidator	configTrue (ap.call.createWithParent.enabled, false) booleanFieldTrue (isHierarchical, false)
				linkedProjectTypeValidator	configTrue (ap.prj.create-by-call.enabled, false)
				startDateAndEndDateValidator	always
				uniqueCodeValidator	always
		delete	**complex**	childAndParentDetectorDeleteValidator	always

Logiche (action/start)

Nella seguente sezione vengono riportate le

- START LOGICS
  Le start logics sono le "azioni" che vengono eseguite in fase di creazione di un nuovo oggetto radice
- ACTION LOGICS
  Le action logics sono delle "azioni" che vengono eseguite al verificarsi di determinati eventi.
  Gli eventi contemplati sono:
  - enter: ingresso in uno stato
  - save: salvataggio dell'oggetto radice

Di seguito viene riportato il dettaglio delle logiche definite per questo flusso.

START LOGICS

- wfStartLogicIdentifier
- wfStartLogicYearFromStartDate
- wfStartLogicParentCall
- wfStartLogicInferLinkedProjectTypeFromSiblingCalls
- wfStartLogicPnrrProject

ACTION LOGICS

- Bozza (draft)
  - **ENTER LOGICS**
  - **SAVE LOGICS**
    - wfActionLogicSaveNewYear
    - wfActionLogicHandleLinkedProjectType
    - wfActionLogicSaveHierarchicalCall
- Approvato (approved)

    - **ENTER LOGICS**
        - wfActionLogicHandleLinkedProjectType
    - **SAVE LOGICS**
        - wfActionLogicSaveNewYear
        - wfActionLogicHandleLinkedProjectType
        - wfActionLogicSaveHierarchicalCall
- Archiviato (archived)
    - **ENTER LOGICS**
        - wfActionLogicHandleLinkedProjectType
    - **SAVE LOGICS**
        - wfActionLogicSaveNewYear
        - wfActionLogicHandleLinkedProjectType
        - wfActionLogicSaveHierarchicalCall