

Integrazione DocSuite con gestionali esterni

Allegato Tecnico: Integrazione DocSuite con gestionali esterni

Fabrizio Lazzarotto
Product Manager

Versione 1.8
Del 23 Aprile 2021

dgroove srl

Via Monte Baldo, 6 | 37069 | Villafranca di Verona (VR) | Italia
P.IVA 02062090234 | Capitale Sociale 100.400€ i.v. | R.E.A. VR-215335
+39 045 86 00 171 | ordini@dgroove.it | www.dgroove.it

Sommario

1. Introduzione	5
2. Requisiti di integrazione	5
3. Architettura dell'integrazione	5
4. Scenari di integrazione	6
Avvio workflow "Crea Protocollo - automatico"	6
Avvio workflow "Crea Protocollo - manuale"	7
Avvio workflow "Invia PEC - automatico"	8
Avvio workflow "Firma digitale"	9
Avvio workflow "Crea Archivio/UDS"	10
Avvio workflow "Crea fascicolo chiuso"	11
5. Documentazione Public Web Api pubbliche di integrazione.....	12
Descrizione schema generico di un comando di avvio workflow	12
StartWorkflowContentType: Descrizione del modello che contiene le proprietà specifiche di avvio del workflow	14
WebApi Descrizione del modello – CollaborationModel	17
WebApi Descrizione del modello – SignerModel.....	18
WebApi Descrizione del modello – CategoryModel	18
WebApi Descrizione del modello – ContainerModel.....	19
WebApi Descrizione del modello – ArchiveModel.....	19
WebApi Descrizione del modello – DocumentUnitModel	20
WebApi Descrizione del modello – MetadataModel	21
WebApi Descrizione del modello – IdentityContext.....	21
WebApi Descrizione del modello – FascicleModel	22
WebApi Descrizione del modello – ContactModel	23
WebApi Descrizione del modello – DocumentModel	24
WebApi Descrizione del modello – DocSuiteSectorModel	25

WebApi Descrizione del modello – IdentityModel	25
WebApi Elenco delle parole chiave usato nei modelli – WorkflowParameterNames.....	26
Risposta del controller “StartWorkflow” contenente la chiave univoca di avvio avvenuto con successo.....	27
Eventi di risposta delle attività di workflow	28
Tabella dei modelli degli eventi di risposta ai workflow più usati	31
6. Interfacce URL di apertura DocSuite mediante Browser Web	32
Modulo “Protocollo”	32
Apertura sommario di un protocollo mediante Anno e Numero	32
Apertura visualizzazioni documenti di protocollo mediante Anno e Numero.....	32
Modulo “Dossier”	33
Apertura sommario dossier mediante parametri IdDossier (identificativo univoco).....	33
Apertura sommario dossier mediante parametri Anno e Numero	33
Apertura sommario dossier mediante parametro metadato specifico del dossier	33
Apertura sommario dossier e relativo fascicolo mediante IdDossier (identificativo univoco) e IdFascicle (identificativo univoco del fascicolo).....	34
Apertura sommario dossier mediante parametro metadato specifico del dossier e apertura cartellinatura specifica	34
Modulo “Fascicolo”	35
Apertura sommario fascicolo mediante parametri IdFascicle (identificativo univoco)	35
Apertura sommario fascicolo mediante parametri Anno e Numero.....	35
Apertura sommario fascicolo mediante parametro metadato specifico del fascicolo	35
7. Esempi dei modelli JSON di Avvio workflow ed eventi di risposta	36
Esempio 1 – Linguaggio C# - Avvio Workflow Firma digitale con protocollazione	36
Esempio 2 – Linguaggio C# - Avvio Workflow Protocollazione manuale.....	37
Esempio 3 – Linguaggio C# - Avvio Workflow Protocollazione automatica	38
Esempio 4 – Linguaggio C# - Avvio Workflow crea archivio	39
Esempio 5 – Linguaggio C# - Avvio Workflow Invia PEC automatica	39

Esempio 6 – Linguaggio C# - Avvio Workflow Creazione Fascicolo	40
Esempio 7 – Modello JSON - Evento “Protocollo creato” a fronte di un workflow “Protocollo documento”	41
Esempio 8 – Modello JSON - Evento “Protocollo creato” a fronte di un workflow “Collaborazione firma digitale”	41
Esempio 9 – Modello JSON - Evento “PEC ricevuta e associata al protocollo” a fronte di un workflow “Collaborazione firma digitale”	42
Esempio 10 – Modello JSON - Evento “PEC ricevuta e associata al protocollo” con esempio di indirizzo destinatario PEC specificato	43

1. Introduzione

Il seguente documento ha lo scopo di descrivere, in modalità tecnica, le interfacce e le metodologie che dovranno essere usate dai fornitori per integrarsi correttamente nella gestione esterna dei flussi documentali da e verso la DocSuite.

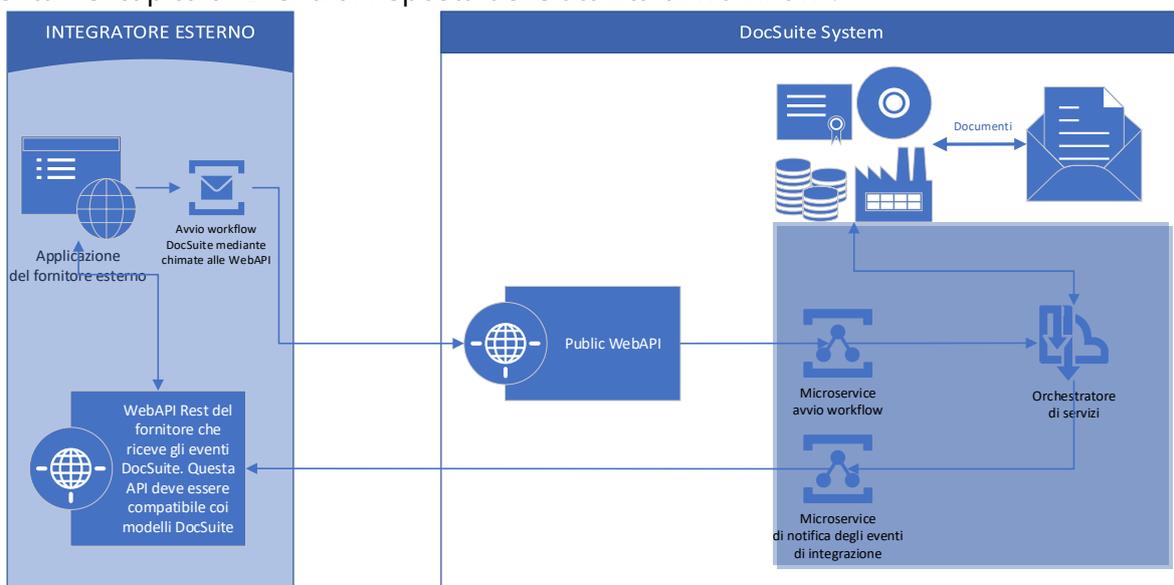
2. Requisiti di integrazione

I requisiti necessari a una corretta integrazione sono:

- Tutte le chiamate verso i servizi di integrazione sono autenticate tramite Windows NTLM.
- La tecnologia di riferimento prevede l'utilizzo di WebAPI REST con convenzione di utilizzo di modelli in formato JSON.
- Gli stream binari sono compatibili con la codifica Base64 (UTF-8)
- Il payload HTTP prevede l'encoding UTF8 con codifica UNICODE dei caratteri.
- Le date sono richieste nel formato ISO 8601 ["yyyy-MM-dd'T'HH:mm:ss.FFFK"]
- Il canale di comunicazione dovrebbe essere veicolato tramite criptazione SSL su HTTPS in quanto i modelli sono leggibili.

3. Architettura dell'integrazione

Di seguito viene mostrata l'architettura di riferimento che è composta principalmente da un unico 'entry point', ovvero le Public WebAPI Restfull che isolano la complessità di accesso al motore della DocSuite. Il fornitore a sua volta esporre una WebAPI compatibile coi modelli descritti nel capitolo "Eventi di risposta delle attività di workflow".

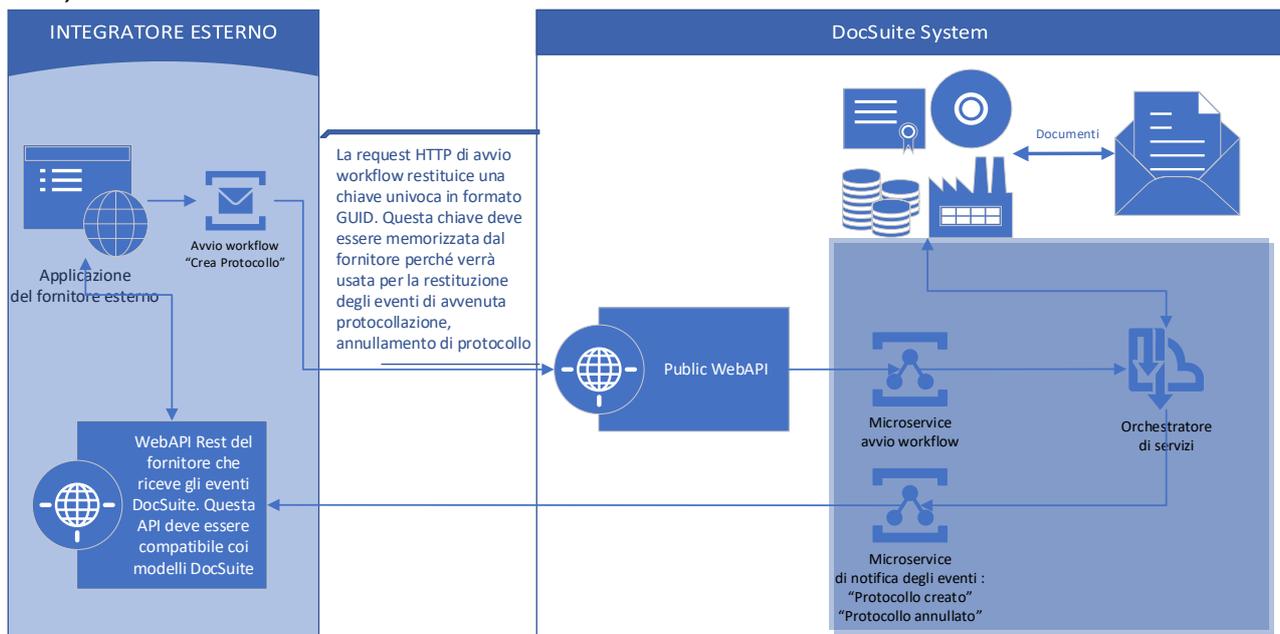


4. Scenari di integrazione

Avvio workflow "Crea Protocollo - automatico"

Lo scenario descrive il modello di integrazione del workflow "Crea protocollo - automatico". Tale workflow è pensato per "l'auto protocollazione" senza necessità di interazione delle segreterie o uffici. Il fornitore dunque deve necessariamente conoscere:

- 1) Classificazione
- 2) Contenitore
- 3) Eventuali settori per la distribuzione
- 4) Contatti mittente/destinatari
- 5) Tipologia ingresso/uscita
- 6) Oggetto
- 7) Documenti



Il processo di integrazione viene spiegato dallo schema sopra descritto. La richiesta viene presa in carico dalle Public WebAPI, validata per verificare l'esistenza e correttezza degli elementi specificati. Se l'avvio è andato a buon fine verrà restituito una "chiave univoca" (vedi dettagli specifico nel capitolo tecnico). Il processo di creazione del protocollo è asincrono rispetto alla richiesta di avvio workflow.

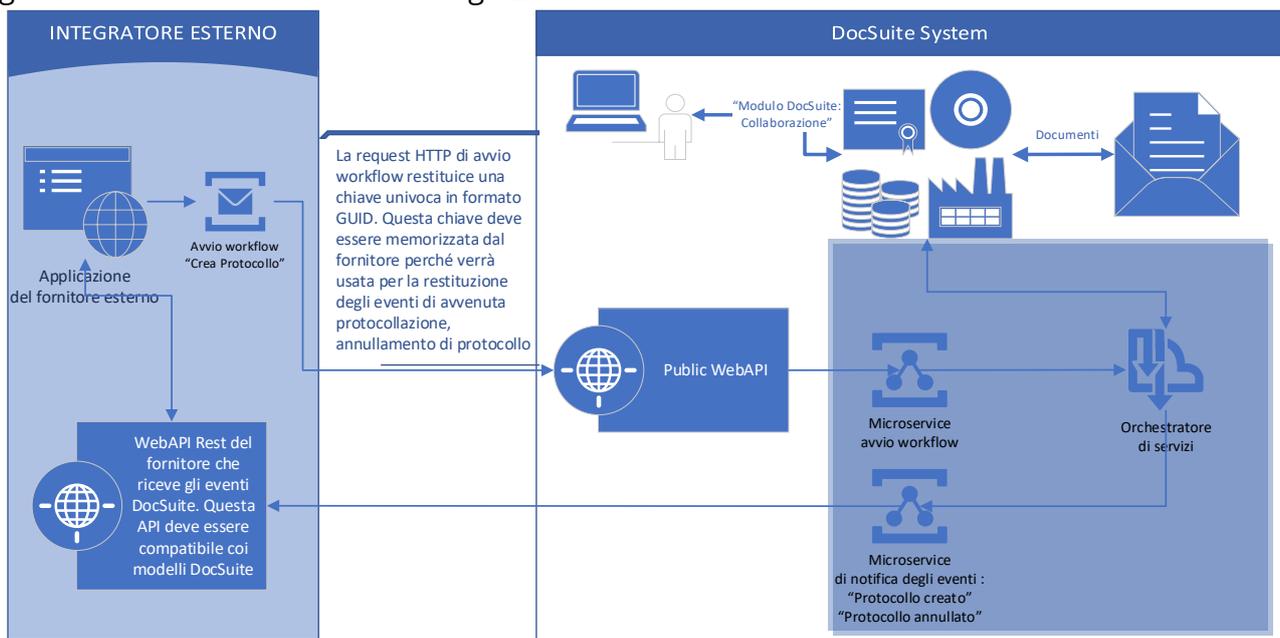
Il microservizio preposto DocSuite avrà il compito notificare gli eventi di creazione protocollo e annullamento protocollo, mediante chiamata alla WebAPI REST che il fornitore dovrà rendere disponibile. Questa API deve essere compatibile coi modelli descritti nel capitolo tecnico.

Avvio workflow "Crea Protocollo - manuale"

Lo scenario descrive il modello di integrazione del workflow "Crea protocollo - manuale". Tale workflow è pensato per una protocollazione manuale con responsabilità in carico all'ufficio competente. Il fornitore dunque deve necessariamente conoscere:

- 1) Il settore responsabile che dovrà prendere in carico la richiesta di protocollazione
- 2) Eventuali metadati di protocollo, se noti.

L'utente DocSuite utilizzerà il modulo "Scrivania di collaborazione" per procedere alla richiesta di protocollazione. E' necessario configurare un apposito template di collaborazione per identificare le collaborazioni provenienti dai possibili workflows in modo tale da differenziare gli eventi di ritorno a fronte di integrazioni con fornitori diversi.



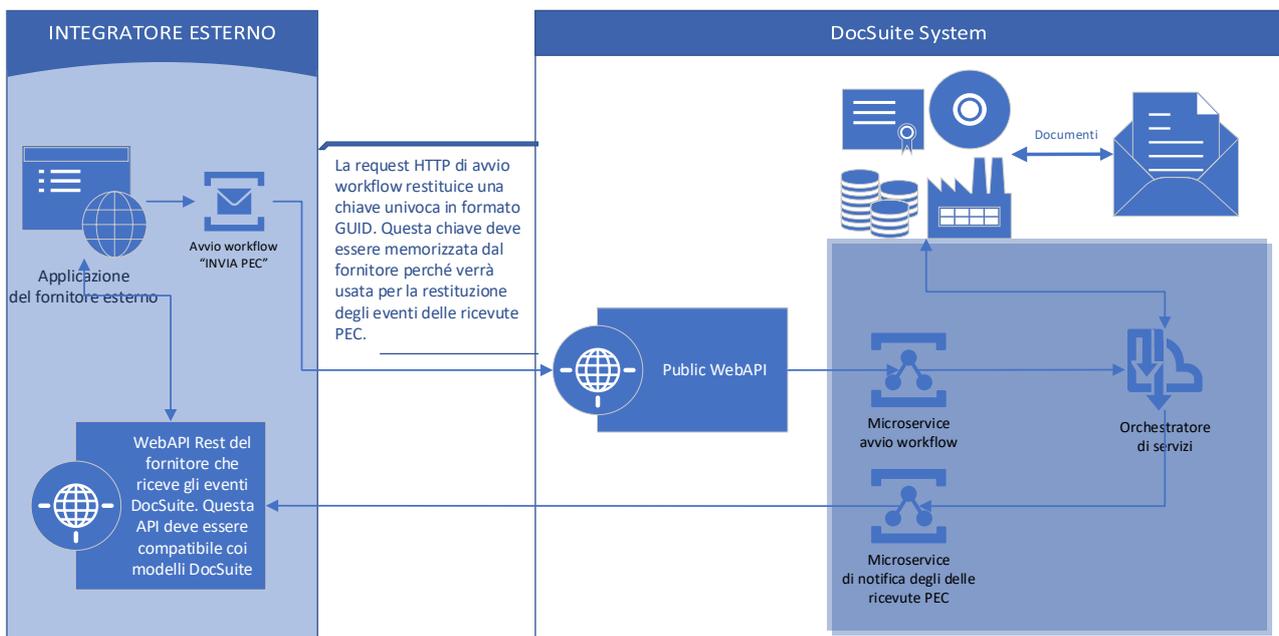
Il processo di integrazione è semplificato dallo schema sopra descritto. La richiesta viene presa in carico dalle Public WebAPI, validata per verificare l'esistenza e correttezza degli elementi specificati. Se l'avvio è andato a buon fine verrà restituito una "chiave univoca" (vedi dettagli

specifico nel capitolo tecnico). Il processo di completamento della richiesta di protocollazione da parte delle segreterie preposte è asincrono rispetto alla richiesta di avvio workflow. Il microservizio DocSuite avrà il compito notificare gli eventi di creazione protocollo e annullamento protocollo, mediante chiamata alla WebAPI REST che il fornitore dovrà rendere disponibile. Questa API deve essere compatibile coi modelli descritti nel capitolo tecnico.

Avvio workflow "Invia PEC - automatico"

Lo scenario descrive il modello di integrazione del workflow "Invia PEC". Tale workflow è pensato come automatismo e non necessità di interazioni con segreterie e/o uffici. Il fornitore dunque deve necessariamente conoscere:

- 1) Casella di invio
- 2) Oggetto
- 3) Testo
- 4) Destinatari
- 5) Allegati



Il processo di integrazione viene spiegato dallo schema sopra descritto. La richiesta viene presa in carico dalle Public WebAPI, validata per verificare l'esistenza e correttezza degli elementi specificati. Se l'avvio è andato a buon fine verrà restituito una "chiave univoca" (vedi dettagli

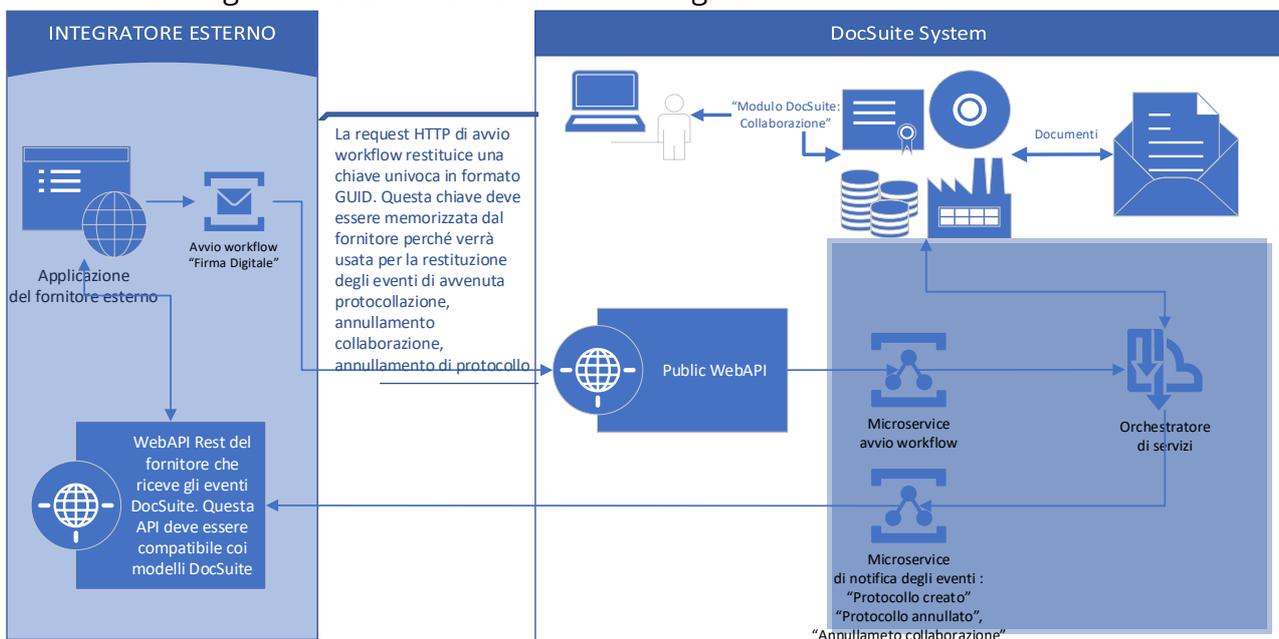
specifico nel capitolo tecnico). Il processo di invio PEC è asincrono rispetto alla richiesta di avvio workflow. Il microservizio preposto DocSuite avrà il compito notificare gli eventi previsti dal protocollo PEC, mediante chiamata alla WebAPI REST che il fornitore dovrà rendere disponibile. Questa API deve essere compatibile coi modelli descritti nel capitolo tecnico.

Avvio workflow "Firma digitale"

Lo scenario descrive il modello di integrazione del workflow "Firma digitale". Tale workflow è pensato per richiedere la firma elettronica qualificata EIDAS (CADES o PADES) di documentazione nella modalità "sequenziale". Il flusso operativo DocSuite viene preso in carico dai singoli firmatari richiesti e termina con la protocollazione, fa parte dell'ufficio competente, della documentazione firmata. Il fornitore dunque deve necessariamente conoscere:

- 1) Elenco dei firmatari, con possibilità di specificare l'ordine di firma e l'obbligatorietà.
- 2) Documenti
- 3) Il settore responsabile che dovrà prendere in carico la richiesta di protocollazione
- 4) Eventuali metadati di protocollo, se noti.

L'utente DocSuite utilizzerà il modulo "Scrivania di collaborazione" per procedere sia alla firma digitale qualificata che alla protocollazione. È necessario configurare un apposito template di collaborazione per identificare le collaborazioni provenienti dai possibili workflows in modo tale da differenziare gli eventi di ritorno a fronte di integrazioni con fornitori diversi.



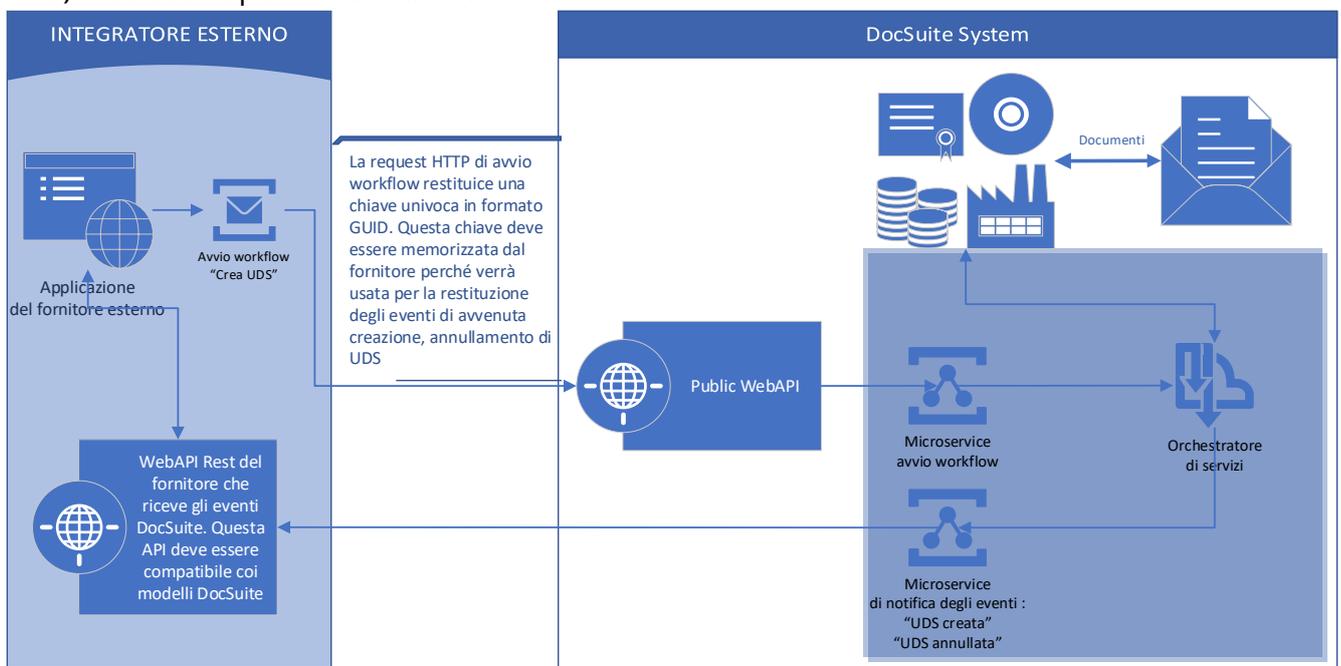
Il processo di integrazione è semplificato dallo schema sopra descritto. La richiesta viene presa in carico dalle Public WebAPI, validata per verificare l'esistenza e correttezza degli elementi specificati. Se l'avvio è andato a buon fine verrà restituito una "chiave univoca" (vedi dettagli specifico nel capitolo tecnico). Il processo di completamento della richiesta di protocollazione da parte delle segreterie preposte è asincrono rispetto alla richiesta di avvio workflow.

Il microservizio DocSuite avrà il compito notificare gli eventi di creazione protocollo, annullamento di collaborazione e annullamento protocollo, mediante chiamata alla WebAPI REST che il fornitore dovrà rendere disponibile. Questa API deve essere compatibile coi modelli descritti nel capitolo tecnico.

Avvio workflow "Crea Archivio/UDS"

Lo scenario descrive il modello di integrazione del workflow "Crea archivio - automatico". Tale workflow è pensato per la creazione automatica di archivio senza necessità di interazione delle segreterie o uffici. Il fornitore dunque deve necessariamente conoscere:

- 1) Classificazione
- 2) Nome archivio
- 3) Eventuali settori per la distribuzione
- 4) Oggetto
- 5) Metadati specifici dell'archivio



Il processo di integrazione viene spiegato dallo schema sopra descritto. La richiesta viene presa in carico dalle Public WebAPI, validata per verificare l'esistenza e correttezza degli elementi specificati. Se l'avvio è andato a buon fine verrà restituito una "chiave univoca" (vedi dettagli specifico nel capitolo tecnico). Il processo di creazione dell'archivio è asincrono rispetto alla richiesta di avvio workflow.

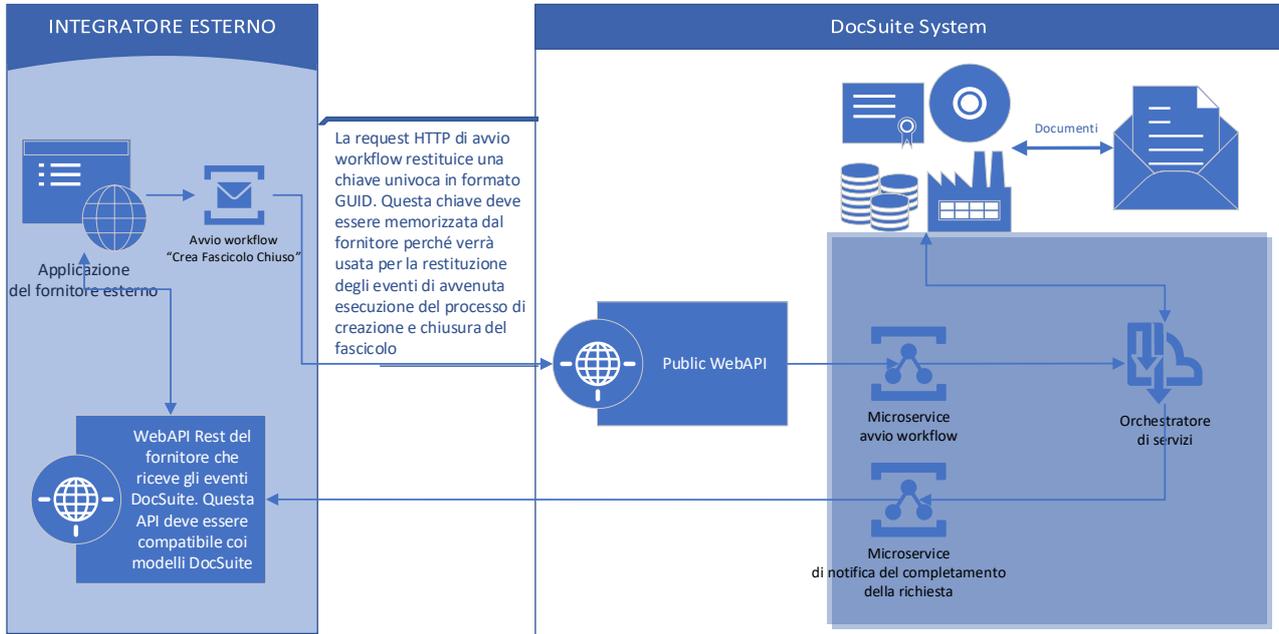
Il microservizio preposto DocSuite avrà il compito notificare gli eventi di creazione archivio e annullamento archivio, mediante chiamata alla WebAPI REST che il fornitore dovrà rendere disponibile. Questa API deve essere compatibile coi modelli descritti nel capitolo tecnico.

Avvio workflow "Crea fascicolo chiuso"

Lo scenario descrive il modello di integrazione del workflow "Crea fascicolo chiuso" pensato per l'archiviazione dei procedimenti/pratiche già concluse nella normale operatività dei flussi esterni alla DocSuite. L'esempio classico è il flusso di approvvigionamento che nella sua operatività risiede nei sistemi gestionali contabili, ma che a istruttoria conclusa passa l'archiviazione agli uffici amministrativi. Tale workflow è pensato come un automatismo che non necessita di interazione di segreterie e/o uffici. Il fornitore dunque deve necessariamente conoscere:

- 1) Classificazione
- 2) Oggetto
- 3) Responsabile di procedimento
- 4) Settori responsabile ed eventuali settori autorizzativi
- 5) Elenco delle unità documentali da inserire nel fascicolo
- 6) Elenco degli inserti (se previsti)
- 7) Metadati specifici del fascicolo (se previsto)

Sarà possibile prendere accordi specifici durante la fase di avvio per cui alcuni metadati siano attribuiti dal workflow stesso mediante configurazione. Tali metadati però saranno da intendere come "statici" per ogni richiesta. Non sarà possibile richiedere una "valutazione" dei metadati passati dal fornitore. Eventualmente fosse necessaria tale operare in situazioni simili, verranno create più tipologie di workflow che il fornitore potrà usare. Ogni tipologia di workflow è sempre identificata dall'attributo "Nome".



Il processo di integrazione viene spiegato dallo schema sopra descritto. La richiesta viene presa in carico dalle Public WebAPI, validata per verificare l'esistenza e correttezza degli elementi specificati. Se l'avvio è andato a buon fine verrà restituito una "chiave univoca" (vedi dettagli specifico nel capitolo tecnico). Il processo di creazione del fascicolo è asincrono rispetto alla richiesta di avvio workflow.

Il microservizio preposto DocSuite avrà il compito notificare l'evento che conterrà gli identificativi del fascicolo creato. Tale notifica avverrà mediante chiamata alla WebAPI REST che il fornitore dovrà rendere disponibile. Questa API deve essere compatibile coi modelli descritti nel capitolo tecnico.

5. Documentazione Public Web Api pubbliche di integrazione

Descrizione schema generico di un comando di avvio workflow

Le Public Web API utilizzano lo standard RESTFULL per la gestione delle richieste di avvio workflow mediante lettura del modello "StartWorkflowCommand" serializzato in formato json contenuto nel payload della richiesta HTTP.

Il controller API che gestisce la richiesta è "StartWorkflow" e dovrà essere sempre invocato mediante l'utilizzo del verbo POST del protocollo HTTP.

Il modello `“StartWorkflowCommand”` è composto da un insieme minimo di proprietà che elenchiamo:

- `Id`: `“<GUID>”`[Valore univoco da inserire nel comando. Serve per agganciare gli eventi al comando. Verrà utilizzato come valore chiave durante l’invio degli eventi restituiti dalla DocSuite.]
- `MessageName`: `“<nome comando>”`[il nome del comando è una stringa. I valori sono documentato nei capitoli successivi]
- `MessageDate`: [valore stringa che rappresenta un `DateTimeOffset` nel formato `“yyyy-MM-dd'T'HH:mm:ss.FFFK”`]
- `TenantName`: Valore stringa statico fornito da Dgroove
- `TenantId`: Valore stringa in formato GUID statico fornito da Dgroove
- `TenantA00Id`: Valore stringa in formato GUID statico fornito da Dgroove
- `ContentType`: Contiene il modello `“StartWorkflowContentType”` dove dovranno essere specificate l’elenco delle proprietà che il workflow si aspetta di ricevere in ingresso.
- `IdentityContext`: Struttura che identifica l’identità utente del chiamante. In questo campo è necessario identificare l’utente “umano” che fisicamente sta effettuando la richiesta.
- `CustomProperties`: è un dizionario che contiene alcune informazioni essenziali per la codifica corretta del comando.
 - `“messageName”`: `“StartWorkflowCommand”` valore statico
 - `“messageType”`: `“StartWorkflowCommand”` valore statico
 - `“messageDate”`: `<specificare la data di richiesta nel formato “yyyy-MM-dd'T'HH:mm:ss.FFFK”>`
 - `“tenantName”`: `<valore stringa statico fornito Dgroove>`
 - `“tenantId”`: `<valore statico GUID fornito da Dgroove>`
 - `“tenantA00Id”`: `<valore statico GUID fornito da Dgroove>`
 - `“id”`: `<valore univoco che identifica il correlationId>`
 - `“identity”`: `<nome utente con dominio o utente del sistema chiamante che sta effettuato la richiesta>`. Va sempre specificato il nome utente di contesto che sta usando l’applicazione reale. Non vanno mai indicati nomi utente di servizio.
 - `“contentType”`: `“WorkflowModel”` valore statico

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "1",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Commands.Workflows.StartWorkflowCommand, VecompSoftware.DocSuite.Public.Core.Models",
  "contentType": {
    "$id": "2",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.ContentTypes.Workflows.StartWorkflowContentType, VecompSoftware.DocSuite.Public.Core.Models",
    "correlationId": "895a4427-c214-4e56-b104-4e3c09bd8feb",
    "executorUser": "MASTER_USL\\docsuite",
    "content": [...]
  },
  "messageDate": "2020-07-16T13:01:12.7135089+00:00",
  "identityContext": [...],
  "tenantA00Id": "2b92a721-f77b-4b6b-afa8-a854a3a27c8c",
  "tenantId": "c0a21f9f-49d4-4701-9863-b7fc2b16493b",
  "tenantName": "AUSL-RE",
  "messageName": "StartWorkflowCommand",
  "customProperties": [...],
  "id": "39141950-893c-437f-b2cd-94807a2d7778"
}
```

StartWorkflowContentType: Descrizione del modello che contiene le proprietà specifiche di avvio del workflow

Il modello "StartWorkflowContentType" è un modello definito come "scatola", ovvero raccoglie l'elenco delle proprietà necessarie al corretto avvio del workflow.

Esponiamo con uno schema sintattico le parti più significative.

```
"contentType": {
  "correlationId": <da non usare, è un valore speciale attribuito internamente dal workflow>,
  "executorUser": <valore string corrisponde all'identità utente che sta effettuando la richiesta>,
  "content": {
    "Id": <valore guid che identifica univocamente il comando>.
    "name": <valore string del nome di workflow da avviare. Il valore verrà comunicato da Dgroove>,
    "workflowParameters": [<Collezione di parametri di avvio del workflow. Ogni parametro contiene le proprietà caratteristiche dell'oggetto che si vuole creare/specificare.>]
  }
}
```

I modelli attualmente gestiti dal motore di workflow sono:

- **ArchiveModel**: rappresenta il modello descrittivo dell'unità documentale "Repertorio"/"UDS".
- **CategoryModel**: rappresenta il modello descrittivo del metadato "Classificatore".
- **ContainerModel**: rappresenta il modello descrittivo del metadato "Contenitore".
- **ContactModel**: rappresenta il modello descrittivo del metadato "Contatto".
- **DocSuiteSectorModel**: rappresenta il modello descrittivo del metadato "Settore".
- **DocumentModel**: rappresenta il modello descrittivo del metadato "Documento".
- **DossierModel**: rappresenta il modello descrittivo dell'unità archivistica "Dossier".
- **FascicleModel**: rappresenta il modello descrittivo dell'unità archivistica "Fascicolo".

- **MetadataModel**: rappresenta il modello generico di “metadato”. Viene usato ad esempio per la definizione dei metadati di fascicolo o dei metadati specifici del repertorio.
- **DocumentUnitModel**: rappresenta il modello generale che identifica l'unità documentale. Questo modello viene usato per specificare i riferimenti identificativi della UD.
- **ReferenceModel**: rappresenta il modello generico per specificare il riferimento di qualunque elemento DocSuite.
- **SignerModel** rappresenta il modello descrittivo “Firmatario”.

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
contentType": {
  "$id": "2",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.ContentTypes.Workflows.StartWorkflowContentType, VecompSoftware.DocSuite.Public.Core.Models",
  "correlationId": "895a4427-c214-4e56-b104-4e3c09bd8feb",
  "executorUser": "MASTER_USL\\docsuite",
  "content": {
    "$id": "3",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowModel, VecompSoftware.DocSuite.Public.Core.Models",
    "activityTitlePrefix": "",
    "workflowParameters": {
      "$id": "4",
      "$values": [
        {
          "$id": "5",
          "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
          "parameterName": "dsw_p_fascicle_model",
          "parameterModel": {
            "$id": "6",
            "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.FascicleModel, VecompSoftware.DocSuite.Public.Core.Models",
            "subject": "Oggetto del fascicolo",
            "category": {
              "$id": "7",
              "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.CategoryModel, VecompSoftware.DocSuite.Public.Core.Models",
              "uniqueId": "82faca3c-08c5-439d-940d-a2b11394d6fe",
              "name": "",
              "mappingTag": ""
            },
            "manager": {
              "$id": "8",
              "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.ContactModel, VecompSoftware.DocSuite.Public.Core.Models",
              "contactId": "b7a73278-3635-4229-a2d9-3d2b4b274bbe",
              "archiveSection": "",
              "contactType": 64,
              "mappingTag": ""
            }
          }
        }
      ]
    }
  }
}
```

Per gli integratori che adottano la tecnologia Microsoft .NET, i modelli sono pubblicati sulla in appositi pacchetti distribuiti sulla piattaforma NUGET <https://www.nuget.org>.

Le librerie vengono fornite con compatibilità Microsoft .NET 4.8.



1.388 packages returned for DocSuite

[Hide prerelease](#)

-  [VecompSoftware.DocSuite.Public.Core.Models](#) by: [Fabrizio.Lazzarotto](#)
↓ 2.503 total downloads | ⌚ last updated an hour ago | 📄 Latest version: 9.7.0 | 🔗 VecompSoftware Models DocSuite Integrazioni Standard DocSuite...
Modelli standard per l'integrazione, tramite WebAPI, con il sistema documentale DocSuite
-  [VecompSoftware.DocSuite.Public.Helpers](#) by: [Fabrizio.Lazzarotto](#)
↓ 1.310 total downloads | ⌚ last updated 19/12/2016 | 📄 Latest version: 8.56.0 | 🔗 VecompSoftware Models DocSuite Integrazioni Helpers DocSuite...
Libreria coi metodi di helpers standard per l'integrazione, tramite WebAPI, con il sistema documentale DocSuite
-  [VecompSoftware.DocSuite.Public.Core.Models.Customs](#) by: [Fabrizio.Lazzarotto](#)
↓ 1.365 total downloads | ⌚ last updated 19/12/2016 | 📄 Latest version: 8.56.0 | 🔗 VecompSoftware Models DocSuite Integrazioni Custom DocSuite...
Modelli custom per l'integrazione, tramite WebAPI, con il sistema documentale DocSuite
-  [VecompSoftware.DocSuite.Public.Core.Models.4.5](#) by: [Fabrizio.Lazzarotto](#)
↓ 738 total downloads | ⌚ last updated 21/12/2016 | 📄 Latest version: 8.56.0.3 | 🔗 VecompSoftware Models DocSuite Integrazioni Standard DocSuite...
Modelli standard per l'integrazione, tramite WebAPI, con il sistema documentale DocSuite
-  [VecompSoftware.DocSuite.Public.Core.Models.Helpers](#) by: [Fabrizio.Lazzarotto](#)
↓ 442 total downloads | ⌚ last updated an hour ago | 📄 Latest version: 9.7.0 | 🔗 VecompSoftware Models DocSuite Integrazioni Helpers DocSuite.Pu...
Libreria coi metodi di helpers dei modelli per l'integrazione, tramite WebAPI, con il sistema documentale DocSuite
-  [VecompSoftware.DocSuite.Public.Helpers.4.5](#) by: [Fabrizio.Lazzarotto](#)
↓ 309 total downloads | ⌚ last updated 19/12/2016 | 📄 Latest version: 8.56.0 | 🔗 VecompSoftware Models DocSuite Integrazioni Helpers DocSuite.Pu...
Libreria coi metodi di helpers standard per l'integrazione, tramite WebAPI, con il sistema documentale DocSuite

WebApi Descrizione del modello - CollaborationModel

Modello dell'Unità Documentaria prodotta dalla collaborazione di firma/protocollo

- Direction: Specifica se l'unità documentaria è in ingresso o in uscita
 - Inbound = -1 (Ingresso)
 - Internal = 0 (Interno / Tra uffici)
 - Outgoing = 1 (Uscita)
- DocumentUnitType: Tipologie di unità documentarie della DocSuite
 - Protocol = 1 (Protocollo)
 - Resolution = 2 (Atto)
 - DocumentSeries = 4 (Serie documentali)
 - Archive = 8 (Archivio / Unità documentaria specifica)
- Category = CategoryModel del Classificatore/Titolario dell'unità documentaria
- Container = ContainerModel del contenitore dell'unità documentaria
- Subject = Oggetto dell'unità documentaria
- Note = Note dell'unità documentaria
- Metadatas = Collezione di MetadataModel coi metadati dell'archivio
- Contacts = Collezione di ContactModel coi contatti
- Sectors = Collezione di DocSuiteSectorModel dei settori autorizzato all'unità documentaria

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "5",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
  "parameterName": "dsw_p_collaboration_model",
  "parameterModel": {
    "$id": "6",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.CollaborationModel, VecompSoftware.DocSuite.Public.Core.Models",
    "subject": "Richiesta firma digitale",
    "note": "campo note di collaborazione",
    "direction": 1,
    "documentUnitType": 1,
    "collaborationPriority": 2,
    "metadatas": {
      "$id": "7",
      "$values": []
    },
    "contacts": {
      "$id": "8",
      "$values": []
    },
    "sectors": {
      "$id": "9",
      "$values": []
    }
  },
  "id": "505786db-41eb-482d-9c93-83ae1467c351",
  "name": "CollaborationModel"
},
```

WebApi Descrizione del modello - SignerModel

Modello del firmatario tipicamente usato nei workflow di firma digitale

- SignerType: Tipologia di utente di firma
 - AD = 1 (Utente di dominio)
 - DSWRole = 2 (Settore)
 - Mapping = 4 (Codice di mapping concordato con Dgroove)
- Identity: IdentityModel dell'utente (da usare in caso di SignerType = AD)
- Order = Posizione del firmatario. Deve iniziare col valore 1
- Required = Valore true/false che identifica l'obbligatorietà della firma qualificata

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "10",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
  "parameterName": "dsw_p_collaboration_signer",
  "parameterModel": {
    "$id": "11",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.SignerModel, VecompSoftware.DocSuite.Public.Core.Models",
    "order": 1,
    "required": true,
    "identity": {
      "$id": "12",
      "$type": "VecompSoftware.DocSuite.Public.Core.Models.Securities.IdentityModel, VecompSoftware.DocSuite.Public.Core.Models",
      "account": "<utente firmatario nel formato dominio\\samAccount>",
      "authorization": 1,
      "id": "db99ee1a-b27e-4561-b111-4301a44f60c5",
      "name": "<utente firmatario nel formato dominio\\samAccount>"
    },
    "signerType": 1
  },
  "id": "9f4b5caa-be21-4524-b45e-d0b356d9f09a",
  "name": "SignerModel"
},
```

WebApi Descrizione del modello - CategoryModel

Modello del Classificatore / Titolare

- Name: Voce del classificatore
- UniqueId: Codice del classificatore della DocSuite
- MappingTag: E' possibile usare un codice di trascodifica da concordare con Dgroove

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
    "$id": "17",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
    "parameterName": "dsw_p_protocol_category",
    "parameterModel": {
      "$id": "18",
      "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.CategoryModel, VecompSoftware.DocSuite.Public.Core.Models",
      "uniqueId": "37579c9e-8883-49ad-bf64-5bd95e64e0e1",
      "name": "",
      "mappingTag": ""
    },
    "id": "14adf817-f6f9-4398-969b-b6be66dbb148",
    "name": "CategoryModel"
  },
}
```

WebApi Descrizione del modello – ContainerModel

Modello del Contenitore dell'unità documentaria

- Name: Nome del contenitore
- UniqueId: Codice del contenitore della DocSuite
- MappingTag: E' possibile usare un codice di trascodifica da concordare con Dgroove

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
    "$id": "19",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
    "parameterName": "dsw_p_protocol_container",
    "parameterModel": {
      "$id": "20",
      "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.ContainerModel, VecompSoftware.DocSuite.Public.Core.Models",
      "uniqueId": "4b0f483b-a6a3-43de-ba9d-149815e518f1",
      "name": "",
      "mappingTag": ""
    },
    "id": "92d7f776-a520-4fa6-8d32-beb00ac4d214",
    "name": "ContainerModel"
  },
}
```

WebApi Descrizione del modello – ArchiveModel

Modello dell'Archivio / Unità Documentaria Specifica

- ArchiveName: Nome dell'archivio documentale
- ArchiveId: Identificativo dell'archivio documentale
- Metadatas: Collezione di MetadataModel coi metadati dell'archivio

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "5",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
  "parameterName": "dsw_p_archive_model",
  "parameterModel": {
    "$id": "6",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.ArchiveModel, VecompSoftware.DocSuite.Public.Core.Models",
    "archiveName": "Contratti",
    "archiveId": "5a9586c9-ea83-45df-8816-e05f058ad0b1",
    "metadatas": {
      "$id": "7",
      "$values": [
        {
          "$id": "8",
          "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.MetadataModel, VecompSoftware.DocSuite.Public.Core.Models",
          "keyName": "NUMERO RDA",
          "value": "12345",
          "labelName": "",
          "archiveSection": ""
        }
      ]
    }
  }
},
  "id": "7ee38432-9fbc-4b20-90f0-a409e543879f",
  "name": "ArchiveModel"
}
```

WebApi Descrizione del modello – DocumentUnitModel

Modello Unità Documentaria Generica

- UniqueId: Identificativo univoco dell'unità documentale
- DocumentUnitType: Tipologia di unità documentale
 - Protocol = 1 (Protocollo)
 - Resolution = 2 (Atto/Delibera/Determina)
 - DocumentSeries = 4 (Serie documentale della Trasparenza Pubblica)
 - Archive = 8 (Unità documentaria specifica)

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "12",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.DocumentUnitModel, VecompSoftware.DocSuite.Public.Core.Models",
  "uniqueId": "8aeef4f8-0b91-4816-8bc1-4f562404cdb9",
  "year": 2020,
  "number": 10,
  "documentUnitName": "",
  "title": "",
  "subject": "Oggetto del protocollo",
  "direction": -1,
  "documentUnitType": 1
}
```

WebApi Descrizione del modello – MetadataModel

Modello del Metadata dinamico.

- KeyName: Nome del metadata
- Value: Valore del metadata
- MetadataId: Identificativo interno del metadata.
- LabelName: Etichetta di visualizzazione del metadata.
- ArchiveSection: Se l'unità documentaria è un archivio, è necessario specificare la sezione del metadata.

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "8",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.MetadataModel, VecompSoftware.DocSuite.Public.Core.Models",
  "keyName": "NUMERO RDA",
  "value": "12345",
  "labelName": "",
  "archiveSection": ""
}
```

WebApi Descrizione del modello – IdentityContext

Classe che contiene il contesto di sicurezza (es utente o ruoli dell'utente) del comando o dell'evento.

- Identity: Modello IdentityModel per la gestione dell'identità dell'utente.
- Roles: Collezione di modelli RoleModel che contiene i ruoli dell'utente

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{"$type": "VecompSoftware.DocSuite.Public.Core.Models.Securities.IdentityContext, VecompSoftware.DocSuite.Public.Core.Models",
  "roles": {
    "$id": "19",
    "$values": []
  },
  "identity": {
    "$id": "20",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Securities.IdentityModel, VecompSoftware.DocSuite.Public.Core.Models",
    "account": "MASTER_USL\\docsuite",
    "authorization": 0,
    "id": "126fc2f1-68cf-4cec-b360-ade2ca8b4905",
    "name": "MASTER_USL\\docsuite"
  }
}
```

WebApi Descrizione del modello – FascicleModel

Modello del fascicolo.

- Subject = Oggetto del fascicolo
- Note = Note del fascicolo
- Category = CategoryModel del Classificatore/Titolario del fascicolo
- Manager = ContactModel che identifica il responsabile di procedimento
- Sectors = Collezione di DocSuiteSectorModel dei settori autorizzati al fascicolo
- DocumentUnits = Collezione di DocumentUnitModel delle document unit da fascicolare

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
"$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.FascicleModel, VecompSoftware.DocSuite.Public.Core.Models",
"subject": "Oggetto del fascicolo",
"category": {
  "$id": "7",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.CategoryModel, VecompSoftware.DocSuite.Public.Core.Models",
  "uniqueId": "82faca3c-08c5-439d-940d-a2b11394d6fe",
  "name": "",
  "mappingTag": ""
},
"manager": {
  "$id": "8",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.ContactModel, VecompSoftware.DocSuite.Public.Core.Models",
  "contactId": "b7a73278-3635-4229-a2d9-3d2b4b274bbe",
  "archiveSection": "",
  "contactType": 64,
  "mappingTag": ""
},
"sectors": {
  "$id": "9",
  "$values": [
    {
      "$id": "10",
      "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.DocSuiteSectorModel, VecompSoftware.DocSuite.Public.Core.",
      "sectorRoleId": "05a03038-7623-47ea-9899-158eb70861f1",
      "name": "NOME DEL SETTORE AUTORIZZATO",
      "archiveSection": "",
      "mappingTag": ""
    }
  ]
}
}
```

WebApi Descrizione del modello - ContactModel

Modello Standard del Contatto Anagrafico

- **ContactType:** Tipologia del contatto
 - Administration = 1 (Amministrazione)
 - AOO = 2 (Area Organizzativa Omogenea (AOO))
 - AO = 4 (Unità Organizzativa (AO))
 - Role = 8 (Ruolo)
 - Group = 16 (Gruppo)
 - Sector = 32 (Settore)
 - Citizen = 64 (Persona)
 - IPA = 128 (Pubblica amministrazione da IPA)
- **ContactDirectionType:** Tipologia mittente/destinatario del contatto
 - Sender = 1 (Mittente)
 - Recipient = 2 (Destinatario)
- **ArchiveSection:** Se l'unità documentaria è un archivio, è necessario specificare la sezione del contatto
- **ContactId:** Codice univoco del contatto anagrafico della DocSuite
- **MappingTag:** Codice di mapping concordato con Dgroove
- **Description:** Denominazione
- **BirthDate:** Data di nascita nel formato "yyyy-MM-dd'T'HH:mm:ss.FFFK"
- **BirthPlace:** Luogo di nascita
- **ExternalCode:** Codice esterno del contatto
- **EmailAddress:** Indirizzo email
- **PECAAddress:** Indirizzo PEC
- **PhoneNumber:** Numero di telefono mobile
- **Address:** Indirizzo
- **City:** Città
- **ZipCode:** CAP
- **FiscalCode:** Codice Fiscale
- **FiscalTaxCode:** Partita Iva o Tax Code
- **TelephoneNumber:** Numero di telefono
- **FaxNumber:** Numero di FAX
- **Nationality:** Nazionalità (descrizione)
- **Note:** Campo note

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
    "$id": "17",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
    "parameterName": "dsw_p_archive_contact",
    "parameterModel": {
      "$id": "18",
      "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.ContactModel, VecompSoftware.DocSuite.Public.Core.Models",
      "archiveSection": "Fornitore",
      "description": "Dgroove Srl",
      "pecAddress": "info@pec.it",
      "address": "via Monte Baldo 6, Villafranca di Verona",
      "contactType": 1,
      "contactDirectionType": 2,
      "mappingTag": ""
    },
    "id": "3663414e-b745-4524-9932-ecdd65359e02",
    "name": "ContactModel"
  },
  ],
```

WebApi Descrizione del modello – DocumentModel

Modello del Documento

- DocumentName: Filename del documento
- Stream: Stream dati
- ArchiveSection: Se l'unità documentaria è un archivio, è necessario specificare anche l'identificativo o Label del documento
- DocumentType: Tipologia di documento
 - Main = 1 (Documento principale)
 - Attachment = 2 (Documento allegato)
 - Annexed = 4 (Documento annesso)

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "15",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.WorkflowParameterModel, VecompSoftware.DocSuite.Public.Core.Models",
  "parameterName": "dsw_p_protocol_attachment_document",
  "parameterModel": {
    "$id": "16",
    "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.DocumentModel, VecompSoftware.DocSuite.Public.Core.Models",
    "documentName": "documentoallegato.docx",
    "stream": {
      "$type": "System.Byte[], mscorlib",
      "$value": "AAE="
    },
    "documentType": 2,
    "archiveSection": ""
  },
  "id": "50541445-7b36-4a76-b1a8-52c47665ebff",
  "name": "DocumentModel"
},
],
```

WebApi Descrizione del modello – DocSuiteSectorModel

Modello del Settore / Ruolo configurato nella DocSuite

- SectorRoleId: Codice univoco del settore specifico DocSuite
- Name: Nome del settore specifico DocSuite
- ArchiveSection: Se l'unità documentaria è un archivio, è necessario specificare la sezione del settore
- MappingTag: E' possibile usare un codice di trascodifica da concordare con Dgroove

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
{
  "$id": "31",
  "$type": "VecompSoftware.DocSuite.Public.Core.Models.Workflows.Parameters.DocSuiteSectorModel, VecompSoftware.DocSuite.Public.Core.M
  "sectorRoleId": "56a8a8e9-a811-44b7-8bf6-2046e7cf8633",
  "name": "",
  "archiveSection": "",
  "mappingTag": ""
}
```

WebApi Descrizione del modello – IdentityModel

Modello per la gestione dell'identità dell'utente. Un buon esempio di utilizzo è passare il samAccountName nei contesti di autenticazione NTLM. Nei contesti di autenticazione OAuth si dovrebbe specificare l'accountId, o l'indirizzo email. In generale è necessario identifica univocamente un utente, quindi si deve passare un valore che permette di disambiguare le omonimie.

- Account: Riferimento all'identificativo di sicurezza. Es: samAccountName
- Email: Email dell'utente (da utilizzare in contesti di workflow che prevedono l'invio email)
- Name: Nome dell'utente (da intendersi come elemento che verrà visualizzato nelle pagine DocSuite, dove previst)
- Authorization: Modello astratto del contesto di sicurezza
 - External = 0 (Modello di autenticazione non integrato nella DocSuite)
 - NTLM = 1 (Autenticazione integrata di Windows)
 - DocSuiteSecurity = 2 (Sicurezza integrata della DocSuite (SecurityUser))
 - DocSuiteToken = 4 (Sicurezza OAuth - Owin DocSuite WebAPI Token Bearer)
 - OAuth = 8 (Sicurezza OAuth (non DocSuite ex: Google/Live))
 - JWT = 16 (Autenticazione JSON Web Token)
 - SPID = 32 (Autenticazione SPID)

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```
"$type": "VecompSoftware.DocSuite.Public.Core.Models.Securities.IdentityModel, VecompSoftware.DocSuite.Public.Core.Models",  
"account": "MASTER_USL\\docsuite",  
"authorization": 1,  
"id": "db99ee1a-b27e-4561-b111-4301a44f60c5",  
"name": "docsuite",  
"email": "noreply@master.org"
```

WebApi Elenco delle parole chiare usato nei modelli - WorkflowParameterNames

I parametri dei workflow utilizzano delle chiavi identificative "statiche". Questo elenco li racchiude e li descrive.

- Parametri di protocollo:
 - PROTOCOL_MODEL = "dsw_p_protocol_model" (Modello del protocollo)
 - MANAGE = "dsw_p_protocol_manage"
(Settore competente della richiesta di protocollazione)
 - REFERENCE = "dsw_p_protocol" (Riferimento)
 - MAIN_DOCUMENT = "dsw_p_protocol_main_document"
(Documento principale)
 - ATTACHMENT_DOCUMENT = "dsw_p_protocol_attachment_document"
(Documento allegato)
 - ANNEXED_DOCUMENT = "dsw_p_protocol_annexed_document"
(Documento annesso)
 - CATEGORY_MODEL = "dsw_p_protocol_category"
(Classificatore)
 - CONTAINER_MODEL = "dsw_p_protocol_container"
(Contenitore)
- Parametri della collaborazione:
 - COLLABORATION_MODEL = "dsw_p_collaboration_model" (Modello della collaborazione)
 - DOCUMENT_UNIT_MODEL = "dsw_p_collaboration_manage_model"
(Modello dell'unità documentale in gestione alla collaborazione)
 - MAIN_DOCUMENT = "dsw_p_collaboration_main_document"
(Documento principale)
 - ATTACHMENT_DOCUMENT = "dsw_p_collaboration_attachment_document"
(Documento allegato)
 - ANNEXED_DOCUMENT = "dsw_p_collaboration_annexed_document"
(Documento annesso)
 - SIGNER = "dsw_p_collaboration_signer"
(Firmatario)

- MANAGE = "dsw_p_collaboration_manage"
(Settore della segreteria di collaborazione)

- Parametri di archivio:
 - ARCHIVE_MODEL= "dsw_p_archive_model" (Modello dell'archivio)
 - MAIN_DOCUMENT = "dsw_p_archive_main_document"
(Documento principale)
 - ATTACHMENT_DOCUMENT = "dsw_p_archive_attachment_document"
(Documento allegato)
 - ANNEXED_DOCUMENT = "dsw_p_archive_annexed_document"
(Documento annesso)
 - CATEGORY_MODEL = "dsw_p_archive_category"
(Classificatore)
 - CONTACT_MODEL = "dsw_p_archive_container"
(Contenitore)

- Parametri di fascicolo:
 - FASCICLE_MODEL= "dsw_p_fascicle_model" (Modello del fascicolo)
 - ATTACHMENT_DOCUMENT = "dsw_p_fascicle_attachment_document"
(Inserito del fascicolo)
 - MANAGE = "dsw_p_fascicle_manage"
(Settore responsabile del fascicolo)

Risposta del controller "StartWorkflow" contenente la chiave univoca di avvio avvenuto con successo

In questa immagine viene mostrato il contenuto del payload della risposta HTTP con esito positivo. Il contenuto è la chiave in formato GUID che dovrà essere memorizzato dal fornitore per la corretta lettura degli eventi notificati dal servizio delle integrazioni.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 27 Jan 2021 15:25:51 GMT
Content-Length: 38

"0b4e1f6a-19a7-4fce-a145-c61ad01a4b12"
```

Eventi di risposta delle attività di workflow

In questa sezione sono descritti i modelli degli eventi che il microservizio delle integrazioni utilizzerà per invocare la WebAPI del fornitore. Come richiesto nei paragrafi precedenti è necessario e obbligatorio che l'API REST esponga un controller compatibile.

Questi eventi verranno inviati mediante chiamate al controller Rest mediante verbo POST del protocollo HTTP. Il Fornitore dovrà necessariamente comunicare a Dgroove l'endpoint del controller REST e per ogni workflow quali saranno i modelli, tra quelli sopra esposti, che andrà a specificare.

Mostriamo i vari modelli con relativa documentazione delle singole proprietà, descritta in meta linguaggio C#.

Il modello DocSuiteEvent:

```
public class DocSuiteEvent
{
    /// <summary>
    /// Identificativo univoco dell'evento
    /// es: F1345CE3-71B0-4D79-9E9A-57ABD0EF4978
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public Guid UniqueId { get; }
    /// <summary>
    /// Questa proprietà corrisponde all'istanza del workflow che ha generato l'evento.
    /// In workflow di tipo "Public" corrisponde al valore specificato nella proprietà "Id"
    /// del modello StartWorkflowCommand.
    /// es: D11EB32F-5386-4DC5-A0D1-CA0A55D834A7
    /// </summary>
    2 references | TFVC Import, 93 days ago | 1 author, 1 change
    public Guid? WorkflowReferenceId { get; set; }
    /// <summary>
    /// Modello contenente le informazioni del risultato del workflow.
    /// es: i riferimenti del protocollo creato.
    /// </summary>
    9 references | TFVC Import, 93 days ago | 1 author, 1 change
    public DocSuiteModel EventModel { get; set; }
    /// <summary>
    /// Modello contenente il riferimento dell'entità che ha generato l'attività.
    /// Ad esempio la gestione di una protocollazione del flusso di collaborazione conterrà
    /// nella proprietà EventModel il riferimento del protocollo appena creato, ma in ReferenceModel
    /// ci saranno gli estremi che identificano la collaborazione.
    /// Analogamente per il flusso di invio PEC di protocollo, i ReferenceModel conterrà i dati del protocollo
    /// e nell'EventModel ci saranno i dati della PEC.
    /// </summary>
    7 references | TFVC Import, 93 days ago | 1 author, 1 change
    public DocSuiteModel ReferenceModel { get; set; }
    /// <summary>
    /// Date afferente all'attività comunicata.
    /// es: se EventModel è di tipologia PEC sarà la data di invio della PEC
    /// se EventModel è di tipologia PROTOCOLLO sarà la data di creazione del Protocollo
    /// se EventModel è di tipologia COLLABORAZIONE sarà la data di creazione della Collaborazione
    /// </summary>
    2 references | TFVC Import, 93 days ago | 1 author, 1 change
    public DateTimeOffset EventDate { get; set; }
```

Il modello DocSuiteModel:

```
public class DocSuiteModel
{
    /// <summary>
    /// Rappresentazione testuale del modello
    /// es: Protocollo 004589 del 2017
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public string Title { get; set; }
    /// <summary>
    /// Anno di creazione dell'entità passata
    /// es: 2017
    /// NB: non tutti gli eventi contengono questo valore, ad esempio la PEC non ha anno
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public short? Year { get; set; }
    /// <summary>
    /// Numero di creazione dell'entità passata (dove prevista)
    /// es: 004589
    /// NB: non tutti gli eventi contengono questo valore, ad esempio la PEC non ha un numero
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public int? Number { get; set; }
    /// <summary>
    /// Identificativo univoco dell'entità passata (dove prevista)
    /// es: C892E9E6-0C67-43AE-90C4-0C36BE71E947
    /// </summary>
    4 references | TFVC Import, 93 days ago | 1 author, 1 change
    public Guid UniqueId { get; set; }
    /// <summary>
    /// Identificativo univoco dell'entità passata in forma numerica (dove prevista)
    /// es: 004589
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public int? EntityId { get; set; }
    /// <summary>
    /// Lista che conterrà proprietà specifiche/custom dell'integrazione che il fornitore potrà richiedere.
    /// Ogni richiesta dovrà essere valuta da Dgroove e se possibile verrà resa disponibile.
    /// </summary>
    5 references | TFVC Import, 93 days ago | 1 author, 1 change
    public IDictionary<string, string> CustomProperties { get; set; }
    /// <summary>
    /// Tipologia che determina a quale entità DocSuite si riferisce l'istanza del modello
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public DocSuiteType ModelType { get; set; }
    /// <summary>
    /// Stato dell'entità DocSuite
    /// </summary>
    0 references | TFVC Import, 93 days ago | 1 author, 1 change
    public DocSuiteStatus ModelStatus { get; set; }
}
```

Il modello DocSuiteType e DocSuiteStatus:

```
public enum DocSuiteType : short
{
    /// <summary>
    /// Non valido
    /// </summary>
    Invalid = 0,
    /// <summary>
    /// Protocollo
    /// </summary>
    Protocol = 1,
    /// <summary>
    /// PEC
    /// </summary>
    PEC = 2,
    /// <summary>
    /// Collaborazione di firma
    /// </summary>
    Collaboration = 3,
    /// <summary>
    /// Archivio/UDS
    /// </summary>
    UDS = 4,
    /// <summary>
    /// Fascicolo
    /// </summary>
    Fascicle = 5,
    /// <summary>
    /// Dossier
    /// </summary>
    Dossier = 6
}

public enum DocSuiteStatus : short
{
    /// <summary>
    /// Non valido
    /// </summary>
    Invalid = 0,
    /// <summary>
    /// Attivo
    /// </summary>
    Activated = 1,
    /// <summary>
    /// Anullato
    /// </summary>
    Canceled = 2,
    /// <summary>
    /// Inviato
    /// </summary>
    Sended = 4,
    /// <summary>
    /// Ricevuto
    /// </summary>
    Received = 8,
    /// <summary>
    /// Rifiutato
    /// </summary>
    Rejected = 16,
}
```

Tabella dei modelli degli eventi di risposta ai workflow più usati

Evento	Identificazione evento	Informazioni comunicate
Protocollazione	EventModel: ModelType=Protocol ModelStatus=Activated	EventModel: "Year"=anno di protocollazione "Number"=numero di protocollazione ReferenceModel: "EntityID"=numero collaborazione Event: "EventDate"=data protocollazione
Annulla protocollo	EventModel: ModelType=Protocol ModelStatus=Canceled	EventModel: "Year"=anno di protocollazione "Number"=numero di protocollazione Event: "EventDate"=data annullamento protocollo
Ricezione PEC da parte del destinatario	EventModel: ModelType=PEC ModelStatus=Received	EventModel: "CustomProperties[Sender]" = mittente della PEC ReferenceModel: "Year"=anno di protocollazione "Number"=numero di protocollazione Event: "EventDate"=data ricezione PEC

Mostriamo un esempio semplificato di come viene rappresentato in formato Json.

```

{
  "UniqueId": "d0064306-96b5-404f-af04-5e0d1510f6c0",
  "WorkflowReferenceId": "1e7190f8-5a8c-4e1e-90f7-5eb759ce9c3a",
  "EventModel": {
    "Title": "Protocollo 2017/004589",
    "Year": 2017,
    "Number": 4589,
    "UniqueId": "51fe20c4-69ee-4c7d-8b36-a25cefd5d275",
    "EntityId": null,
    "ModelType": 1,
    "ModelStatus": 1
  },
  "ReferenceModel": {
    "Title": "Collaborazione n° 157",
    "Year": null,
    "Number": null,
    "UniqueId": "6e80266f-ce5e-4cfc-a101-2cd2137f0b78",
    "EntityId": 157,
    "ModelType": 3,
    "ModelStatus": 1
  },
  "EventDate": "2017-12-11T21:56:05.0898462+00:00"
}

```

6. Interfacce URL di apertura DocSuite mediante Browser Web

La modalità non prevede architetture particolari, se non sfruttare la modalità di parametri in querystring previsti dal protocollo HTTP.

La modalità di autenticazione è basata sulla windows autenticazione (NTLM) come da procedura standard DocSuite. L'utente che effettua l'accesso deve essere preventivamente già configurato correttamente negli specifici settori e/o contenitori DocSuite. Non sono previste altre modalità extra procedura non standard.

Modulo "Protocollo"

Apertura sommario di un protocollo mediante Anno e Numero

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: prot	Si
Anno	Dinamico di tipo stringa, specificare anno del protocollo	Si
Numero	Dinamico di tipo stringa, specificare numero del protocollo	Si
TenantAOO	Dinamico di tipo stringa, specificare nome della AOO di pertinenza del protocollo. Se non specificato verrà preso quello di default configurato per l'utente	No

Mostriamo un esempio di url:

[http://\(Url produzione\)?Azione=Apri&Anno=2020&Numero=1&Tipo=prot](http://(Url produzione)?Azione=Apri&Anno=2020&Numero=1&Tipo=prot)

Apertura visualizzazioni documenti di protocollo mediante Anno e Numero

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
OpenDocument	Statico: Yes	Si
Tipo	Statico: prot	Si
Anno	Dinamico di tipo stringa, specificare anno del protocollo	Si
Numero	Dinamico di tipo stringa, specificare numero del protocollo	Si
TenantAOO	Dinamico di tipo stringa, specificare nome della AOO di pertinenza del protocollo. Se non specificato verrà preso quello di default configurato per l'utente	No

Mostriamo un esempio di url:

[http://\(Url produzione\)?Azione=Apri&Anno=2020&Numero=1&OpenDocument=Yes&Tipo=prot](http://(Url produzione)?Azione=Apri&Anno=2020&Numero=1&OpenDocument=Yes&Tipo=prot)

Modulo "Dossier"

Apertura sommario dossier mediante parametri IdDossier (identificavo univoco)

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: dossier	Si
IdDossier	Dinamico di tipo stringa, specificare anno del dossier	Si

Mostriamo un esempio di url:

[http://\(url produzione\)?Azione=Apri&IdDossier=9D96B31D-6FDE-4124-84EA-10540496A02F&Tipo=dossier](http://(url produzione)?Azione=Apri&IdDossier=9D96B31D-6FDE-4124-84EA-10540496A02F&Tipo=dossier)

Apertura sommario dossier mediante parametri Anno e Numero

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: dossier	Si
Anno	Dinamico di tipo stringa, specificare anno del dossier	Si
Numero	Dinamico di tipo stringa, specificare numero del dossier	Si

Mostriamo un esempio di url:

[http://\(url produzione\)?Azione=Apri&Anno=2020&Numero=1&Tipo=dossier](http://(url produzione)?Azione=Apri&Anno=2020&Numero=1&Tipo=dossier)

Apertura sommario dossier mediante parametro metadato specifico del dossier

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: dossier	Si
Metadato	Dinamico di tipo stringa, specificare il nome del metadato presente e valorizzato nello specifico dossier che si vuole aprire	Si
Valore	Dinamico di tipo stringa, specificare il valore del metadato presente e valorizzato nello specifico dossier che si vuole aprire	Si
Anno	Dinamico di tipo stringa, se specificato effettua un ulteriore filtro sull'anno di creazione del dossier	Opzionale

Mostriamo un esempio di url:

[http://\(url produzione\)?Azione=Apri&Metadato=CodicePersonale&Valore=ABCD&Anno=2020&Tipo=dossier](http://(url produzione)?Azione=Apri&Metadato=CodicePersonale&Valore=ABCD&Anno=2020&Tipo=dossier)

Apertura sommario dossier e relativo fascicolo mediante IdDossier (identificativo univoco) e IdFascicle (identificativo univoco del fascicolo)

Chiave	Valore	Obbligatorietà
Azione	Statico : ApriConFascicolo	Si
Tipo	Statico : dossier	Si
IdDossier	Dinamico di tipo stringa, specificare identificativo del dossier	Si
IdFascicle	Dinamico di tipo stringa, specificare identificativo del fascicolo	Si

Mostriamo un esempio di url: [http://\(url produzione\)?Azione=Apri&IdDossier=9D96B31D-6FDE-4124-84EA-10540496A02F&IdFascicle=C49A9CE0-157D-4D52-85FC-F934BA4DF9A5&Tipo=dossier](http://(url produzione)?Azione=Apri&IdDossier=9D96B31D-6FDE-4124-84EA-10540496A02F&IdFascicle=C49A9CE0-157D-4D52-85FC-F934BA4DF9A5&Tipo=dossier)

Apertura sommario dossier mediante parametro metadato specifico del dossier e apertura cartellinatura specifica

Chiave	Valore	Obbligatorietà
Azione	Statico: ApriConFascicolo	Si
Tipo	Statico: dossier	Si
Metadato	Dinamico di tipo stringa, specificare il nome del metadato presente e valorizzato nello specifico dossier che si vuole aprire	Si
Valore	Dinamico di tipo stringa, specificare il valore del metadato presente e valorizzato nello specifico dossier che si vuole aprire	Si
Fascicolo	Dinamico di tipo stringa, specificare l'oggetto del fascicolo. Se nello stesso dossier sono presenti più fascicolo con lo stesso oggetto, verrà aperto l'ultimo in ordine cronologico (più recente).	Si
Anno	Dinamico di tipo stringa, se specificato effettua un ulteriore filtro sull'anno di creazione del dossier	Opzionale

Mostriamo un esempio di url:

[http://\(urlproduzione\)?Azione=ApriConFascicolo&Metadato=Matricola&Valore=ABCD&Fascicolo=Pensioni&Tipo=dossier](http://(urlproduzione)?Azione=ApriConFascicolo&Metadato=Matricola&Valore=ABCD&Fascicolo=Pensioni&Tipo=dossier)

Modulo "Fascicolo"

Apertura sommario fascicolo mediante parametri IdFascicle (identificavo univoco)

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: fasc	Si
IdFascicle	Dinamico di tipo stringa, specificare identificativo del fascicolo	Si

Mostriamo un esempio di url:

[http://\(url produzione\)?Azione=Apri&IdFascicle=C49A9CE0-157D-4D52-85FC-F934BA4DF9A5&Tipo=fasc](http://(url produzione)?Azione=Apri&IdFascicle=C49A9CE0-157D-4D52-85FC-F934BA4DF9A5&Tipo=fasc)

Apertura sommario fascicolo mediante parametri Anno e Numero

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: fasc	Si
Anno	Dinamico di tipo stringa, specificare anno del fascicolo	Si
Numero	Dinamico di tipo stringa, specificare numero del fascicolo	Si

Mostriamo un esempio di url:

[http://\(url produzione\)?Azione=Apri&Anno=2020&Numero=20.3.2-0000004&Tipo=fasc](http://(url produzione)?Azione=Apri&Anno=2020&Numero=20.3.2-0000004&Tipo=fasc)

Apertura sommario fascicolo mediante parametro metadato specifico del fascicolo

Chiave	Valore	Obbligatorietà
Azione	Statico: Apri	Si
Tipo	Statico: fasc	Si
Metadato	Dinamico di tipo stringa, specificare il nome del metadato presente e valorizzato nello specifico fascicolo che si vuole aprire	Si
Valore	Dinamico di tipo stringa, specificare il valore del metadato presente e valorizzato nello specifico fascicolo che si vuole aprire	Si
Anno	Dinamico di tipo stringa, se specificato effettua un ulteriore filtro sull'anno di creazione del dossier	Opzionale

Mostriamo un esempio di url:

[http://\(url produzione\)?Azione=Apri&Metadato=Matricola&Valore=12345&Anno=2020&Tipo=fasc](http://(url produzione)?Azione=Apri&Metadato=Matricola&Valore=12345&Anno=2020&Tipo=fasc)

7. Esempi dei modelli JSON di Avvio workflow ed eventi di risposta

Esempio 1 - Linguaggio C# - Avvio Workflow Firma digitale con protocollazione

Mostriamo un esempio generico di un avvio di un workflow (preconfigurato nella DocSuite) chiamato "Firma digitale con protocollazione". L'esempio evidenzia le strutture e i modelli necessari scritto nel linguaggio C#, che un generico fornitore potrebbe riutilizzare per comprendere le strutture.

```
///Modello che specifica l'Identità dell'utente che sta eseguendo l'avvio del Workflow
IdentityModel identityModel = new IdentityModel(accountName, AuthorizationType.External);
IdentityContext identityContext = new IdentityContext(identityModel);

/// Modello che permette l'avvio del Workflow "Firma digitale con protocollazione"
WorkflowModel workflowModel = new WorkflowModel("Firma digitale con protocollazione");

///Modello della collaborazione
CollaborationModel collaborationModel = new CollaborationModel("Richiesta firma digitale", "campo note di collaborazione", DocumentUnitDirection.Outgoing, DocumentUnitType.Protocol);
WorkflowParameterModel workflowParameterModel12 = new WorkflowParameterModel(collaborationModel, WorkflowParameterNames.CollaborationNames.COLLABORATION_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel12);

///Modello della firmatario
SignerModel signerModel01 = new SignerModel(1, true, SignerType.AD, new IdentityModel("<utente firmatario nel formato dominio\\samAccount>", AuthorizationType.NTLM));
WorkflowParameterModel workflowParameterModel13 = new WorkflowParameterModel(signerModel01, WorkflowParameterNames.CollaborationNames.SIGNER);
workflowModel.WorkflowParameters.Add(workflowParameterModel13);

///Modello della firmatario
SignerModel signerModel02 = new SignerModel(2, true, SignerType.AD, new IdentityModel("<utente firmatario nel formato dominio\\samAccount>", AuthorizationType.NTLM));
WorkflowParameterModel workflowParameterModel14 = new WorkflowParameterModel(signerModel02, WorkflowParameterNames.CollaborationNames.SIGNER);
workflowModel.WorkflowParameters.Add(workflowParameterModel14);

///modello che determina la segreteria di collaborazione
DocSuiteSectorModel docSuiteSectorModel = new DocSuiteSectorModel("VALORE DA CONCORDARE", mappingTag: "VALORE DA CONCORDARE");
WorkflowParameterModel workflowParameterModel11 = new WorkflowParameterModel(docSuiteSectorModel, WorkflowParameterNames.CollaborationNames.MANAGE);
workflowModel.WorkflowParameters.Add(workflowParameterModel11);

///Documento principale della collaborazione
DocumentModel collaborationDocument = new DocumentModel("documentoprincipale.docx", new byte[] { 0x00, 0x01 }, DocumentType.Main);
WorkflowParameterModel workflowParameterModel15 = new WorkflowParameterModel(collaborationDocument, WorkflowParameterNames.CollaborationNames.MAIN_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel15);

///Modello del protocollo
CollaborationModel collaborationToProtocolModel = new CollaborationModel("Prima protocollazione di esempio", "node della collaborazione",
    DocumentUnitDirection.Outgoing, DocumentUnitType.Protocol,
    category: new CategoryModel(uniqueId: Guid.Parse("1B9586C9-EA83-45DF-8816-E05F058AD0CA")),
    container: new ContainerModel(uniqueId: Guid.Parse("0A0E03B7-E459-41A1-B64D-94405AEC0B6C")),
    collaborationPriority: CollaborationPriorityType.Normal);
collaborationToProtocolModel.Contacts.Add(new ContactModel(ContactType.Administration, ContactDirectionType.Recipient)
{
    Description = "Dgroove Srl",
    Address = "via Monte Baldo 6, Villafranca di Verona",
    PECAAddress = "info@pec.it"
});
collaborationToProtocolModel.Contacts.Add(new ContactModel(ContactType.AOO, ContactDirectionType.Sender, contactId: Guid.Parse("1B9586C9-EA83-45DF-8816-E05F058AD0CA")));
collaborationToProtocolModel.Contacts.Add(new ContactModel(ContactType.AOO, ContactDirectionType.Recipient) { Description = "specificare descrizione contatto non di rubrica" });
collaborationToProtocolModel.Sectors.Add(new DocSuiteSectorModel(string.Empty, sectorRoleId: Guid.Parse("56A8A8E9-A811-44B7-8BF6-2046E7CF8633")));
collaborationToProtocolModel.Metadata.Add(new MetadataModel("Esempio di metadato dinamico", "valore di esempio"));

WorkflowParameterModel workflowParameterModel16 = new WorkflowParameterModel(collaborationToProtocolModel, WorkflowParameterNames.CollaborationNames.DOCUMENT_UNIT_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel16);

/// Modello o busta del comando di avvio workflow. Questo contiene il modello sopra definito
StartWorkflowContentType startWorkflowContentType = new StartWorkflowContentType(workflowModel, accountName, Guid.NewGuid());
```

Esempio 2 – Linguaggio C# - Avvio Workflow Protocollo manuale

Mostriamo un esempio generico di un avvio di un workflow (preconfigurato nella DocSuite) chiamato “Protocollo semplice”. L’esempio evidenzia le strutture e i modelli necessari scritto nel linguaggio C#, che un generico fornitore potrebbe riutilizzare per comprendere le strutture.

```
//Modello che specifica l'Identità dell'utente che sta eseguendo l'avvio del Workflow
IdentityModel identityModel = new IdentityModel(accountName, AuthorizationType.External);
IdentityContext identityContext = new IdentityContext(identityModel);

/// Modello che permette l'avvio del Workflow "Protocollo Semplice"
WorkflowModel workflowModel = new WorkflowModel("Protocollo Semplice");

DocSuiteSectorModel docSuiteSectorModel = new DocSuiteSectorModel("VALORE DA CONCORDARE", mappingTag: "VALORE DA CONCORDARE");
WorkflowParameterModel workflowParameterModel11 = new WorkflowParameterModel(docSuiteSectorModel, WorkflowParameterNames.ProtocolNames.MANAGE);
workflowModel.WorkflowParameters.Add(workflowParameterModel11);

///Modello del protocollo
CollaborationModel collaborationModel = new CollaborationModel("Prima protocollazione di esempio", string.Empty,
    DocumentUnitDirection.Outgoing, DocumentUnitType.Protocol);
collaborationModel.Contacts.Add(new ContactModel(ContactType.Administration, ContactDirectionType.Recipient)
{
    Description = "Dgroove Srl",
    Address = "via Monte Baldo 6, Villafranca di Verona",
    PECAddress = "info@pec.it"
});
collaborationModel.Contacts.Add(new ContactModel(ContactType.AOO, ContactDirectionType.Sender, contactId: Guid.Parse("1B9586C9-EA83-45DF-8816-E05F058AD0CA")));
collaborationModel.Sectors.Add(new DocSuiteSectorModel(string.Empty, sectorRoleId: Guid.Parse("56A8A8E9-A811-44B7-8BF6-2046E7CF8633")));
collaborationModel.Metadata.Add(new MetadataModel("Esempio di metadato dinamico", "valore di esempio"));

WorkflowParameterModel workflowParameterModel12 = new WorkflowParameterModel(collaborationModel, WorkflowParameterNames.ProtocolNames.PROTOCOL_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel12);

///Documento principale del protocollo
DocumentModel collaborationDocument = new DocumentModel("documentoprincipale.docx", new byte[] { 0x00, 0x01 }, DocumentType.Main);
WorkflowParameterModel workflowParameterModel13 = new WorkflowParameterModel(collaborationDocument, WorkflowParameterNames.ProtocolNames.MAIN_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel13);
DocumentModel collaborationAttachment = new DocumentModel("documentoallegato.docx", new byte[] { 0x00, 0x01 }, DocumentType.Attachment);
WorkflowParameterModel workflowParameterModel14 = new WorkflowParameterModel(collaborationAttachment, WorkflowParameterNames.ProtocolNames.ATTACHMENT_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel14);

///Classificatore
CategoryModel categoryModel = new CategoryModel(uniqueId: Guid.Parse("37579C9E-8883-49AD-BF64-5BD95E64E0E1"));
WorkflowParameterModel workflowParameterModel15 = new WorkflowParameterModel(categoryModel, WorkflowParameterNames.ProtocolNames.CATEGORY_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel15);

///Contenitore
ContainerModel containerModel = new ContainerModel(uniqueId: Guid.Parse("480F483B-A6A3-43DE-BA9D-149815E518F1"));
WorkflowParameterModel workflowParameterModel16 = new WorkflowParameterModel(containerModel, WorkflowParameterNames.ProtocolNames.CONTAINER_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel16);

/// Modello o busta del comando di avvio workflow. Questo contiene il modello sopra definito
StartWorkflowContentType startWorkflowContentType = new StartWorkflowContentType(workflowModel, accountName, Guid.NewGuid());

/// Creazione del comando di start. Nell'esempio si potrebbe sostituire Guid.NewGuid() con una chiave esterna che valore per il fornitore.
/// Utile per agganciare i messaggi di fine processo.
StartWorkflowCommand startWorkflowCommand = new StartWorkflowCommand(Guid.NewGuid(), tenantName, tenantId, tenantAooId, identityContext, startWorkflowContentType);
```

Esempio 3 – Linguaggio C# - Avvio Workflow Protocollazione automatica

Mostriamo un esempio generico di un avvio di un workflow (preconfigurato nella DocSuite) chiamato “Protocollazione automatica”. L’esempio evidenzia le strutture e i modelli necessari scritto nel linguaggio C#, che un generico fornitore potrebbe riutilizzare per comprendere le strutture.

```
//Modello che specifica l'Identità dell'utente che sta eseguendo l'avvio del Workflow
IdentityModel identityModel = new IdentityModel(accountName, AuthorizationType.External);
IdentityContext identityContext = new IdentityContext(identityModel);

/// Modello che permette l'avvio del workflow "Protocollazione automatica"
WorkflowModel workflowModel = new WorkflowModel("Protocollazione automatica");

///Modello del protocollo
CollaborationModel collaborationModel = new CollaborationModel("Prima protocollazione di esempio", string.Empty,
    DocumentUnitDirection.Outgoing, DocumentUnitType.Protocol);
collaborationModel.Contacts.Add(new ContactModel(ContactType.Administration, ContactDirectionType.Recipient)
{
    Description = "Dgroove Srl",
    Address = "via Monte Baldo 6, Villafranca di Verona",
    PECAddress = "info@pec.it"
});
collaborationModel.Contacts.Add(new ContactModel(ContactType.AOO, ContactDirectionType.Sender, contactId: Guid.Parse("189586C9-EA83-45DF-8816-E05F058AD0CA"));
collaborationModel.Sectors.Add(new DocSuiteSectorModel(string.Empty, sectorRoleId: Guid.Parse("56A8ABE9-A811-4487-8BF6-2046E7CF8633")));
WorkflowParameterModel workflowParameterModel2 = new WorkflowParameterModel(collaborationModel, WorkflowParameterNames.ProtocolNames.PROTOCOL_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel2);

///Documento principale del protocollo
DocumentModel collaborationDocument = new DocumentModel("documentoprincipale.docx", new byte[] { 0x00, 0x01 }, DocumentType.Main);
WorkflowParameterModel workflowParameterModel3 = new WorkflowParameterModel(collaborationDocument, WorkflowParameterNames.ProtocolNames.MAIN_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel3);
DocumentModel collaborationAttachment = new DocumentModel("documentoallegato.docx", new byte[] { 0x00, 0x01 }, DocumentType.Attachment);
WorkflowParameterModel workflowParameterModel4 = new WorkflowParameterModel(collaborationAttachment, WorkflowParameterNames.ProtocolNames.ATTACHMENT_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel4);

///Classificatore
CategoryModel categoryModel = new CategoryModel(uniqueId: Guid.Parse("37579C9E-8883-49AD-8F64-58D95E64E0E1"));
WorkflowParameterModel workflowParameterModel5 = new WorkflowParameterModel(categoryModel, WorkflowParameterNames.ProtocolNames.CATEGORY_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel5);

///Contenitore
ContainerModel containerModel = new ContainerModel(uniqueId: Guid.Parse("480F483B-A6A3-43DE-BA9D-149815E518F1"));
WorkflowParameterModel workflowParameterModel6 = new WorkflowParameterModel(containerModel, WorkflowParameterNames.ProtocolNames.CONTAINER_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel6);

/// Modello o busta del comando di avvio workflow. Questo contiene il modello sopra definito
StartWorkflowContentType startWorkflowContentType = new StartWorkflowContentType(workflowModel, accountName, Guid.NewGuid());

/// Creazione del comando di start. Nell'esempio si potrebbe sostituire Guid.NewGuid() con una chiave esterna che valore per il fornitore.
/// Utile per agganciare i messaggi di fine processo.
StartWorkflowCommand startWorkflowCommand = new StartWorkflowCommand(Guid.NewGuid(), tenantName, tenantId, tenantAooId, identityContext, startWorkflowContentType);
```

Esempio 4 – Linguaggio C# - Avvio Workflow crea archivio

Mostriamo un esempio generico di un avvio di un workflow (preconfigurato nella DocSuite) chiamato “Crea archivio”. L’esempio evidenzia le strutture e i modelli necessari scritto nel linguaggio C#, che un generico fornitore potrebbe riutilizzare per comprendere le strutture.

```
//Modello che specifica l'Identità dell'utente che sta eseguendo l'avvio del Workflow
IdentityModel identityModel = new IdentityModel(accountName, AuthorizationType.External);
IdentityContext identityContext = new IdentityContext(identityModel);

/// Modello che permette l'avvio del Workflow "Crea contratto"
WorkflowModel workflowModel = new WorkflowModel("Crea contratto");

///Modello dell'archivio - il nome "Contratti" è un esempio ma in produzione va concordato
ArchiveModel archiveModel = new ArchiveModel("Contratti", archiveId: Guid.Parse("5A9586C9-EA83-45DF-8816-E05F058AD081"));
/// l'elenco dei metadati specifici dell'archivio sono da concordare
archiveModel.Metadata.Add(new MetadataModel("NUMERO RDA", "12345"));
archiveModel.Metadata.Add(new MetadataModel("NUMERO ODA", "K123"));
archiveModel.Metadata.Add(new MetadataModel("CIG", "Z123456"));
WorkflowParameterModel workflowParameterModel2 = new WorkflowParameterModel(archiveModel, WorkflowParameterNames.ArchiveNames.ARCHIVE_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel2);

///Documento principale del protocollo
DocumentModel collaborationDocument = new DocumentModel("documentoprincipale.docx", new byte[] { 0x00, 0x01 }, DocumentType.Main);
WorkflowParameterModel workflowParameterModel3 = new WorkflowParameterModel(collaborationDocument, WorkflowParameterNames.ArchiveNames.MAIN_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel3);
DocumentModel collaborationAttachment = new DocumentModel("documentoallegato.docx", new byte[] { 0x00, 0x01 }, DocumentType.Attachment);
WorkflowParameterModel workflowParameterModel4 = new WorkflowParameterModel(collaborationAttachment, WorkflowParameterNames.ArchiveNames.ATTACHMENT_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel4);

///Classificatore
CategoryModel categoryModel = new CategoryModel(uniqueId: Guid.Parse("37579C9E-8883-49AD-BF64-5B095E64E0E1"));
WorkflowParameterModel workflowParameterModel5 = new WorkflowParameterModel(categoryModel, WorkflowParameterNames.ArchiveNames.CATEGORY_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel5);

///Contatto - archivesection è obbligatoria e da concordare
ContactModel contactModel = new ContactModel(ContactType.Administration, ContactDirectionType.Recipient, archiveSection: "Fornitore")
{
    Description = "Dgroove Srl",
    Address = "via Monte Baldo 6, Villafranca di Verona",
    PECAddress = "info@pec.it"
};
WorkflowParameterModel workflowParameterModel6 = new WorkflowParameterModel(contactModel, WorkflowParameterNames.ArchiveNames.CONTACT);
workflowModel.WorkflowParameters.Add(workflowParameterModel6);

/// Modello o busta del comando di avvio workflow. Questo contiene il modello sopra definito
StartWorkflowContentType startWorkflowContentType = new StartWorkflowContentType(workflowModel, accountName, Guid.NewGuid());

/// Utile per agganciare i messaggi di fine processo.
StartWorkflowCommand startWorkflowCommand = new StartWorkflowCommand(Guid.NewGuid(), tenantName, tenantId, tenantAooId, identityContext, startWorkflowContentType);
```

Esempio 5 – Linguaggio C# - Avvio Workflow Invia PEC automatica

Mostriamo un esempio generico di un avvio di un workflow (preconfigurato nella DocSuite) chiamato “Invia PEC automatica”. L’esempio evidenzia le strutture e i modelli necessari scritto nel linguaggio C#, che un generico fornitore potrebbe riutilizzare per comprendere le strutture.

```
//Modello che specifica l'Identità dell'utente che sta eseguendo l'avvio del Workflow
IdentityModel identityModel = new IdentityModel(accountName, AuthorizationType.External);
IdentityContext identityContext = new IdentityContext(identityModel);

/// Modello che permette l'avvio del Workflow "Invio PEC"
WorkflowModel workflowModel = new WorkflowModel("Invia PEC automatica");

///Modello del protocollo
PECMailModel pecmailModel = new PECMailModel("oggetto della pec", "messaggio della PEC", "mittente@dominio.it", "destinatario@dominio.it", "destinatarioccc@dominio.it");
WorkflowParameterModel workflowParameterModel2 = new WorkflowParameterModel(pecmailModel, WorkflowParameterNames.PECMAIL_NAMES.PECMAIL_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel2);

///Documento principale del protocollo
DocumentModel collaborationDocument = new DocumentModel("documentoprincipale.docx", new byte[] { 0x00, 0x01 }, DocumentType.Main);
WorkflowParameterModel workflowParameterModel3 = new WorkflowParameterModel(collaborationDocument, WorkflowParameterNames.PECMAIL_NAMES.MAIN_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel3);

/// Modello o busta del comando di avvio workflow. Questo contiene il modello sopra definito
StartWorkflowContentType startWorkflowContentType = new StartWorkflowContentType(workflowModel, accountName, Guid.NewGuid());

/// Creazione del comando di start. Nell'esempio si potrebbe sostituire Guid.NewGuid() con una chiave esterna che valore per il fornitore.
/// Utile per agganciare i messaggi di fine processo.
StartWorkflowCommand startWorkflowCommand = new StartWorkflowCommand(Guid.NewGuid(), tenantName, tenantId, tenantAooId, identityContext, startWorkflowContentType);
```

Esempio 6 – Linguaggio C# - Avvio Workflow Creazione Fascicolo

Mostriamo un esempio generico di un avvio di un workflow (preconfigurato nella DocSuite) chiamato "Crea Fascicolo". L'esempio evidenzia le strutture e i modelli necessari scritto nel linguaggio C#, che un generico fornitore potrebbe riutilizzare per comprendere le strutture.

```
//Modello che specifica l'Identità dell'utente che sta eseguendo l'avvio del Workflow
IdentityModel identityModel = new IdentityModel(accountName, AuthorizationType.External);
IdentityContext identityContext = new IdentityContext(identityModel);

/// Modello che permette l'avvio del Workflow "Crea Fascicolo"
WorkflowModel workflowModel = new WorkflowModel("Crea Fascicolo");

///Modello dell'fascicolo
FascicleModel fascicleModel = new FascicleModel("Oggetto del fascicolo",
//Classificatore del fascicolo
new CategoryModel(uniqueId: Guid.Parse("82FACA3C-08C5-439D-940D-A2B11394D6FE")),
//Contatto responsabile di procedimento
new ContactModel(ContactType.Citizen, contactId: Guid.Parse("B7A73278-3635-4229-A2D9-3D28482748BE")));
//Elenco delle document units (protocolli)
fascicleModel.DocumentUnits.Add(new DocumentUnitModel(Guid.Parse("8AEFF4F8-0891-4816-8BC1-4F562404CDB9"), 0, 0, string.Empty,
string.Empty, null, DocumentUnitDirection.Inbound, DocumentUnitType.Protocol, string.Empty));
fascicleModel.DocumentUnits.Add(new DocumentUnitModel(Guid.Parse("FEB130F8-DBC7-4F9B-A29F-DF59F7EE8353"), 0, 0, string.Empty,
string.Empty, null, DocumentUnitDirection.Inbound, DocumentUnitType.Protocol, string.Empty));
//Elenco dei settori autorizzati al fascicolo
fascicleModel.Sectors.Add(new DocSuiteSectorModel("NOME DEL SETTORE AUTORIZZATO", sectorRoleId: Guid.Parse("05A03038-7623-47EA-9899-158EB70861F1")));
WorkflowParameterModel workflowParameterModel = new WorkflowParameterModel(fascicleModel, WorkflowParameterNames.FASCICLE_NAMES.FASCICLE_MODEL);
workflowModel.WorkflowParameters.Add(workflowParameterModel);

///Settore responsabile in competenza del fascicolo
DocSuiteSectorModel docSuiteSectorModel = new DocSuiteSectorModel("VALORE DA CONCORDARE", mappingTag: "VALORE DA CONCORDARE");
WorkflowParameterModel workflowParameterModel1 = new WorkflowParameterModel(docSuiteSectorModel, WorkflowParameterNames.FASCICLE_NAMES.MANAGE);
workflowModel.WorkflowParameters.Add(workflowParameterModel1);

///Documento principale del protocollo
DocumentModel collaborationDocument = new DocumentModel("documentoprincipale.docx", new byte[] { 0x00, 0x01 }, DocumentType.Main);
WorkflowParameterModel workflowParameterModel2 = new WorkflowParameterModel(collaborationDocument, WorkflowParameterNames.FASCICLE_NAMES.MAIN_DOCUMENT);
workflowModel.WorkflowParameters.Add(workflowParameterModel2);

/// Modello o busta del comando di avvio workflow. Questo contiene il modello sopra definito
StartWorkflowContentType startWorkflowContentType = new StartWorkflowContentType(workflowModel, accountName, Guid.NewGuid());
```

Esempio 7 – Modello JSON - Evento “Protocollo creato” a fronte di un workflow “Protocollo documento”.

```
{
  "UniqueId": "d0064306-96b5-404f-af04-5e0d1510f6c0",
  "WorkflowReferenceId": "1e7190f8-5a8c-4e1e-90f7-5eb759ce9c3a",
  "EventModel": {
    "Title": "Protocollo 2017/004589",
    "Year": 2017,
    "Number": 4589,
    "UniqueId": "51fe20c4-69ee-4c7d-8b36-a25cefd5d275",
    "EntityId": null,
    "ModelType": 1,
    "ModelStatus": 1
  },
  "EventDate": "2017-12-11T21:56:05.0898462+00:00"
}
```

Esempio 8 – Modello JSON - Evento “Protocollo creato” a fronte di un workflow “Collaborazione firma digitale”.

```
{
  "UniqueId": "d0064306-96b5-404f-af04-5e0d1510f6c0",
  "WorkflowReferenceId": "1e7190f8-5a8c-4e1e-90f7-5eb759ce9c3a",
  "EventModel": {
    "Title": "Protocollo 2017/004589",
    "Year": 2017,
    "Number": 4589,
    "UniqueId": "51fe20c4-69ee-4c7d-8b36-a25cefd5d275",
    "EntityId": null,
    "ModelType": 1,
    "ModelStatus": 1
  },
  "ReferenceModel": {
    "Title": "Collaborazione n° 157",
    "Year": null,
    "Number": null,
    "UniqueId": "6e80266f-ce5e-4cfc-a101-2cd2137f0b78",
    "EntityId": 157,
    "ModelType": 3,
    "ModelStatus": 1
  },
  "EventDate": "2017-12-11T21:56:05.0898462+00:00"}
}
```

Esempio 9 – Modello JSON - Evento “PEC ricevuta e associata al protocollo” a fronte di un workflow “Collaborazione firma digitale”.

```
{
  "UniqueId": "50e249b1-3910-42e8-88c2-a3bf17048b8a",
  "WorkflowReferenceId": "d4fce45e-ba38-462a-aef0-f77ca79757d1",
  "EventModel": {
    "Title": "PEC 1597",
    "Year": null,
    "Number": null,
    "UniqueId": "fcd723f9-ce5b-4e21-93ca-7cfe85524a72",
    "EntityId": 1597,
    "ModelType": 2,
    "ModelStatus": 4
  },
  "ReferenceModel": {
    "Title": "Protocollo 2017/004589",
    "Year": 2017,
    "Number": 4589,
    "UniqueId": "58958516-3d23-4f3a-a413-0bbd8d3361c7",
    "EntityId": null,
    "ModelType": 1,
    "ModelStatus": 1
  },
  "EventDate": "2017-12-11T21:56:05.0898462+00:00"
}
```

Esempio 10 – Modello JSON - Evento “PEC ricevuta e associata al protocollo” con esempio di indirizzo destinatario PEC specificato

```
{
  "UniqueId": "920a0bf4-7b28-4717-af2d-336681e1d841",
  "WorkflowReferenceId": null,
  "EventModel": {
    "Title": "PEC 1597",
    "Year": null,
    "Number": null,
    "UniqueId": "d89ee72d-fb38-4129-ac86-26d75b545d5c",
    "EntityId": 1597,
    "CustomProperties": { "Receiver": "info@dgroove.it" },
    "ModelType": 2,
    "ModelStatus": 8
  },
  "ReferenceModel": {
    "Title": "Protocollo 2019/0000123",
    "Year": 2019,
    "Number": 123,
    "UniqueId": "5f8c188b-ad97-4440-b2ed-f5a91b8a65a3",
    "EntityId": null,
    "CustomProperties": null,
    "ModelType": 1,
    "ModelStatus": 1
  },
  "EventDate": "2019-06-12T09:43:54.4807717+00:00"
}
```

Gli esempi json sopra descritti sono inclusi come allegati al PDF stesso. Verificare dunque la sezione “allegati” dall’adobe reader

