Trasmissione dati Provider tramite XML a sistema nazionale Agenas

Il modulo è stato realizzato per implementare un meccanismo di invio dati verso il servizio esposto da AGENAS. L'invio ha come oggetto le informazioni relative ai Provider.

1. Job Invii E' stato creato un task executor InvioAgenasTask (seguendo le linee guida degli altri task presenti). Il task executor è gestito da 2 property: • agenas.cronTimer • agenas.scheduler.enabled agenas.cronTimer → valore in cron expression per indicare quando deve girare il thread agenas.scheduler.enabled → se messo a true, il thread eseguito con frequenza agenas.cronTimer esegue le operazioni definite nel metodo sendInviiAgenas del service InvioAgenasTask. invioAgenasTask.sendInviiAgenas() La business logic del modulo è in questo metodo. L'Entity di riferimento è InvioAgenas.java, che registra le informazioni relative agli invii effettuati al servizio Agenas (incluso esito, xml inviato e xml risposta). La logica si articola in 3 fasi: Fase 1: recupera tutti gli inviiAgenas in errore (esito=KO) relativi ai provider e tenta un reinvio. Il reinvio in caso di esito positivo cancella l'invio precedente in errore. Il reinvio rielabora sempre l'xml per la richiesta. Fase 2: Per ogni Provider • se non esiste un precedente invio → effettua invioDati • se esiste un invioAgenas con esito OK si verifica se l'xml inviato o è cambiato → effettua un aggiornaDati o nonècambiato→nulla • se esiste un invioAgenas con esito KO → fa un reinvio Fase 3: invia una email con riepilogo dell'esecuzione. L'email è parametrizzata a seconda dell'esito dell'elaborazione. In linea di massina invia info relative a quando è stata fatta l'ultima elaborazione e se ci sono tati o no invii al servizio Agenas. Inoltre, evidenzia la presenza di eventuali invii con esito KO (vedi punto 2) e la presenza di eventuali eccezioni (da segnalare al HelpDesk per controllare i log).

2. Gestione Errore 3. Generatore XML 4. REST Client Implementazione Punto 1)

Punto 2) Se nella mail di Report viene evidenziata la presenza di errori durante l'ultimo invio, l'utente segreteria può visionare gli invii in errore mediante un'apposita sezione dell'applicativo. Tale sezione mostra solo gli InviiAgenas con esito KO. Per ciascun invio è possibile scaricare l'xml inviato e l'xml della risposta ricevuta per identificare il tipo di errore. Punto 3) La generazione dell'xml della richiesta viene fatta mediante un oggetto DTO appositamente creato per mappare Provider.xsd fornito da Agenas. L'oggetto in questione, DatiProvider.java, viene generato in automatico dal metodo prepareDatiProviderDTO del service AgenasService. Il service AgenasService è anche il service che espone l'interfacciamento con il servizio Agenas. Punto 4) Il servizio REST di Agenas è chiamato mediante un client AgenasWsRestClient, che una volta configurato mediante le property esposte, si occupa di effettuare le chiamate http REST e di convertire la risposta in un oggetto di tipo AgenasResponse. Tuttavia, non si utilizza direttamente AgenasWsRestClient per inviare i dati, ma i metodi esposti dal service AgenasService, che in maniera completa, una volta fornito in input l'id del provider si occupano di: • creare l'oggetto DatiProvider • chiamare il ws mediante il client AgenasWsRestClient • registrare sul db InvioAgensa con l'esito Le property configurabili per questo modulo sono:

agenas.protocol=https agenas.host=serviziweb.agenas.it agenas.port=-1 Servono per costruire la parte inziale dell'endpoint

agenas.ambiente=flussi-test Identifica il flusso Agenas • flussi-test (ambiente collaudo) • flussi (ambiente produzione)

agenas.codice_ente_accreditante=050 Valore che identifica la Regione Veneto nei flussi Agenas agenas.username=PRNDNC86H23Z112R agenas.password=3Dinformatica! Credenziali per accedere al servizio Agenas agenas.proxy.attivo=false Attiva il proxy per le chiamate REST. Il proxy è quello comune all'applicativo e configurabile mediante le property della sezione ######### PROXY ######### agenas.cronTimer=45 26 20 * * * agenas.scheduler.enabled=true ATTIVAZIONE DEL MODULO - cronTimer: Cron expression per definire la frequenza di esecuzione del modulo. N.B. deve essere sempre specificato e valido altrimenti l'applicazione non parte. - scheduler.enabled: definisce se effettivamente il modulo è acceso o spento

agenas.emailReport=dpranteda@3di.it Email a cui verrà inviata l'email con il report