

[Funambol JSON Connector]

[<https://core.forge.funambol.org/ds/viewMessage.do?dsForumId=405&dsMessageId=88921>|Post su forum].

IGSuite connector

- [<http://www.funambol.org>]Funambol homepage] - [<https://www.forge.funambol.org/download/documentation>]
- [<http://www.gsmblog.net/lang-en/guides-and-howtos/34-guides-and-howtos/45-installing-funambol-and-synch>]
- [<http://download.forge.objectweb.org/sync4j/funambol-v7.1-developers-guide.pdf>]Developer guide]
- [<http://www.programmazione.it/index.php?entity=eitem&idItem=38988>]Scrivere un connettore per Funambol]

Note relative alla creazione del connettore

Note relative alla creazione del connettore

E' stato necessario scaricare manualmente il file `funambol-plugin-1.0.8.jar` dall'indirizzo <http://m2.funambol.org/repositories/artifacts/funambol/funambol-plugin/1.0.8/funambol-plugin-1.0.8.jar> e installarlo con

```
mvn install:install-file -DgroupId=funambol -DartifactId=funambol-plugin -Dversion=1.0.8
```

Accesso al database

Come gestire l'accesso al database da parte del connettore?

Il connettore sarà scritto in Java e può usare i driver JDBC per accedere al db. Va bene o si vuole in qualche modo interfacciarsi al database?

[LucaS \(06.06.2009\)](#) La prima che hai detto! secondo me non c'è motivo per collegare il connettore a IG. Il connettore è gestito dal server funambol. L'unica cosa da architettare è trovare un sistema per gestire l'accesso al database (login,pwd,host etc) ma la cosa si risolve facilmente con un piccolo script Perl (che usa IG.pm) e che se chiamato restituisce i dati del database. realizziamo il prototipo

SyncSource

Per implementare il metodo `getUpdatedSyncItemKeys(Timestamp since, Timestamp to)` occorre fornire i Timestamp relativi alla **creazione** e all' **ultima modifica**. Campi da aggiungere sia agli eventi dell'agenda che ai contatti. [LucaS \(06.06.2009\)](#) sei sicuro che per questo metodo occorra anche il campo relativo alla creazione? in fondo il metodo `getUpdatedSyncItemKeys` Comunque nessun problema ad inserire nelle tabelle `contacts` e `calendar` due campi `BIGINT` che potremmo chiamare `ctimestamp` e `mtimestamp` (il valore di `mtimestamp` sarà uguale a `ctimestamp`. In `DBStruct.pm` occorrerà inserire delle query che popolano questi campi (importante il suo valore nella fase iniziale della sincronizzazione). Ci penso io? procedo?

[Ale \(06.06.2009\)](#): Sì, per questo metodo serve effettivamente solo il timestamp di creazione.

Il metodo `getNewSyncItemKeys(Timestamp since, Timestamp to)` richiede invece che venga tenuta traccia almeno della **data di creazione** (o anche dell'ora?) di ogni elemento. [LucaS \(06.06.2009\)](#) Che vuoi dire con "anche dell'ora?" un valore timestamp contiene data e ora (o meglio può fornire data e ora). `Timestamp to` credo che il connettore debba interrogare il db sul campo che inseriremo "`ctimestamp`" utilizzando proprio `ctimestamp`. [Ale \(06.06.2009\)](#): Sì sì... la domanda sull'ora l'avevo inserita prima di capire effettivamente quale fosse il formato di timestamp esistenti... ma meglio aggiungere due campi timestamp `BIGINT`.

Il metodo `getDeletedSyncItemKeys(Timestamp since, Timestamp to)` richiede di tenere traccia degli **elementi cancellati** tra i due timestamp specificati (si possono usare data e ora). `Timestamp to` dalla data+ora, la data/ora in `system_log` è sul fuso orario locale... per ricostruire il Timestamp occorre la data/ora

A cosa serve il metodo `getSyncItemKeysFromTwin(SyncItem syncItem)`???

LucaS(06.06.2009) Ho provato a cercare su internet e sembra che questo metodo serve a verificare (quando si fa lo slo sopra riportato dice questo: *Risulta utile conoscere anche il prototipo di `getSyncItemKeysFromTwin`, che è così definito: questo metodo viene passato un oggetto di tipo `SyncItem` e ritorna le chiavi di più oggetti simili, se esistono. Gli oggetti chiave dell'oggetto, il contenuto e lo stato; successivamente viene istanziato un nuovo `SyncItemImpl`, che setta il valore* Java mi è totalmente oscuro, proverò a cercare ancora su Internet.

Timestamp è la classe [<http://java.sun.com/javase/6/docs/api/java/sql/Timestamp.html>][`java.sql.Timestamp`], per cui i utilizzando un campo `BIGINT` (`BIGINT` è supportato dal framework?) che conterrà i secondi trascorsi da *January 1, 1*

LucaS (06.06.2009) Non ho capito cosa vuoi sapere. A te o meglio al connettore non interessa il framework ma solo il non dovremo far altro che inserire nei due nuovi campi `ctimestamp` e `mtimestamp` il valore della funzione `time()` che res

Ale (06.06.2009): L'unica cosa che volevo sapere era se il framework ha qualche problema ad usare campi `BIGINT`, m