

Versione

2.5



» Web Service Manual

Novembre 2017

Author Tecnoteca srl

www.tecnoteca.com

ITA

www.cmdbuild.org

No part of this document may be reproduced, in whole or in part, without the express written permission of Tecnoteca s.r.l.

CMDBuild ® leverages many great technologies from the open source community: PostgreSQL, Apache, Tomcat, Eclipse, Ext JS, JasperReports, IReport, Enhydra Shark, TWE, OCS Inventory, Liferay, Alfresco, GeoServer, OpenLayers, Prefuse, Quartz, BiMserver. We are thankful for the great contributions that led to the creation of that products.

CMDBuild ® è un prodotto di Tecnoteca S.r.l. che ne ha curato la progettazione e realizzazione, è maintainer dell'applicazione e ne ha registrato il logo.



Al progetto ha anche partecipato come committente iniziale il Comune di Udine – Servizio Sistemi Informativi e Telematici.



CMDBuild ® è rilasciato con licenza open source AGPL (<http://www.gnu.org/licenses/agpl-3.0.html>)

CMDBuild ® è un marchio depositato da Tecnoteca Srl .

In tutte le situazioni in cui viene riportato il logo di CMDBuild® deve essere esplicitamente citato il nome del maintainer Tecnoteca Srl e deve essere presente in modo evidente un link al sito del progetto:

<http://www.cmdbuild.org>.

Il marchio di CMDBuild ®:

- non può essere modificato (colori, proporzioni, forma, font) in nessun modo, nè essere integrato in altri marchi
- non può essere utilizzato come logo aziendale nè l'azienda che lo utilizza può presentarsi come autore / proprietario / maintainer del progetto,
- non può essere rimosso dalle parti dell'applicazione in cui è riportato, ed in particolare dall'intestazione in alto di ogni pagina.

Il sito ufficiale di CMDBuild è <http://www.cmdbuild.org>

Sommario

Introduzione.....	4
I moduli di CMDBuild.....	5
Documentazione disponibile.....	5
Criteri di interoperabilità.....	6
Architettura SOA.....	6
Web service e protocollo SOAP.....	7
Esempi di utilizzo.....	7
I servizi offerti.....	7
Descrizione strutture dati di supporto.....	10
Descrizione metodi.....	13
Definizione di un file di policy per l'autenticazione.....	16
Web service REST.....	18
Servizi.....	18
APPENDICE: Glossario.....	85

Introduzione

CMDBuild è una applicazione Open Source finalizzata a supportare la gestione della configurazione degli oggetti e dei servizi informatici in carico al Dipartimento ICT di una organizzazione e a guidarne i processi di controllo, eventualmente secondo le “best practice” ITIL.

Gestire un Database della Configurazione (CMDB) significa mantenere aggiornata e disponibile per gli altri processi la base dati relativa agli elementi informatici utilizzati, alle loro relazioni ed alle loro modifiche nel tempo.

Con CMDBuild l'amministratore del sistema può costruire ed estendere autonomamente il proprio CMDB (da cui il nome del progetto), modellandolo su misura della propria organizzazione tramite un apposito Modulo di Amministrazione che consente di aggiungere progressivamente nuove classi di oggetti, nuovi attributi e nuove tipologie di relazioni. E' anche possibile definire filtri, “viste” e permessi di accesso ristretti a righe e colonne di ciascuna classe.

CMDBuild è in grado di fornire un completo supporto all'adozione delle “best practice” ITIL, ormai affermatesi come "standard de facto", non proprietario, per la gestione dei servizi informatici secondo criteri orientati ai processi.

Tramite un apposito sistema di gestione dei workflow è possibile definire in modo visuale, con un editor esterno, nuovi processi operanti sulle classi modellate nel database, importarli in CMDBuild ed eseguirli secondo i flussi previsti e con gli automatismi configurati.

E' disponibile un task manager integrato nell'interfaccia utente del Modulo di Amministrazione che consente di gestire in background diverse tipologie di operazioni (avvio di processi, ricezione e invio di mail, esecuzione di connettori) e di controlli sui dati del CMDB (eventi sincroni e asincroni) a fronte delle quali eseguire notifiche, avviare workflow ed eseguire script.

CMDBuild consente la stampa di report tramite il motore open source JasperReports, sia di tipo tabulare prodotti tramite un wizard interno, che di maggiore complessità ottenibili importando template disegnati tramite un apposito editor visuale esterno.

Possono essere poi definite delle dashboard costituite da grafici che mostrino in modo immediato la situazione di alcuni indicatori dello stato corrente del sistema (KPI).

Grazie all'integrazione con il diffuso sistema documentale open source Alfresco è inoltre possibile allegare documenti, immagini, video ed altre tipologie di file alle schede archiviate in CMDBuild.

E' anche possibile utilizzare funzionalità GIS per il georiferimento degli asset e la loro visualizzazione su una mappa geografica (servizi mappe esterni) e / o sulla planimetria di un ufficio (server locale GeoServer) e funzionalità BIM per la visualizzazione di modelli 3D in formato IFC.

Sono poi inclusi nel sistema un webservice SOAP ed un webservice REST, utili per implementare soluzioni di interoperabilità con architettura SOA.

CMDBuild comprende di base due framework denominati Basic Connector e Advanced Connector, che tramite il webservice SOAP sono in grado di sincronizzare le informazioni registrate nel CMDB con fonti dati esterne, ad esempio con sistemi di automatic inventory (quali lo strumento open source OCS Inventory) o con sistemi di virtualizzazione o di monitoraggio.

Un ulteriore strumento, il CMDBuild GUI Framework, consente invece tramite il webservice REST di pubblicare su portali esterni pagine web personalizzate in grado di interagire con il CMDB.

E' infine disponibile una interfaccia utente ottimizzata per strumenti “mobile” (smartphone e tablet), implementata come “app” multiplatforma (iOS, Android) e anch'essa collegata a CMDBuild tramite il webservice REST.

I moduli di CMDBuild

Il sistema CMDBuild comprende due moduli principali:

- il Modulo di Amministrazione, dedicato alla definizione iniziale ed alle successive modifiche del modello dati e delle configurazioni di base (classi e tipologie di relazioni, utenti e permessi, upload report e workflow, opzioni e parametri)
- il Modulo di Gestione dati, dedicato alla consultazione ed aggiornamento delle schede e delle relazioni nel sistema, alla gestione di documenti allegati, all'avanzamento dei processi, alla visualizzazione di dashboard e produzione di report

Il Modulo di Amministrazione è riservato agli utenti abilitati al ruolo di amministratore, il Modulo di Gestione è utilizzato dagli operatori addetti alla consultazione ed aggiornamento dei dati.

Documentazione disponibile

Il presente manuale è dedicato a tecnici informatici gestori di applicativi esterni interessati ad interoperare con CMDBuild e fornisce le indicazioni di dettaglio necessarie ad implementare il sistema di comunicazione tramite cui accedere in lettura e scrittura ai suoi dati ed alle sue funzioni.

Sono disponibili sul sito di CMDBuild (<http://www.cmdbuild.org>) ulteriori specifici manuali dedicati a:

- overview concettuale del sistema (“Overview Manual”)
- amministrazione del sistema (“Administrator Manual”)
- utilizzo del sistema (“User Manual”)
- installazione e gestione tecnica del sistema (“Technical Manual”)
- utilizzo del webservice per l'interoperabilità con sistemi esterni (“Webservice Manual”)
- utilizzo di connettori per la sincronizzazione di dati con sistemi esterni (“ConnectorsManual”)

Criteri di interoperabilità

Architettura SOA

Per rendere interoperabili applicazioni diverse è necessario che queste siano realizzate sotto forma di componenti cooperanti alla implementazione di servizi e che tali servizi vengano esposti attraverso interfacce di alto livello definite secondo protocolli standard.

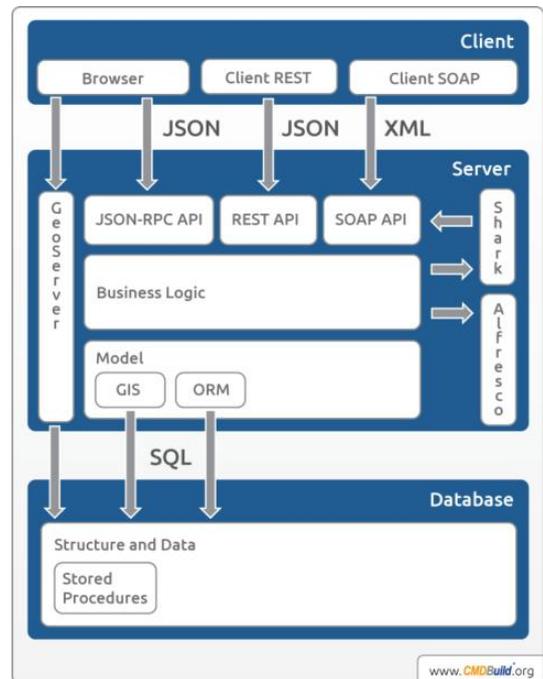
CMDBuild è stato progettato in architettura SOA (Service Oriented Architecture):

- disaccoppiando i diversi livelli logici (vedi schema)
- implementando ed esponendo in ciascuno interfacce esterne specifiche come unica modalità di accesso ai relativi dati e metodi
- utilizzando le stesse interfacce sia per accesso interattivo da parte del client web che per accesso programmatico da parte di applicazioni esterne

Dal punto di vista tecnico è stato scelto di utilizzare la seguenti tecnologie dei web service:

1. protocollo SOAP
2. protocollo REST

Attraverso i web service, e compatibilmente con le politiche di sicurezza definite, CMDBuild rende quindi disponibili i i dati archiviati nel CMDB ed i relativi metodi di gestione per consentirne l'utilizzo nell'ambito di altre applicazioni interessate alle stesse informazioni, sia di gestione tecnica che dedicate ad esigenze amministrative.



Web service e protocollo SOAP

Un web service è un'interfaccia che descrive una collezione di operazioni, accessibili attraverso una rete mediante messaggistica XML.

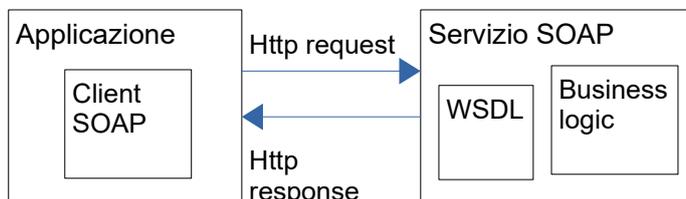
Tramite un web service una applicazione può rendere accessibili le proprie funzionalità ad altre applicazioni operanti attraverso il web.

SOAP è un protocollo standard, basato su XML, per accedere ad un web service.

Le specifiche del protocollo SOAP sono standardizzate da W3C.

La soluzione del web service offre importantissimi vantaggi architetturali:

- consente di ridurre la dipendenza fra applicazioni client e server (“accoppiamento debole”)
- offre un sistema di interoperabilità indipendente dalla piattaforma e dalle tecnologie utilizzate
- agevola l'interoperabilità nel web grazie al fatto di essere basato sul protocollo SOAP che generalmente utilizza HTTP come protocollo base (di base abilitato da tutti i firewall)
- è basato su descrittori XML (WSDL)



Esempi di utilizzo

Il meccanismo di interoperabilità applicativa reso disponibile da CMDBuild tramite il web service SOAP può essere utilizzato per attivare il dialogo con ogni altro sistema informativo preesistente nella stessa organizzazione e che supporti tale protocollo standard.

Esempi non esaustivi di utilizzo possono comprendere:

- attivazione in portali intranet non “JSR 168 compliant” di semplici funzionalità di interazione con CMDBuild dedicate ad utenti non tecnici (stampa report, avvio o avanzamento workflow, ecc)
- sincronizzazione con altri strumenti CMDB
- interazione con applicativi gestionali che necessitino di recuperare da CMDBuild l'inventario aggiornato dei cespiti o debbano riportare in CMDBuild dati amministrativi di un cespite
- integrazione con strumenti tecnici di monitoraggio che necessitino di conoscere informazioni sugli asset soggetti al loro controllo

I servizi offerti

Categorie di metodi

Il web service di CMDBuild rende disponibili metodi tramite cui eseguire dall'esterno le principali funzioni gestite nel sistema, dedicate in particolare alla gestione e stampa delle schede dati (card) ed alla esecuzione di processi.

Più in particolare sono disponibili le seguenti categorie di metodi:

- gestione di schede dati: creazione, modifica, cancellazione, ricerca, storico

- gestione delle liste utilizzate per vincolare dei campi informativi a insiemi di valori predefiniti: creazione, modifica, cancellazione, ricerca sui tipi di liste e sui relativi valori
- gestione delle relazioni fra schede: creazione, modifica, cancellazione, ricerca, storico
- gestione degli allegati alle schede (archiviati nel repository del sistema documentale Alfresco utilizzato da CMDBuild): upload, download, modifica, cancellazione
- gestione dei processi configurati nel sistema (e interpretati dal motore di workflow Enhydra Shark integrato in CMDBuild): avvio, registrazione dati, avanzamento allo step successivo

Tutti i metodi previsti nel web service sono utilizzabili previa autenticazione presso il sistema CMDBuild. L'autenticazione viene effettuata secondo la specifica WSS Username Token profile 1.0¹ con password digest.

La descrizione di dettaglio dei singoli metodi disponibili è riportata al capitolo successivo.

Strutture dati di supporto

I metodi del web service esposto da CMDBuild utilizzano specifiche strutture dati di supporto corrispondenti alle seguenti tipologie di oggetti:

- card (scheda dati)
- attribute (singolo attributo custom della scheda dati)
- lookup (valore da lista predefinita utilizzabile per la valorizzazione di un attributo informativo)
- query (query di filtro per selezione lista card)
- filter (specifica di una condizione atomica di filtro nella query)
- filterOperator (concatenazione condizioni di filtro)
- order (ordinamento fra le schede dati estratte dalla query di filtro)
- relation (correlazione fra schede dati)
- attachment (documento allegato ad una scheda dati)

La descrizione di dettaglio dei singoli oggetti utilizzati è riportata al capitolo successivo.

Sistema di autenticazione

Per poter fruire di tutti i servizi offerti dai web services di CMDBuild, è necessario che l'utente di accesso sia autenticato nel sistema.

Questo è possibile grazie all'utilizzo di autenticazione basata su UsernameToken.

CMDBuild utilizza Axis2 per fornire i propri servizi all'esterno ed il modulo Apache Rampart per gestire l'autenticazione degli utenti via web service, sfruttando lo standard WSS Username Token profile 1.0 con password digest.

Questo standard prevede l'invio delle credenziali, intese come username e hash della password calcolato secondo l'algoritmo definito nella specifica, nell'header del messaggio SOAP.

Il modulo Rampart, una volta arrivato il messaggio, provvederà a verificare se su CMDBuild è presente un utente che corrisponda alle credenziali inviate.

Se questo avviene, e se l'utente ha il permesso di accedere al servizio richiesto, allora il web

¹ Il documento di riferimento è reperibile all'indirizzo <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>

service ritornerà il risultato, altrimenti verrà restituito un messaggio di errore.

Sistema di gestione degli errori

La gestione degli errori nel sistema CMDBuild prevede l'utilizzo di codici di errore custom appositamente definiti.

Nell'utilizzo del web service oltre agli errori propri di Axis (e del modulo Rampart per quanto riguarda l'autenticazione) possono essere quindi ritornati anche errori custom di CMDBuild.

Segue alla tabella successiva la lista dei codici di errore custom di CMDBuild potenzialmente restituibili dal web service, con il relativo significato.

Codice	Significato
NOTFOUND_ERROR	Elemento non trovato
AUTH_MULTIPLE_GROUPS	All'utente sono associati gruppi multipli
AUTH_UNKNOWN_GROUP	Gruppo inesistente
AUTH_NOT_AUTHORIZED	Autorizzazioni non sufficienti ad eseguire l'operazione
ORM_GENERIC_ERROR	Si è verificato un errore durante la lettura/il salvataggio dei dati
ORM_DUPLICATE_TABLE	Esiste già una classe con questo nome
ORM_CAST_ERROR	Errore nella conversione di tipo
ORM_UNIQUE_VIOLATION	Vincolo di non nullità violato
ORM_CONTAINS_DATA	Non è possibile cancellare classi o attributi di tabelle o domini contenenti dati
ORM_TYPE_ERROR	Tipo non corrispondente
ORM_ERROR_GETTING_PK	Impossibile determinare la chiave primaria
ORM_ERROR_LOOKUP_CREATION	Impossibile creare la lookup
ORM_ERROR_LOOKUP_MODIFY	Impossibile modificare la lookup
ORM_ERROR_LOOKUP_DELETE	Impossibile cancellare la lookup
ORM_ERROR_RELATION_CREATE	Impossibile creare la relazione
ORM_ERROR_RELATION_MODIFY	Impossibile modificare la relazione
ORM_CHANGE_LOOKUPTYPE_ERROR	Impossibile cambiare il tipo della lookup
ORM_READ_ONLY_TABLE	Tabella in sola lettura
ORM_READ_ONLY_RELATION	Relazione in sola lettura
ORM_DUPLICATE_ATTRIBUTE	Esiste già un attributo con questo nome
ORM_DOMAIN_HAS_REFERENCE	Impossibile cancellare un dominio a cui sono associati attributi reference
ORM_FILTER_CONFLICT	Conflitto nella definizione del filtro
ORM_AMBIGUOUS_DIRECTION	Impossibile determinare automaticamente la direzione della relazione

Descrizione strutture dati di supporto

Oggetto “Card”

Rappresenta una generica tipologia di scheda dati configurata nel sistema.

Nome	Tipo	Obbligatorio	Default	Descrizione
className	string			Nome della classe cui appartiene la scheda dati. Corrisponde al nome della tabella nel database.
id	integer			Identificativo della card, viene assegnato in automatico dal database.
attributeList	Attribute[]			Array di oggetti di tipo “Attribute” contenente i valori degli ulteriori attributi personalizzati della classe. Corrispondono agli attributi aggiuntivi definiti nel Modulo di Amministrazione di CMDBuild e poi disponibili nella gestione della scheda. La lista è presente anche il ClassId (non il className).
beginDate	data			Indica la data di creazione della versione corrente della scheda (a seguito dell'inserimento iniziale o dell'ultima modifica apportata). E' un oggetto di tipo data nel formato di definizione dei TimeZone nello standard XML Schema (YYYY-MM-DDThh:mm:ssZ).
user	string			Username dell'utente che ha eseguito l'ultima operazione sulla card.

Oggetto “Attribute”

Rappresenta un singolo attributo “custom” (aggiuntivo rispetto i due attributi di default “Code” e “Description”) appartenente ad una tipologia di scheda dati configurata nel sistema.

Nome	Tipo	Obbligatorio	Default	Descrizione
name	string			Nome dell'attributo, come creato dal Modulo di Amministrazione di CMDBuild. Corrisponde al nome della colonna nella tabella del database.
value	string			Corrisponde al valore dell'attributo.
code	string			E' valorizzato solo per attributi di tipo “Reference” con l'id della scheda riferita.

Oggetto “Lookup”

Rappresenta un valore di una lista a valori predefiniti (lista Lookup) utilizzabile per la valorizzazione di un attributo di tipo Lookup.

Nome	Tipo	Obbligatorio	Default	Descrizione
------	------	--------------	---------	-------------

id	integer			Identificativo della voce Lookup, viene assegnato in automatico dal database
type	string			Nome della lista Lookup cui appartiene la voce corrente.
description	string			Descrizione della voce Lookup (una singola voce di una lista Lookup).
code	string			Codice della voce Lookup (una singola voce di una lista Lookup).
parent	Lookup			Oggetto Lookup corrispondente alla voce padre della voce corrente.
parentid	integer			Identificativo della voce Lookup padre della voce corrente (se applicabile)
position	integer			Posizione della voce Lookup nella lista Lookup ordinata cui appartiene
notes	string			Note associate alla voce Lookup

Oggetto “Query”

Rappresenta un filtro sui valori da ricercare. Il filtro può essere unico (Filter) o può essere un insieme di condizioni con un comune operatore di confronto (FilterOperator).

Nome	Tipo	Obbligatorio	Default	Descrizione
filter	Filter			Condizione atomica di filtro
oppure				
filterOperator	FilterOperator			Concatenazione di condizioni di filtro

Oggetto “Filter”

Rappresenta una condizione atomica di filtro tramite cui selezionare una lista di schede dati.

Nome	Tipo	Obbligatorio	Default	Descrizione
name	string			Attributo su cui si applica la condizione di filtro.
value	string			Valore con cui viene comparato il contenuto dell'attributo scelto.
operator	string			Operatore di confronto (sono ammessi i valori EQUALS, LIKE).

Oggetto “FilterOperator”

Rappresenta una concatenazione di condizioni atomiche di filtro collegate da uno stesso operatore.

Nome	Tipo	Obbligatorio	Default	Descrizione
operator	string			Operatore di concatenamento fra condizioni di filtro (sono ammessi i valori AND, OR)
subquery	Filter			Oggetto di tipo “Filter” contenente la

				condizione di filtro da applicare alla query.
--	--	--	--	---

Esempio operazione da eseguire: creazione del filtro "Fornitore = Quasartek s.r.l. OR Fornitore = IBM Italia s.p.a."

Filter1:

name: Fornitore
value: Quasartek s.r.l.
operator: EQUALS

Filter2:

name: Fornitore
value: IBM Italia s.p.a.
operator: EQUALS

FilterOperator:

subquery: [Filter1, Filter2]
operator: OR

Query:

filterOperator: FilterOperator

Oggetto "Order"

Rappresenta il criterio di ordinamento fra le schede dati estratte dalla query di filtro.

Nome	Tipo	Obbligatorio	Default	Descrizione
columnName	string			Attributo su cui effettuare l'ordinamento.
type	string			Tipologia dell'ordinamento da applicare (sono ammessi solo i valori ASC e DESC).

Oggetto "Relation"

Rappresenta una correlazione fra coppie di schede dati archiviate nel sistema.

Nome	Tipo	Obbligatorio	Default	Descrizione
domainName	string			Dominio da utilizzare per la relazione.
class1Name	string			ClassName della prima scheda partecipante alla relazione.
card1Id	integer			Identificatore della prima scheda partecipante alla relazione.
class2Name	string			ClassName della seconda scheda partecipante alla relazione.
card2Id	integer			Identificatore della seconda scheda partecipante alla relazione.
status	string		A	Stato della relazione ('A' = attiva, 'N' = rimossa)
beginDate	date			Data in cui è stata creata la relazione (formato YYYY-MM-DDThh:mm:ssZ)

endDate	date			Data in cui è stata creata la relazione (formato YYYY-MM-DDThh:mm:ssZ)
---------	------	--	--	--

Oggetto “Attachment”

Rappresenta un documento allegato ad una scheda dati archiviata nel sistema.

Nome	Tipo	Obbligatorio	Default	Descrizione
category	string			Categoria cui appartiene l'allegato (da apposita lista Lookup).
description	string			Descrizione associata all'allegato.
filename	string			Nome del file allegato comprensivo dell'estensione
version	string			Versione del documento nel DMS Alfresco
author	string			Utente che esegue l'upload del file
created	date			Data in cui il documento è stato inserito nel DMS
modified	date			Data ultima modifica al documento nel DMS

Descrizione metodi

Area gestione schede dati

Segue la lista dei metodi dedicati alla gestione delle schede dati.

Nome	Input	Output	Descrizione
createCard	Card card	integer id	Crea nel database una nuova scheda dati, contenente le informazioni inserite nell'oggetto “Card”. Restituisce l'attributo identificativo “id”.
deleteCard	string className integer cardId	boolean return	Elimina in modo logico nella classe indicata la scheda preesistente avente “id” indicato. Restituisce “true” se l'operazione è andata a buon fine.
updateCard	string className integer cardId Attribute[] attributeList	boolean return	Aggiorna una scheda preesistente. Restituisce “true” se l'operazione è andata a buon fine.
getCard	string className integer cardId Attribute[] attributeList	Card card	Restituisce la scheda richiesta completa di tutti gli attributi specificati in “attributeList” (di tutti gli attributi della scheda se “attributeList” è nullo).

getCardList	string className Attribute[] attributeList Query queryType Order[] orderType	Card[] cardList	Restituisce la lista di schede risultante dalla query specificata, completa di tutti gli attributi specificati in "attributeList" (di tutti gli attributi della scheda se "attributeList" è nullo). Se la query viene fatta su una superclasse, l'attributo "className" degli oggetti Card restituiti contiene il nome della specifica sottoclasse a cui la card appartiene mentre nella attributeList compare il ClassId della medesima sottoclasse.
getCardHistory	string className integer cardId	Card[] cardList	Restituisce la lista delle versioni storizzate della scheda dati specificata.

Area gestione voci lookup

Segue la lista dei metodi dedicati alla gestione delle voci Lookup (liste di valori predefinite a cui vicolare un attributo di una scheda dati).

Nome	Input	Output	Descrizione
createLookup	Lookup lookup	integer id	Crea nel database una nuova voce di una lista Lookup dati contenente le informazioni inserite nell'oggetto "Lookup". Restituisce l'attributo identificativo "id".
deleteLookup	integer lookupId	boolean return	Cancella in modo logico la voce Lookup avente "id" indicato. Restituisce "true" se l'operazione è andata a buon fine.
updateLookup	Lookup lookup	boolean return	Aggiorna una voce Lookup preesistente. Restituisce "true" se l'operazione è andata a buon fine.
getLookupById	integer id	Lookup lookup	Restituisce la voce Lookup avente l'identificativo "id" specificato.
getLookupList	string type string value boolean parentList	Lookup[] lookupList	Restituisce una intera lista di valori Lookup corrispondenti al tipo "type" specificato. Se è specificato il parametro "value" viene restituita solo la voce corrispondente. Se "parentList" assume il valore "True" viene restituita la gerarchia completa disponibile per le liste Lookup multilivello.

Area gestione relazioni

Segue la lista dei metodi dedicati alla gestione delle relazioni fra schede dati.

Nome	Input	Output	Descrizione
------	-------	--------	-------------

createRelation	Relation relation	boolean return	Crea nel database una nuova relazione fra la coppia di schede dati specificate nell'oggetto "Relation". Restituisce "true" se l'operazione è andata a buon fine.
deleteRelation	Relation relation	boolean return	Cancella la relazione fra la coppia di schede dati specificate nell'oggetto "Relation". Restituisce "true" se l'operazione è andata a buon fine.
getRelationList	string domain string className integer cardId	Relation[] relationList	Restituisce la lista completa delle relazioni della scheda dati specificata per il dominio specificato.
getRelationHistory	Relation relation	Relation[] relationList	Restituisce la lista della storia di tutte le relazioni di una scheda a partire da un oggetto di tipo Relation in cui sono stati definiti unicamente i valori "ClassName" e "CardId".

Area gestione workflow

Segue la lista dei metodi dedicati alla gestione dei processi configurabili nel sistema CMDBuild.

Nome	Input	Output	Descrizione
startWorkflow	Card card boolean CompleteTask	integer id	Avvia una nuova istanza del workflow descritto nella "Card" specificata. Se il parametro "CompleteTask" assume il valore "true" il processo viene fatto avanzare allo step successivo. Restituisce l'attributo identificativo "id".
updateWorkflow	string processId Attribute[] attributeList boolean CompleteTask	boolean ret	Aggiorna le informazioni presenti nella scheda dell'istanza di processo specificata. Se il parametro "CompleteTask" assume il valore "true" il processo viene fatto avanzare allo step successivo. Restituisce "true" se l'operazione è andata a buon fine.

Area gestione documenti allegati

Segue la lista dei metodi dedicati alla gestione dei documenti allegati ad una scheda dati.

Nome	Input	Output	Descrizione
uploadAttachment	string className integer cardId Base64Binary file string fileName string category string description	boolean return	Esegue l'upload del file specificato nel DMS Alfresco ed il relativo collegamento alla scheda dati di CMDBuild appartenente alla classe "className" ed avente l'identificativo "id". Restituisce "true" se l'operazione è andata a buon fine.

downloadAttachment	string className integer cardId string fileName	Base64Binary file	Restituisce il file allegato alla scheda dati specificata ed avente il nome specificato.
deleteAttachment	string className integer cardId string fileName	boolean return	Elimina dal DMS Alfresco il file allegato alla scheda dati specificata ed avente il nome specificato. Restituisce "true" se l'operazione è andata a buon fine.
updateAttachment	string className integer cardId string fileName string description	boolean return	Aggiorna la descrizione del file allegato alla scheda dati specificata ed avente il nome specificato. Restituisce "true" se l'operazione è andata a buon fine.

Definizione di un file di policy per l'autenticazione

Struttura del file

Il file di policy da passare al modulo Rampart lato client, deve essere definito utilizzando il linguaggio Web Services Security Policy Language (WS-SecurityPolicy) .

Questo linguaggio, derivato dall'XML, permette di definire delle regole di sicurezza che l'applicazione deve rispettare.

Ad esempio, per definire l'utilizzo dell'autenticazione tramite Username Token con Password Digest si deve definire la policy nel seguente modo:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsp:Policy wsu:Id="UTOverTransport"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-ssecurity-utility-1.0.xsd" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsp:ExactlyOne>
    <sp:SignedSupportingTokens
      xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
      <wsp:Policy>
        <sp:UsernameToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient">
          <wsp:Policy>
            <sp:HashPassword/>
          </wsp:Policy>
        </sp:UsernameToken>
      </wsp:Policy>
    </sp:SignedSupportingTokens>
  </wsp:ExactlyOne>
</wsp:Policy>
```

È possibile anche indicare nel file di policy, alcuni parametri di configurazione del modulo Rampart.

Se ad esempio non si volesse trasmettere direttamente la password nel codice ma, ad esempio, reperirla da una base di dati, è possibile indicare a Rampart di utilizzare una classe personalizzata che implementa la funzione CallbackHandler in cui può essere definito come reperire la password.

In tal caso il file di policy va modificato nel seguente modo.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsp:Policy wsu:Id="UTOverTransport"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-ssecurity-utility-1.0.xsd" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsp:ExactlyOne>
    <sp:SignedSupportingTokens xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
      <wsp:Policy>
        <sp:UsernameToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient">
          <wsp:Policy>
            <sp:HashPassword/>
          </wsp:Policy>
        </sp:UsernameToken>
      </wsp:Policy>
    </sp:SignedSupportingTokens>
    <ramp:RampartConfig xmlns:ramp="http://ws.apache.org/rampart/policy">
      <ramp:passwordCallbackClass>
        org.cmdbuild.services.soap.secure.MyPasswordHandler
      </ramp:passwordCallbackClass>
    </ramp:RampartConfig>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Per una trattazione più approfondita di WS-SecurityPolicy si rimanda al relativo documento ufficiale².

² <http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>

Web service REST

Servizi

Attenzione: le informazioni in questa sezione potrebbe risultare non aggiornato o non dettagliato; è buona norma fare sempre riferimento al WADL, consultando l'endpoint desiderato (es. http://example.com/cmdbuild/services/rest/v1?_wadi).

Salvo diversamente specificato, il content type dei vari metodi HTTP sarà "application/json".

Al fine di privilegiare la descrizione delle risorse:

- si semplificheranno gli oggetti JSON omettendo i caratteri “
- a proposito dei metodi HTTP, verranno omessi riferimenti all'endpoint selezionato

Versione 1

Endpoint

`http://hostname:port/cmdbuild/services/rest/v1`

Formato date

Tutte le date (attributi time, timestamp, date) utilizzano il formato "**yyyy-MM-dd'T'HH:mm:ss**".

Gestione della sessione

L'accesso alle risorse è subordinato all'apertura di una sessione a cui sono associati una serie di privilegi. Ogni sessione è identificata da una stringa alfanumerica o "token".

Accesso alle risorse che richiedono una sessione

Dopo aver creato o aggiornato una sessione è necessario usare il suo identificatore per tutte quelle risorse che la richiedono. L'invio dell'identificatore viene fatto tramite l'header della richiesta.

```
GET ../classes/foo/cards/ HTTP/1.1
[various other headers]
CMDBuild-Authorization: 0123456789ABCDF...
```

Qualora la risorsa richieda una sessione e il token non venga inviato o non sia non corretto allora la risposta sarà:

```
HTTP/1.1 401 Unauthorized
[various headers]
```

Creazione di una sessione

Questo è l'unico metodo che non richiede una sessione.

```
POST ../sessions/ HTTP/1.1
[various other headers]
```

```
{
  username : ...,
  password : ...
}

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _id : ${SESSION}
    username : ...
    password : null
    role : ...
  }
}
```

Letture di una sessione

```
GET .../sessions/${SESSION}/ HTTP/1.1
[various other headers]

HTTP/1.1 200 OK
[various headers]
{
  username : ...
  password : null
  role : ...
}
```

Facendo attenzione all'attributo ruolo, se valorizzato con "null" o assente allora si avranno delle autorizzazioni parziali, altrimenti si avranno delle autorizzazioni complete. Qualora l'utente a cui abbia un solo ruolo, allora già dopo la creazione questo sarà presente all'interno della sessione. In alternativa sarà necessario aggiornare le credenziali.

Aggiornamento della sessione

```
PUT .../sessions/${SESSION}/ HTTP/1.1
[various other headers]
{
  role : ...,
}

HTTP/1.1 204 No Content
[various headers]
```

È possibile modificare il ruolo associato ad una sessione (passando quindi da una sessione con autorizzazioni parziali a complete e viceversa).

Cancellazione di una sessione

```
DELETE ../sessions/${SESSION}/ HTTP/1.1
[various other headers]

HTTP/1.1 204 OK
[various headers]
```

Lettura dei ruoli associati ad una sessione

```
GET /sessions/${SESSION}/roles/ HTTP/1.1
[various other headers]

HTTP/1.1 200 OK
[various headers]
{
  data : [foo, bar]
}
```

Gestione delle lookup

Lettura di tutti i tipi

```
GET ../lookup_types HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      _id : ...,
      name : ...,
      parent : ...
    },
    ...
  ]
}
```

```
    },  
  ]  
}
```

Letture dei dettagli di un tipo

```
GET ../lookup_types HTTP/1.1  
[various headers]
```

```
HTTP/1.1 200 OK  
[various headers]  
{  
  data : {  
    _id : ...,  
    name : ...,  
    parent : ...  
  }  
}
```

Letture di tutti i valori associati ad un tipo

```
GET ../lookup_types/AlfrescoCategory/values HTTP/1.1  
[various headers]
```

```
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  },  
  data : [  
    {  
      "_type" : AlfrescoCategory,  
      "_id" : 703,  
      "code" : null,  
      "description" : Document,  
      "parent_type" : null,  
      "active" : true,  
      "number" : 1,  
      "default" : false,  
      "parent_id" : null  
    },  
    ...  
  ]  
}
```

```
        ...
    }
]
}
```

Letture dei dettagli di un valore associato ad un tipo

```
GET .../lookup_types/AlfrescoCategory/values/703 HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
{
  data : {
    _type" : AlfrescoCategory,
    _id" : 703,
    code" : null,
    description" : Document,
    parent_type" : null,
    active" : true,
    number" : 1,
    default" : false,
    parent_id" : null
  }
}
```

Gestione di classi e card

Letture di tutte le classi

```
GET .../classes HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _id : Asset,
      name : Asset,
      description : Asset,
```

```
        parent : Class,  
        prototype : true  
    },  
    ...  
    {  
        ...  
    }  
]  
}
```

Letture dei dettagli di una classe

```
GET ../classes/Asset HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  data : {  
    _id : Asset,  
    name : Asset,  
    description : Asset,  
    description_attribute_name : Description,  
    parent : Class,  
    prototype : true  
  }  
}
```

Letture di tutti gli attributi di una classe

```
GET ../classes/Asset/attributes HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  },  
  data : [  
    {  
      lookupType : null,  
      index : 1,  
      group : General data,  
    }  
  ]  
}
```

```
        unique : false,  
        scale : null,  
        editorType : null,  
        targetClass : null,  
        active : true,  
        description : Code,  
        type : string,  
        displayableInList : true,  
        values : [],  
        length : 100,  
        name : Code,  
        _id : Code,  
        filter : null,  
        mandatory : false,  
        precision : null,  
        defaultValue" : null,  
        inherited" : true  
    },  
    ...  
  {  
    ...  
  }  
]  
}
```

Creazione di una card

```
POST .../classes/Building/cards HTTP/1.1  
[various headers]  
{  
  Code = test,  
  ...  
}  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  data : 3217  
}
```

Lettura di tutte le card

```
GET .../classes/Asset/cards HTTP/1.1  
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      Room : null,
      Description : Acer - AL1716,
      Notes : null,
      Workplace : null,
      ScreenSize : null,
      Brand : 138,
      PurchaseDate : null,
      SerialNumber : null,
      AcceptanceNotes : null,
      Code : MON0001,
      TechnicalReference" : null,
      FinalCost : null,
      AcceptanceDate : null,
      Model : AL1716,
      _id : 550,
      _type : Asset,
      Assignee : 134,
      Type" : null,
      Supplier" : null
    },
    {
      ...
    }
  ]
}
```

Lettura del dettaglio di una card

```
GET .../classes/Asset/cards/550 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
```

```
    Room : null,  
    Description : Acer - AL1716,  
    Notes : null,  
    Workplace : null,  
    ScreenSize : null,  
    Brand : 138,  
    PurchaseDate : null,  
    SerialNumber : null,  
    AcceptanceNotes : null,  
    Code : MON0001,  
    TechnicalReference" : null,  
    FinalCost : null,  
    AcceptanceDate : null,  
    Model : AL1716,  
    _id : 550,  
    _type : Asset,  
    Assignee : 134,  
    Type" : null,  
    Supplier" : null  
  }  
}
```

Aggiornamento di una card

```
PUT ../classes/Building/cards/3217 HTTP/1.1  
[various headers]  
{  
  Code = test,  
  ...  
}  
  
HTTP/1.1 204 No Content
```

Cancellazione di una card

```
DELETE ../classes/Building/cards/3217 HTTP/1.1  
[various headers]  
  
HTTP/1.1 204 No Content
```

Gestione di domini e relazioni

Letture di tutti i domini

```
GET ../domains HTTP/1.1
```

```
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _id : WorkplaceComposition,
      name : WorkplaceComposition,
      description : Workplace composition
    },
    ...
    {
      ...
    }
  ]
}
```

Letture dei dettagli di un dominio

```
GET .../domains/WorkplaceComposition HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data" : {
    _id : WorkplaceComposition,
    descriptionMasterDetail : Asset,
    sourceProcess : false,
    descriptionInverse : belongs to workplace,
    descriptionDirect : includes assets,
    destinationProcess : false,
    name : WorkplaceComposition,
    source : Workplace,
    destination : Asset,
    cardinality : 1:N,
    description : Workplace composition
  }
}
```

Lettura di tutti gli attributi di un dominio

```
GET .../domains/UserRole/attributes HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : 1
  },
  data : [
    {
      targetClass : null,
      editorType : null,
      unique : false,
      description : Default Group,
      index : 1,
      defaultValue : null,
      lookupType : null,
      active : true,
      precision : null,
      inherited : false,
      displayableInList : true,
      name : DefaultGroup,
      mandatory : false,
      type : boolean,
      group : null,
      scale : null,
      values : [],
      "_id" : DefaultGroup,
      "filter" : null,
      "length" : null
    }
  ]
}
```

Creazione di una relazione

```
POST .../domains/UserRole/relations/ HTTP/1.1
[various headers]
{
  data : {
    "_sourceType" : User,
```

```
    _sourceId : 123,  
    _destinationType : Role,  
    _destinationId : 456,  
    DefaultGroup : false  
  }  
}  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  data : 789  
}
```

Letture di tutte le relazioni

```
GET .../UserRole/relations HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  },  
  data : [  
    {  
      _destinationDescription : Helpdesk,  
      _sourceDescription : Jones Patricia,  
      _sourceType : User,  
      _sourceId : 678,  
      _destinationType : Role,  
      _type : UserRole,  
      _destinationId : 677,  
      _id : 681  
    },  
    ...  
    {  
      ...  
    }  
  ]  
}
```

Letture del dettaglio di una relazione

```
GET .../domains/UserRole/relations/681 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _id : 681,
    _destinationId : 677,
    _destinationType : Role,
    _destinationDescription : Helpdesk,
    _sourceId : 678,
    _sourceDescription : Jones Patricia,
    _type : UserRole,
    DefaultGroup : null,
    _sourceType : User
  }
}
```

Aggiornamento di una relazione

```
POST .../domains/UserRole/relations/123 HTTP/1.1
[various headers]
{
  data : {
    DefaultGroup : true
  }
}

HTTP/1.1 204 No Content
```

Cancellazione di una relazione

```
DELETE .../domains/UserRole/relations/123 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Gestione di processi, istanze ed attività

Letture di tutti i processi

```
GET .../processes HTTP/1.1
```

```
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _id : RequestForChange,
      name : RequestForChange,
      prototype : false,
      parent : Activity,
      description : Request for change
    }
    ...
    {
      ...
    }
  ]
}
```

```
}
```

Letture dei dettagli di un processo

```
GET .../processes/RequestForChange HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    name : RequestForChange,
    statuses : [
      6,
      7,
      8,
      10
    ],
    description : Request for change,
    description_attribute_name : Description,
    prototype : false,
    _id : RequestForChange,
    parent : Activity,
```

```
        defaultStatus : 6
    }
}
```

Letture di tutti gli attributi di un processo

```
GET .../processes/RequestForChange/attributes HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      name : Dummy,
      scale : null,
      _id : Dummy,
      inherited : false,
      precision" : null,
      description : Dummy,
      editorType : null,
      type : string,
      unique : false,
      targetClass : null,
      defaultValue : null,
      values : [],
      filter : null,
      group : null,
      lookupType : null,
      length : 100,
      mandatory : false,
      index : 0,
      active : true,
      displayableInList : false
    },
    ...
    {
      ...
    }
  ]
}
```

Lettura di tutte le attività di avvio di un processo

```
GET .../processes/RequestForChange/start_activities HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : 1
  }
  data : [
    {
      _id : RegisterRFC,
      description : Register RFC,
      writable : true
    }
  ]
}
```

Lettura dei dettagli di un'attività di avvio di un processo

```
GET .../processes/RequestForChange/start_activities/RegisterRFC HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    attributes : [
      {
        mandatory : true,
        _id : Requester,
        index : 0,
        writable : true
      },
      {
        writable : true,
        index : 1,
        _id : RFCDescription,
        mandatory : true
      }
    ],
  }
}
```

```
    instructions : ...,
    description : Register RFC,
    widgets : [],
    _id : RegisterRFC
  }
}
```

Letture di tutte le attività dell'istanza di un processo

```
GET .../processes/RequestForChange/instances/1460/activities HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      description : Formal evaluation,
      _id : foo,
      writable : true
    },
    ...
    {
      ...
    }
  ]
}
```

Letture dei dettagli di un'attività dell'istanza di un processo

```
GET .../processes/RequestForChange/instances/1460/activities/foo HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    widgets : [
      {
        type : .OpenAttachment,
        active : true,

```

```

        _id : widget-57a9a9d9,
        required : false,
        data : {
            ...
        },
        label : Attachments
    },
    ...
    {
        ...
    }
],
_id : FormalEvaluation,
attributes : [
    {
        mandatory : false,
        writable : false,
        _id : RFCStartDate,
        index : 0
    },
    ...
    {
        ...
    }
],
instructions : ...,
description : Formal evaluation
}
}

```

Creazione dell'istanza di un processo

```

POST .../processes/RequestForChange/instances HTTP/1.1
[various headers]
{
  Requester : 123,
  RFCDescription : test,
  _advance = true
}

HTTP/1.1 200 OK
[various headers]
{
  data : 1460
}

```

```
}
```

Sulla base dell'attributo “_advance”, un'istanza può essere fatta avanzare o solamente salvata.

Letture di tutte le istanze di un processo

```
GET .../processes/RequestForChange/instances HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      Category : null,
      FinalResult : null,
      RFCEndDate : null,
      ExecutionEndDate : null,
      RFCDescription : test,
      Requester : 123,
      ...
      _type : RequestForChange,
      _name : null,
      _id : 1460,
      _status : 6
    },
    ...
    {
      ...
    }
  ]
}
```

Letture del dettaglio di un'istanza di un processo

```
GET .../processes/RequestForChange/instances/1460 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
```

```
{
  data : {
    FinalResult : null,
    RFCEndDate : null,
    ExecutionEndDate : null,
    RFCDescription : test,
    Requester : 123,
    ...
    _type : RequestForChange,
    _name : null,
    _id : 1460,
    _status : 6
  }
}
```

Aggiornamento dell'istanza di un processo

```
PUT ../processes/RequestForChange/instances/1460 HTTP/1.1
[various headers]
{
  Requester : 123,
  RFCDescription : test,
  _advance = true
}

HTTP/1.1 204 No Content
```

Sulla base dell'attributo “_advance”, un'istanza può essere fatta avanzare o solamente salvata.

Cancellazione dell'istanza di un processo

```
DELETE ../processes/RequestForChange/instances/1460 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Gestione del menu

Letture del menu

```
GET ../menu HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
{
  data : {
    menuType : root,
    children : [
      {
        menuType : folder,
        objectId : 0,
        children" : [
          ...
        ],
        ...
      },
      ...
      {
        objectDescription : Employee,
        children : [],
        objectType : Employee,
        objectId : 0,
        index : 34,
        menuType : class
      },
    ]
  }
}
```

Gestione delle configurazioni

Letture delle categorie degli allegati

```
GET ../configuration/attachments/categories HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      _id : Document,
      description : Document"
    },
  ],
}
```

```
    ...
    {
        description : Image,
        _id : Image
    }
]
}
```

Letture degli attributi di una categoria degli allegati

```
> GET .../configuration/attachments/categories/Document/attributes HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      length : null,
      index : 0,
      editorType : null,
      targetClass : null,
      inherited : null,
      ...
    },
    ...
    {
      ...
    }
  ]
}
```

Letture degli stati dei processi

```
GET .../configuration/processes/statuses HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
```

```
    total : ...
  },
  data : [
    {
      _id : 6,
      description : Open,
      value : open
    },
    ...
    {
      ...
    }
  ]
}
```

Gestione dei privilegi

Letture di tutti i privilegi per un gruppo

```
GET .../roles/Foo/classes_privileges HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      name : UPS,
      description : UPS,
      mode : w,
      _id : UPS
    },
    ...
    {
      name : License,
      _id : License,
      description : License,
      mode" : r
    }
  ]
}
```

Gestione degli allegati

Quanto riportato vale nello stesso modo per le istanze del processo. Quindi sostituendo

```
.../classes/${CLASS_ID}/cards/${CARD_ID}
```

con

```
.../processes/${PROCESS_ID}/instances/${PROCESS_ID}
```

si avranno le stesse funzionalità.

Caricamento di un allegato

```
POST .../classes/Building/cards/64/attachments HTTP/1.1
[various headers]
Content-Type : multipart/form-data

== Part, name : "attachment", content-type : "application/json" ==
{
  _description : this is a test,
  _category" : Image,
}
====

== Part, name : "file", content-type : "*/*" ==
binary
====

HTTP/1.1 200 OK
[various headers]
{
  data : abc123
}
```

Note:

- il metodo usa il content-type "multipart/form-data"
- carica il file ed imposta le proprietà

Lettura di tutti gli allegati

```
GET .../classes/Building/cards/64/attachments HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
```

```
meta : {
  total : ...
}
data : [
  {
    _name : test.jpg,
    _id" : "abc123,
    _version : 1.3,
    _description : this is a test,
    _modified" : 2014-12-16T12:35:12,
    _category" : Image,
    _created" : 2014-11-20T14:20:04,
    _author" : admin
  },
  ...
  {
    ...
  }
]
}
```

Letture dei dettagli di un allegato

```
GET ../classes/Building/cards/64/attachments/abc123 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _category : Image,
    _created : 2014-11-20T14:20:04,
    _description : this is a test,
    _author : admin,
    _id : abc123,
    _version : 1.3,
    _modified : 2014-12-16T12:35:12,
    _name : test.jpg
  }
}
```

Download di un allegato

```
GET ../classes/Building/cards/64/attachments/abc123/download.jpg HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
Content-Type : application/octet-stream
```

Aggiornamento di un allegato

```
PUT ../classes/Building/cards/64/attachments/abc123 HTTP/1.1
[various headers]
Content-Type : multipart/form-data

== Part, name : "attachment", content-type : "application/json" ==
{
  _description : this is a test,
  _category" : Image,
}
====

== Part, name : "file", content-type : "*/*" ==
binary
====

HTTP/1.1 204 No Content
```

Note:

- il metodo usa il content-type "multipart/form-data"
- carica il file ed aggiorna le proprietà

Cancellazione di un allegato

```
DELETE ../classes/Building/cards/64/attachments/abc123 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Versione 2

Endpoint

<http://hostname:port/cmdbuild/services/rest/v2>

Formato date

Tutte le date (attributi time, timestamp, date) utilizzano il formato "yyyy-MM-dd'T'HH:mm:ss".

Gestione della sessione

L'accesso alle risorse è subordinato all'apertura di una sessione a cui sono associati una serie di privilegi. Ogni sessione è identificata da una stringa alfanumerica o "token".

Accesso alle risorse che richiedono una sessione

Dopo aver creato o aggiornato una sessione è necessario usare il suo identificatore per tutte quelle risorse che la richiedono. L'invio dell'identificatore viene fatto tramite l'header della richiesta.

```
GET .../classes/foo/cards/ HTTP/1.1
[various other headers]
CMDBuild-Authorization: 0123456789ABCDF...
```

Qualora la risorsa richieda una sessione e il token non venga inviato o non sia non corretto allora la risposta sarà:

```
HTTP/1.1 401 Unauthorized
[various headers]
```

Creazione di una sessione

Questo è l'unico metodo che non richiede una sessione.

```
POST .../sessions/ HTTP/1.1
[various other headers]
{
  username : ...,
  password : ...
}

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _id : ${SESSION}
    username : ...
    password : null
    role : ...
  }
}
```

Lettura di una sessione

```
GET .../sessions/${SESSION}/ HTTP/1.1
```

```
[various other headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{  
  username : ...  
  password : null  
  role : ...  
}
```

Facendo attenzione all'attributo ruolo, se valorizzato con "null" o assente allora si avranno delle autorizzazioni parziali, altrimenti si avranno delle autorizzazioni complete. Qualora l'utente a cui abbia un solo ruolo, allora già dopo la creazione questo sarà presente all'interno della sessione. In alternativa sarà necessario aggiornare le credenziali.

Aggiornamento della sessione

```
PUT .../sessions/${SESSION}/ HTTP/1.1
```

```
[various other headers]
```

```
{  
  role : ...,  
}
```

```
HTTP/1.1 204 No Content
```

```
[various headers]
```

È possibile modificare il ruolo associato ad una sessione (passando quindi da una sessione con autorizzazioni parziali a complete e viceversa).

Cancellazione di una sessione

```
DELETE .../sessions/${SESSION}/ HTTP/1.1
```

```
[various other headers]
```

```
HTTP/1.1 204 OK
```

```
[various headers]
```

Lettura dei ruoli associati ad una sessione

```
GET /sessions/${SESSION}/roles/ HTTP/1.1
```

```
[various other headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  data : [foo, bar]
}
```

Gestione delle lookup

Letture di tutti i tipi

```
GET .../lookup_types HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      _id : ...,
      name : ...,
      parent : ...
    },
    ...
    {
      ...
    },
  ]
}
```

Letture dei dettagli di un tipo

```
GET .../lookup_types HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  data : {
    _id : ...,
    name : ...,
    parent : ...
  }
}
```

Lettura di tutti i valori associati ad un tipo

```
GET .../lookup_types/AlfrescoCategory/values HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      "_type" : AlfrescoCategory,
      "_id" : 703,
      "code" : null,
      "description" : Document,
      "parent_type" : null,
      "active" : true,
      "number" : 1,
      "default" : false,
      "parent_id" : null
    },
    ...
  ]
}
```

Lettura dei dettagli di un valore associato ad un tipo

```
GET .../lookup_types/AlfrescoCategory/values/703 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    "_type" : AlfrescoCategory,
    "_id" : 703,
    "code" : null,
    "description" : Document,
```

```
    parent_type" : null,  
    active" : true,  
    number" : 1,  
    default" : false,  
    parent_id" : null  
  }  
}
```

Gestione di classi e card

Letture di tutte le classi

```
GET ../classes HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  }  
  data : [  
    {  
      _id : Asset,  
      name : Asset,  
      description : Asset,  
      parent : Class,  
      prototype : true  
    },  
    ...  
    {  
      ...  
    }  
  ]  
}
```

Letture dei dettagli di una classe

```
GET ../classes/Asset HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{
```

```
data : {
  _id : Asset,
  name : Asset,
  description : Asset,
  description_attribute_name : Description,
  parent : Class,
  prototype : true
}
```

Letture di tutti gli attributi di una classe

```
GET .../classes/Asset/attributes HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      lookupType : null,
      index : 1,
      group : General data,
      unique : false,
      scale : null,
      editorType : null,
      targetClass : null,
      active : true,
      description : Code,
      type : string,
      displayableInList : true,
      values : [],
      length : 100,
      name : Code,
      _id : Code,
      filter : null,
      mandatory : false,
      precision : null,
      defaultValue" : null,
      inherited" : true
    },
```

```
    ...
    {
        ...
    }
]
}
```

Creazione di una card

```
POST ../classes/Building/cards HTTP/1.1
[various headers]
{
  Code = test,
  ...
}

HTTP/1.1 200 OK
[various headers]
{
  data : 3217
}
```

Letture di tutte le card

```
GET ../classes/Asset/cards HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      Room : null,
      Description : Acer - AL1716,
      Notes : null,
      Workplace : null,
      ScreenSize : null,
      Brand : 138,
      PurchaseDate : null,
      SerialNumber : null,
      AcceptanceNotes : null,
```

```
        Code : MON0001,  
        TechnicalReference" : null,  
        FinalCost : null,  
        AcceptanceDate : null,  
        Model : AL1716,  
        _id : 550,  
        _type : Asset,  
        Assignee : 134,  
        Type" : null,  
        Supplier" : null  
    },  
    {  
        ...  
    }  
]  
}
```

Letture del dettaglio di una card

```
GET ../classes/Asset/cards/550 HTTP/1.1
```

```
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{  
  data : {  
    Room : null,  
    Description : Acer - AL1716,  
    Notes : null,  
    Workplace : null,  
    ScreenSize : null,  
    Brand : 138,  
    PurchaseDate : null,  
    SerialNumber : null,  
    AcceptanceNotes : null,  
    Code : MON0001,  
    TechnicalReference" : null,  
    FinalCost : null,  
    AcceptanceDate : null,  
    Model : AL1716,  
    _id : 550,  
    _type : Asset,  
    Assignee : 134,  
    Type" : null,  
  }  
}
```

```
    "Supplier" : null
  }
}
```

Aggiornamento di una card

```
PUT ../classes/Building/cards/3217 HTTP/1.1
[various headers]
{
  Code = test,
  ...
}

HTTP/1.1 204 No Content
```

Cancellazione di una card

```
DELETE ../classes/Building/cards/3217 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Gestione di domini e relazioni

Letture di tutti i domini

```
GET ../domains HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _id : WorkplaceComposition,
      name : WorkplaceComposition,
      description : Workplace composition
    },
    ...
    {
      ...
    }
  ]
}
```

```
    }  
  ]  
}
```

Lettura dei dettagli di un dominio

```
GET ../domains/WorkplaceComposition HTTP/1.1  
[various headers]
```

```
HTTP/1.1 200 OK  
[various headers]  
{  
  "data" : {  
    "_id" : WorkplaceComposition,  
    "descriptionMasterDetail" : Asset,  
    "sourceProcess" : false,  
    "descriptionInverse" : belongs to workplace,  
    "descriptionDirect" : includes assets,  
    "destinationProcess" : false,  
    "name" : WorkplaceComposition,  
    "source" : Workplace,  
    "destination" : Asset,  
    "cardinality" : 1:N,  
    "description" : Workplace composition  
  }  
}
```

Lettura di tutti gli attributi di un dominio

```
GET ../domains/UserRole/attributes HTTP/1.1  
[various headers]
```

```
HTTP/1.1 200 OK  
[various headers]  
{  
  "meta" : {  
    "total" : 1  
  },  
  "data" : [  
    {  
      "targetClass" : null,  
      "editorType" : null,  
      "unique" : false,  
      "description" : Default Group,  
    }  
  ]  
}
```

```
        index : 1,
        defaultValue : null,
        lookupType : null,
        active : true,
        precision : null,
        inherited : false,
        displayableInList : true,
        name : DefaultGroup,
        mandatory : false,
        type : boolean,
        group : null,
        scale : null,
        values : [],
        _id : DefaultGroup,
        filter" : null,
        length" : null
    }
]
}
```

Creazione di una relazione

```
POST .../domains/UserRole/relations/ HTTP/1.1
[various headers]
{
  _sourceType : User,
  _sourceId : 123,
  _destinationType : Role,
  _destinationId : 456,
  DefaultGroup : false
}

HTTP/1.1 200 OK
[various headers]
{
  data : 789
}
```

Lettura di tutte le relazioni

```
GET .../UserRole/relations HTTP/1.1
[various headers]

HTTP/1.1 200 OK
```

```
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      _destinationDescription : Helpdesk,
      _sourceDescription : Jones Patricia,
      _sourceType : User,
      _sourceId : 678,
      _destinationType : Role,
      _type : UserRole,
      _destinationId : 677,
      _id : 681
    },
    ...
    {
      ...
    }
  ]
}
```

Lettura del dettaglio di una relazione

```
GET .../domains/UserRole/relations/681 HTTP/1.1
```

```
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  data : {
    _id : 681,
    _destinationId : 677,
    _destinationType : Role,
    _destinationDescription : Helpdesk,
    _sourceId : 678,
    _sourceDescription : Jones Patricia,
    _type : UserRole,
    DefaultGroup : null,
    _sourceType : User
  }
}
```

Aggiornamento di una relazione

```
POST .../domains/UserRole/relations/123 HTTP/1.1
[various headers]
{
  data : {
    DefaultGroup : true
  }
}

HTTP/1.1 204 No Content
```

Cancellazione di una relazione

```
DELETE .../domains/UserRole/relations/123 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Gestione di processi, istanze ed attività

Letture di tutti i processi

```
GET .../processes HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _id : RequestForChange,
      name : RequestForChange,
      prototype : false,
      parent : Activity,
      description : Request for change
    }
    ...
    {
      ...
    }
  ]
}
```

```
]
}
```

Letture dei dettagli di un processo

```
GET .../processes/RequestForChange HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    name : RequestForChange,
    statuses : [
      6,
      7,
      8,
      10
    ],
    description : Request for change,
    description_attribute_name : Description,
    prototype : false,
    _id : RequestForChange,
    parent : Activity,
    defaultStatus : 6
  }
}
```

Letture di tutti gli attributi di un processo

```
GET .../processes/RequestForChange/attributes HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      name : Dummy,
      scale : null,
      _id : Dummy,
      inherited : false,
```

```
        precision" : null,
        description : Dummy,
        editorType : null,
        type : string,
        unique : false,
        targetClass : null,
        defaultValue : null,
        values : [],
        filter : null,
        group : null,
        lookupType : null,
        length : 100,
        mandatory : false,
        index : 0,
        active : true,
        displayableInList : false
    },
    ...
    {
        ...
    }
]
}
```

Lettura di tutte le attività di avvio di un processo

```
GET .../processes/RequestForChange/start_activities HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : 1
  }
  data : [
    {
      _id : RegisterRFC,
      description : Register RFC,
      writable : true
    }
  ]
}
```

Letture dei dettagli di un'attività di avvio di un processo

```
GET .../processes/RequestForChange/start_activities/RegisterRFC HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    attributes : [
      {
        mandatory : true,
        _id : Requester,
        index : 0,
        writable : true
      },
      {
        writable : true,
        index : 1,
        _id : RFCDescription,
        mandatory : true
      }
    ],
    instructions : ...,
    description : Register RFC,
    widgets : [],
    _id : RegisterRFC
  }
}
```

Letture di tutte le attività dell'istanza di un processo

```
GET .../processes/RequestForChange/instances/1460/activities HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      description : Formal evaluation,
```

```
        _id : foo,  
        writable : true  
    },  
    ...  
    {  
        ...  
    }  
]  
}
```

Letture dei dettagli di un'attività dell'istanza di un processo

```
GET .../processes/RequestForChange/instances/1460/activities/foo HTTP/1.1  
[various headers]
```

```
HTTP/1.1 200 OK  
[various headers]
```

```
{  
  data : {  
    widgets : [  
      {  
        type : .OpenAttachment,  
        active : true,  
        _id : widget-57a9a9d9,  
        required : false,  
        data : {  
          ...  
        },  
        label : Attachments  
      },  
      ...  
    ]  
  },  
  _id : FormalEvaluation,  
  attributes : [  
    {  
      mandatory : false,  
      writable : false,  
      _id : RFCStartDate,  
      index : 0  
    },  
    ...  
  ]  
}
```

```
        {
            ...
        }
    ],
    instructions : ...,
    description : Formal evaluation
}
}
```

Creazione dell'istanza di un processo

```
POST .../processes/RequestForChange/instances HTTP/1.1
[various headers]
{
  Requester : 123,
  RFCDescription : test,
  _advance = true,
  _activity = StartActivityName
}

HTTP/1.1 200 OK
[various headers]
{
  data : 1460
}
```

Sulla base dell'attributo “_advance”, un'istanza può essere fatta avanzare o solamente salvata.

Lettura di tutte le istanze di un processo

```
GET .../processes/RequestForChange/instances HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      Category : null,
      FinalResult : null,

```

```
        RFCEndDate : null,
        ExecutionEndDate : null,
        RFCDescription : test,
        Requester : 123,
        ...
        _type : RequestForChange,
        _name : null,
        _id : 1460,
        _status : 6
    },
    ...
    {
        ...
    }
]
}
```

Letture del dettaglio di un'istanza di un processo

```
GET .../processes/RequestForChange/instances/1460 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    FinalResult : null,
    RFCEndDate : null,
    ExecutionEndDate : null,
    RFCDescription : test,
    Requester : 123,
    ...
    _type : RequestForChange,
    _name : null,
    _id : 1460,
    _status : 6
  }
}
```

Aggiornamento dell'istanza di un processo

```
PUT .../processes/RequestForChange/instances/1460 HTTP/1.1
[various headers]
{
  Requester : 123,
```

```

RFCDescription : test,
  _advance = true
  _activity = CurrentActivityName
}

```

```
HTTP/1.1 204 No Content
```

Sulla base dell'attributo “_advance”, un istanza può essere fatta avanzare o solamente salvata.

Cancellazione dell'istanza di un processo

```

DELETE .../processes/RequestForChange/instances/1460 HTTP/1.1
[various headers]

```

```
HTTP/1.1 204 No Content
```

Gestione del menu

Letture del menu

```

GET .../menu HTTP/1.1
[various headers]

```

```

HTTP/1.1 200 OK
[various headers]

```

```

{
  data : {
    menuType : root,
    children : [
      {
        menuType : folder,
        objectId : 0,
        children" : [
          ...
        ],
        ...
      },
      ...
      {
        objectDescription : Employee,
        children : [],
        objectType : Employee,
        objectId : 0,

```

```
        index : 34,  
        menuType : class  
    },  
]  
}  
}
```

Gestione delle configurazioni

Letture delle categorie degli allegati

```
GET .../configuration/attachments/categories HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  },  
  data : [  
    {  
      _id : Document,  
      description : Document"  
    },  
    ...  
    {  
      description : Image,  
      _id : Image  
    }  
  ]  
}
```

Letture degli attributi di una categoria degli allegati

```
> GET .../configuration/attachments/categories/Document/attributes HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  },  
}
```

```
data : [
  {
    length : null,
    index : 0,
    editorType : null,
    targetClass : null,
    inherited : null,
    ...
  },
  ...
  {
    ...
  }
]
```

Letture degli stati dei processi

```
GET .../configuration/processes/statuses HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      _id : 6,
      description : Open,
      value : open
    },
    ...
    {
      ...
    }
  ]
}
```

Gestione dei privilegi

Letture di tutti i privilegi per un gruppo

```
GET .../roles/Foo/classes_privileges HTTP/1.1
```

```
[various headers]
HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      name : UPS,
      description : UPS,
      mode : w,
      _id : UPS
    },
    ...
    {
      name : License,
      _id : License,
      description : License,
      mode" : r
    }
  ]
}
```

Gestione degli allegati

Quanto riportato vale nello stesso modo per le istanze del processo. Quindi sostituendo

```
.../classes/${CLASS_ID}/cards/${CARD_ID}
```

con

```
.../processes/${PROCESS_ID}/instances/${PROCESS_ID}
```

si avranno le stesse funzionalità.

Caricamento di un allegato

```
POST .../classes/Building/cards/64/attachments HTTP/1.1
[various headers]
Content-Type : multipart/form-data

== Part, name : "attachment", content-type : "application/json" ==
{
  _description : this is a test,
  _category" : Image,
}
====
```

```
== Part, name : "file", content-type : "*/*" ==
binary
====

HTTP/1.1 200 OK
[various headers]
{
  data : abc123
}
```

Note:

- il metodo usa il content-type "multipart/form-data"
- carica il file ed imposta le proprietà

Letture di tutti gli allegati

```
GET .../classes/Building/cards/64/attachments HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _name : test.jpg,
      _id : "abc123",
      _version : 1.3,
      _description : this is a test,
      _modified" : 2014-12-16T12:35:12,
      _category" : Image,
      _created" : 2014-11-20T14:20:04,
      _author" : admin
    },
    ...
    {
      ...
    }
  ]
}
```

Letture dei dettagli di un allegato

```
GET .../classes/Building/cards/64/attachments/abc123 HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _category : Image,
    _created : 2014-11-20T14:20:04,
    _description : this is a test,
    _author : admin,
    _id : abc123,
    _version : 1.3,
    _modified : 2014-12-16T12:35:12,
    _name : test.jpg
  }
}
```

Download di un allegato

```
GET .../classes/Building/cards/64/attachments/abc123/download.jpg HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
Content-Type : application/octet-stream
```

Aggiornamento di un allegato

```
PUT .../classes/Building/cards/64/attachments/abc123 HTTP/1.1
[various headers]
Content-Type : multipart/form-data

== Part, name : "attachment", content-type : "application/json" ==
{
  _description : this is a test,
  _category : Image,
}
====

== Part, name : "file", content-type : "*/*" ==
```

```
binary
=====

HTTP/1.1 204 No Content
```

Note:

- il metodo usa il content-type “multipart/form-data”
- carica il file ed aggiorna le proprietà

Cancellazione di un allegato

```
DELETE ../classes/Building/cards/64/attachments/abc123 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Query CQL**Esecuzione di una query**

```
GET .../cql?... HTTP/1.1
[various headers]
```

Parametri:

- **filter:** contiene l'espressione

```
{
  CQL: ...
}
```

- **sort:** contiene l'elenco ordinato degli attributi con cui eseguire l'ordinamento

```
[
  {
    property: Description
    direction: ASC
  }, {
    property: Code
    direction: DESC
  }
]
```

- **limit:** contiene il numero dei risultati restituiti
- **start:** contiene lo spiazzamento rispetto all'inizio della lista

```
HTTP/1.1 200 OK
```

```
[various headers]
{
  meta : {
    total : ...
  }
  data : [
    {
      _id : Asset,
      name : Asset,
      description : Asset,
      parent : Class,
      prototype : true
    },
    ...
    {
      ...
    }
  ]
}
```

Gestione dei template delle e-mail

Letture di tutti i template delle e-mail

```
GET ../email_templates?start=...&limit=... HTTP/1.1
[various headers]
HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [ foo, bar, baz ]
}
```

Letture dei dettagli di un template

```
GET ../email_templates/foo HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _id: ...
  }
}
```

```

    name: ...
    description: ...
    from: ...
    to: ...
    cc: ...
    bcc: ...
    subject: ...
    body: ...
    notifyWith: ...
    noSubjectPrefix: ...
    account: ...
    keepSynchronization: ...
    promptSynchronization: ...
    delay: ...
  }
}

```

Gestione delle e-mail

Quanto riportato vale nello stesso modo per le istanze del processo. Quindi sostituendo

```
.../classes/${CLASS_ID}/cards/${CARD_ID}/emails
```

con

```
.../processes/${PROCESS_ID}/instances/${PROCESS_ID}/emails
```

si avranno le stesse funzionalità.

Letture degli stati di una e-mail

```

GET .../classes/Building/cards/64/emails/statuses HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  }
  data : [ received, draft, outgoing, sent ]
}

```

Affinchè possa essere inviata lo stato di una e-mail deve essere settato come “outgoing” e a quel punto l'e-mail non è più modificabile.

Creazione di una e-mail

```
POST .../classes/Building/cards/64/emails HTTP/1.1
[various headers]
{
  from: ...
  to: ...
  cc: ...
  bcc: ...
  subject: ...
  body: ...
  status: ...
  notifyWith: ...
  noSubjectPrefix: ...
  account: ...
  template: ...
  keepSynchronization: ...
  promptSynchronization: ...
  delay: ...
}

HTTP/1.1 200 OK
[various headers]
{
  data : 123
}
```

Le e-mail vengono sempre create con lo stato in “draft”. Per l’invio è necessario aggiornare l’e-mail dopo la creazione impostando lo stato in “outgoing”.

Lettura di tutte le e-mail

```
GET .../classes/Building/cards/64/emails HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  } ,
  data : [
    {
      from: ...
      to: ...
    }
  ]
}
```

```
        cc: ...
        bcc: ...
        subject: ...
        body: ...
        status: ...
        notifyWith: ...
        noSubjectPrefix: ...
        account: ...
        template: ...
        keepSynchronization: ...
        promptSynchronization: ...
        delay: ...
    },
    {
        ...
    }
]
}
```

Letture del dettaglio di una e-mail

```
GET .../classes/Building/cards/64/emails/123 HTTP/1.1
```

```
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  data : {
    from: ...
    to: ...
    cc: ...
    bcc: ...
    subject: ...
    body: ...
    status: ...
    notifyWith: ...
    noSubjectPrefix: ...
    account: ...
    template: ...
    keepSynchronization: ...
    promptSynchronization: ...
    delay: ...
  }
}
```

Aggiornamento di una e-mail

```
PUT .../classes/Building/cards/64/emails/123 HTTP/1.1
[various headers]
{
  from: ...
  to: ...
  cc: ...
  bcc: ...
  subject: ...
  body: ...
  status: ...
  notifyWith: ...
  noSubjectPrefix: ...
  account: ...
  template: ...
  keepSynchronization: ...
  promptSynchronization: ...
  delay: ...
}

HTTP/1.1 204 No Content
```

Cancellazione di una e-mail

```
DELETE .../classes/Building/cards/64/emails/123 HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Gestione delle funzioni

Letture di tutte le funzioni

```
GET .../functions?start=...&limit=... HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  } ,
}
```

```
data : [  
  {  
    name: ...  
    description: ...  
  },  
  {  
    ...  
  }  
]
```

Lettura dei dettagli di una funzione

```
GET .../functions/foo HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  data : {  
    name: ...  
    description: ...  
  }  
}
```

Lettura dei valori/attributi di ingresso una funzione

```
GET .../functions/foo/parameters?start=...&limit=... HTTP/1.1  
[various headers]  
  
HTTP/1.1 200 OK  
[various headers]  
{  
  meta : {  
    total : ...  
  },  
  data : [  
    {  
      type: ...  
      name: ...  
      description: ...  
      ...  
    },  
    {
```

```
    ...
  }
]
}
```

Letture dei valori/attributi di uscita di una funzione

```
GET .../functions/foo/attributes?start=...&limit=... HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      type: ...
      name: ...
      description: ...
      ...
    },
    {
      ...
    }
  ]
}
```

Esecuzione di una funzione per ottenere i suoi risultati

```
GET .../functions/foo/outputs HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      ...
    },
    {
      ...
    }
  ]
}
```

```
    {
      ...
    }
  ]
}
```

Gestione dei report

Letture di tutti i report

```
GET .../reports?start=...&limit=... HTTP/1.1
```

```
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      _id: ...
      description: ...
    },
    {
      ...
    }
  ]
}
```

Letture dei dettagli di un report

```
GET .../reports/123 HTTP/1.1
```

```
[various headers]
```

```
HTTP/1.1 200 OK
```

```
[various headers]
```

```
{
  data : {
    _id: ...
    title: ...
    description: ...
  }
}
```

Letture dei valori/attributi di ingresso di un report

```
GET .../reports/123/attributes?start=...&limit=... HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      type: ...
      name: ...
      description: ...
      ...
    },
    {
      ...
    }
  ]
}
```

Download di un report

```
GET .../reports/123/somename.someextension?extension=...&parameters={"foo": "FOO",
"bar": "BAR", ...} HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
Content-Type : application/octet-stream
```

Gestione degli alberi dei domini

Letture di tutti gli alberi

```
GET .../domainTrees?start=...&limit=... HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
```

```
{
  meta : {
    total : ...
  },
  data : [
    {
      _id: ...
      description: ...
    },
    {
      ...
    }
  ]
}
```

Letture dei dettagli di un albero

```
GET .../domainTrees/123 HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
```

```
{
  data : {
    _id: ...
    description: ...
    nodes: [
      {
        _id: ...
        parent: ...
        metadata: {
          ...
        }
      },
      ...
    ]
  }
}
```

Gestione delle immagini

Letture di tutte le cartelle

```
GET .../filestores/images/folders HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      _id: ...
      name: ...
      parent: ...
    },
    {
      ...
    }
  ]
}
```

Letture dei dettagli di una cartella

```
GET .../filestores/images/folders/foo HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _id: ...
    name: ...
    parent: ...
  }
}
```

Caricamento di un file in una cartella

```
POST .../filestores/images/folders/foo/files HTTP/1.1
[various headers]
Content-Type : multipart/form-data

== Part, name : "file", content-type : "*/*" ==
binary
=====
```

```
HTTP/1.1 200 OK
[various headers]
{
  data : bar
}
```

Letture di tutti i file in una cartella

```
GET .../filestores/images/folders/foo/files HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  },
  data : [
    {
      _id: ...
      name: ...
      parent: ...
    },
    {
      ...
    }
  ]
}
```

Letture dei dettagli di un file

```
GET .../filestores/images/folders/foo/files/bar HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  data : {
    _id: ...
    name: ...
    parent: ...
  }
}
```

Download di un file

```
GET .../filestores/images/folders/foo/files/bar/download HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
Content-Type : application/octet-stream
```

Cancellazione di un file

```
DELETE .../filestores/images/folders/foo/files/bar HTTP/1.1
[various headers]

HTTP/1.1 204 No Content
```

Gestione delle icone

Modello JSON dell'icona:

```
{
  _id: ...
  type: ...
  details: {
    ...
  }
  image: {
    type: ...
    details: {
      ...
    }
  }
}
```

type: rappresenta il tipo di di icona da caricare, attualmente i tipi supportati sono

- class
- process

details: dipende da type

- *type: class*
id deve contenere il nome della classe
- *type: process*
id deve contenere il nome del processo

image: rappresenta l'immagine da associare

- *type*: "filestore"
- *details*: dipendono da type ma siccome "filestore" è l'unico supportato
 - *store*: id dello store (vedi Metodi per la gestione dei file)
 - *folder*: id del folder (vedi Metodi per la gestione dei file)
 - *file*: id del file (vedi Metodi per la gestione dei file)

Creazione di un'icona

```
POST .../icons HTTP/1.1
[various headers]
{
  type: ...
  details: {
    ...
  }
  image: {
    type: ...
    details: {
      ...
    }
  }
}

HTTP/1.1 200 OK
[various headers]
{
  data : 123
}
```

Letture di tutte le icone

```
GET .../icons HTTP/1.1
[various headers]

HTTP/1.1 200 OK
[various headers]
{
  meta : {
    total : ...
  } ,
  data : [
```

```
{
  _id: ...
  ...
},
{
  ...
}
]
}
```

Letture dei dettagli di un'icona

```
GET .../icons/123 HTTP/1.1
[various headers]
```

```
HTTP/1.1 200 OK
[various headers]
{
  data : {
    ...
  }
}
```

Aggiornamento di un'icona

```
PUT .../icons/123 HTTP/1.1
[various headers]
{
  ...
}
```

```
HTTP/1.1 204 No Content
```

Cancellazione di un file

```
DELETE .../icons/123 HTTP/1.1
[various headers]
```

```
HTTP/1.1 204 No Content
```

APPENDICE: Glossario

ALLEGATO

Per “allegato” si intende un qualunque file associabile ad una scheda dati inserita nel sistema.

Per la gestione degli allegati CMDBuild utilizza in modalità embedded un qualunque sistema documentale compatibile con il protocollo standard CMIS (oppure il DMS Alfresco fino alla versione 3 tramite il proprio webservice nativo).

La gestione degli allegati supporta il versioning di file caricati più volte, con numerazione automatica.

ATTIVITA'

Per “attività” si intende uno dei passaggi che costituiscono il processo.

Una attività è caratterizzata da un nome, un esecutore, un tipo, eventuali attributi, eventuali metodi associati ad API di CMDBuild per poter essere eseguiti.

Per “istanza di attività” si intende una specifica attivazione di una attività, effettuata automaticamente dal sistema o manualmente da un operatore.

Vedi anche: Processo

ATTRIBUTO

Il termine indica nel sistema CMDBuild la generica tipologia di informazione descrittiva di una determinata classe.

CMDBuild consente tramite il Modulo Schema di creare nuovi attributi in una classe o in un dominio e di modificarne alcune caratteristiche.

Nella classe “Fornitore” gli attributi sono ad esempio il nome, l'indirizzo, il numero di telefono, ecc.

Ogni attributo corrisponde nel Modulo di Gestione a campi di inserimento dati sulla apposita scheda di gestione della classe e a colonne della corrispondente tabella nel database.

Vedi anche: Classe, Dominio, Relazione, Superclasse, Tipo di attributo

BIM

Metodologia che si pone l'obiettivo di supportare l'intero ciclo di vita di un edificio, dall'idea iniziale alla fase di costruzione, di utilizzo e manutenzione, fino alla eventuale demolizione finale.

La metodologia BIM (Building Information Modeling) è supportata da numerosi programmi informatici che possono interagire tramite un formato aperto di scambio dati denominato IFC (Industry Foundation Classes).

Vedi anche: GIS

CI

Si definisce Configuration Item (Elemento della Configurazione) ogni elemento che concorre a fornire il servizio IT all'Utente, considerato ad un livello di dettaglio sufficiente per la sua gestione tecnica e patrimoniale.

Esempi di CI sono: server, workstation, programma applicativo, sistema operativo, stampante, ecc

Vedi anche: Configurazione

CLASSE

Il termine rappresenta un tipo di dati complesso caratterizzato da un insieme di attributi che nel loro insieme descrivono quel tipo di dato.

Una classe modella una tipologia di oggetto da gestire nel CMDB, quale ad esempio un computer, una applicazione software, un servizio, un fornitore, ecc

CMDBuild consente all'Amministratore del Sistema, attraverso il Modulo Schema, di definire nuove classi e di cancellare o modificare la struttura di classi già definite.

Una classe è rappresentata a video da una apposita scheda di gestione dati e nel database da una tavola generata automaticamente al momento della definizione della classe.

Vedi anche: Scheda, Attributo

CONFIGURAZIONE

Il processo di Gestione della Configurazione ha lo scopo di mantenere aggiornata e disponibile per gli altri processi la base di informazioni relativa agli oggetti informatici gestiti (CI), alle loro relazioni ed alla loro storia.

E' uno dei principali processi gestiti dal sistema ITIL.

Vedi anche: CI, ITIL

DASHBOARD

Una dashboard corrisponde in CMDBuild ad una raccolta di grafici di diversa tipologia, tramite cui avere immediata evidenza di alcuni parametri chiave (KPI) relativi ad un particolare aspetto di gestione del servizio IT.

Vedi anche: Report

DATABASE

Il termine indica un insieme di informazioni strutturato ed organizzato in archivi residenti sull'elaboratore server, nonché l'insieme dei programmi di utilità dedicati alla gestione dei tali informazioni per attività quali inizializzazione, allocazione degli spazi, ottimizzazione, backup, ecc.

CMDBuild si appoggia sul database PostgreSQL, il più potente, affidabile e completo database Open Source, di cui utilizza in particolare le sofisticate funzionalità e caratteristiche object oriented.

DOMINIO

Un dominio rappresenta una tipologia di relazione fra una coppia di classi.

E' caratterizzato da un nome, dalle descrizioni della funzione diretta ed inversa, dai codici delle due classi e dalla cardinalità (numerosità degli elementi relazionabili) ammessa, nonché dagli eventuali attributi configurati.

CMDBuild consente all'Amministratore del Sistema, attraverso il Modulo Schema, di definire nuovi domini e di cancellare o modificare la struttura di domini già definiti.

E' possibile caratterizzare ciascun dominio tramite definizione di attributi custom.

Vedi anche: Classe, Relazione

FILTRO DATI

Un filtro dati è una restrizione della lista degli elementi contenuti in una classe, ottenuta specificando condizioni booleane (uguale, diverso, contiene, inizia, ecc) sui possibili valori assumibili da ciascun attributo della classe.

I filtri dati possono essere definiti ed utilizzati “una tantum”, oppure possono essere memorizzati dall'operatore e richiamati successivamente (dallo stesso operatore o da operatori di altri gruppi di utenti ai quali l'Amministratore del sistema abbia concesso l'utilizzo).

Vedi anche: Classe, Vista

GIS

Un sistema GIS è un sistema informatico in grado di produrre, gestire e analizzare dati spaziali associando a ciascun elemento geografico una o più descrizioni alfanumeriche.

Le funzionalità GIS implementate in CMDBuild consentono di creare attributi geometrici, in aggiunta a quelli testuali, tramite cui rappresentare su scala locale (planimetrie) o su scala più estesa (mappe esterne) elementi puntuali (ad esempio gli asset IT), poligonali (ad esempio linee dati) o aree (piani, stanze, ecc).

Vedi anche: BIM

GUI FRAMEWORK

E' una interfaccia utente completamente personalizzabile e orientata a fornire un accesso semplificato all'applicazione, pubblicabile su portali web di qualsiasi tecnologia ed interoperabile con CMDBuild tramite il webservice REST standard.

Vedi anche: Mobile, Webservice

ITIL

Sistema di "best practice" ormai affermatosi come "standard de facto", non proprietario, per la gestione dei servizi informatici secondo criteri orientati ai processi (Information Technology Infrastructure Library).

Fra i processi fondamentali coperti da ITIL ci sono quelli del Service Support, comprendenti l'Incident Management, il Problem Management, il Change Management, il Configuration Management ed il Release Management.

Per ogni processo considera la descrizione, i componenti di base, i criteri e gli strumenti consigliati per la misura della qualità del servizio, i ruoli e le responsabilità delle risorse coinvolte, i punti di integrazione con gli altri processi (per eliminare duplicazioni e inefficienze).

Vedi anche: Configurazione

LOOKUP

Con il termine “LookUp” si indica una coppia di valori del tipo (Codice, Descrizione) impostabili dall'Amministratore del Sistema tramite il Modulo Schema.

Tali valori vengono utilizzati dall'applicazione per vincolare la scelta dell'utente, al momento della compilazione del relativo campo sulla scheda dati, ad uno dei valori preimpostati.

Il Modulo Schema consente la definizione di nuove tabelle di “LookUp” secondo le necessità dell'organizzazione.

MOBILE

E' una interfaccia utente ottimizzata per strumenti "mobile" (smartphone e tablet), implementata come "app" multiplatforma (iOS, Android) ed interoperabile con CMDBuild tramite il webservice REST standard.

Vedi anche: GUI Framework, Webservice

PROCESSO

Per "processo" (o workflow) si intende una sequenza di passaggi ("attività") descritti nel sistema per svolgere in forma guidata e secondo regole prestabilite una determinata azione.

Per ogni processo saranno avviate in CMDBuild una serie di "istanze di processo", una per ogni necessità di effettiva esecuzione dell'azione corrispondente, che avrà luogo su "asset" specifici e sarà svolta da utenti specifici.

Una "istanza di processo" viene attivata tramite avvio e conferma del primo passaggio previsto e termina alla esecuzione dell'attività finale prevista nella definizione.

Vedi anche: Attività

RELAZIONE

Per "Relazione" si intende in CMDBuild un collegamento effettivo di due schede appartenenti a due classi, o in altri termini una istanza di un dato dominio.

Una relazione è quindi definita da una coppia di identificativi univoci delle due schede collegate e dall'identificativo del dominio utilizzato per il collegamento, nonché dalla valorizzazione degli eventuali attributi previsti nel dominio.

CMDBuild consente agli operatori del Sistema, attraverso il Modulo Gestione Dati, di definire nuove relazioni fra le schede archiviate nel database.

Vedi anche: Classe, Dominio

REPORT

Il termine indica in CMDBuild una stampa (in formato PDF o CSV) riportante in forma analitica le informazioni estratte da una o più classi fra le quali sia definita una catena di domini.

I report possono essere generati e modificati dagli operatori di CMDBuild tramite una apposita funzione del Modulo di Gestione Dati e la relativa definizione viene memorizzata nel database per poter essere riutilizzata successivamente.

Vedi anche: Classe, Dominio, Database

SCHEDA

Con il termine "Scheda" in CMDBuild si riferisce un elemento archiviato in una determinata classe.

Una scheda è caratterizzata da un insieme di valori assunti da ciascuno degli attributi definiti per la sua classe di appartenenza.

CMDBuild consente agli operatori del Sistema, attraverso il Modulo Gestione Dati, di archiviare nuove schede nel database e di aggiornare schede già archiviate.

Le informazioni di ogni scheda saranno memorizzate nel database alle opportune colonne di una riga della tavola generata per la classe di appartenenza della scheda.

Vedi anche: Classe, Attributo

SUPERCLASSE

Una superclasse è una classe astratta utilizzabile per definire una sola volta attributi condivisi fra più classi. Da tale classe astratta è poi possibile derivare classi reali che conterranno i dati effettivi e che comprenderanno sia gli attributi condivisi (specificati nella superclasse) che quelli specifici della sottoclasse.

Ad esempio è possibile definire la superclasse “Computer” con alcuni attributi base (RAM, HD, ecc) e le sottoclassi derivate “Desktop”, “Notebook”, “Server”, ciascuna delle quali con i soli attributi specifici.

Vedi anche: Classe, Attributo

TIPO DI ATTRIBUTO

Ogni attributo definito per una determinata classe è caratterizzato da un “Tipo” che determina le caratteristiche delle informazioni contenute e la loro modalità di gestione.

Il tipo di attributo viene definito con il Modulo Schema e può essere poi modificato entro alcuni limiti dipendenti dalla tipologia dei dati già archiviati.

CMDBuild gestisce i seguenti tipi di attributo: “Boolean” (booleano, Si / No), “Date” (data), “Decimal” (decimale), “Double” (virgola mobile in doppia precisione), “Inet” (indirizzo IP), “Integer” (numero intero), “LookUp” (tabellato da lista configurabile in “Impostazioni” / “LookUp”), “Reference” (riferimento o foreign key), “String” (stringa), “Text” (testo lungo), “TimeStamp” (data e ora).

Vedi anche: Attributo

VISTA

Una vista è un insieme di schede definito in modo “logico” anziché dal fatto di costituire l'intero contenuto di una classe nel CMDB.

In particolare una vista può essere definita in CMDBuild applicando un filtro ad una classe (quindi conterrà un insieme ridotto delle stesse righe) oppure specificando una funzione SQL che estragga attributi da una o più classi correlate.

La prima tipologia di vista mantiene tutte le funzionalità disponibili per una classe, la seconda consente la sola visualizzazione e ricerca con filtro veloce.

Vedi anche: Classe, Filtro

WEBSERVICE

Un webservice è un'interfaccia che descrive una collezione di operazioni, accessibili attraverso una rete mediante messaggistica XML.

Tramite un webservice una applicazione può rendere accessibili le proprie funzionalità ad altre applicazioni operanti attraverso il web.

CMDBuild dispone di un webservice SOAP e di un webservice REST.

WIDGET

Un widget è un componente grafico di una interfaccia utente di una applicazione software, che ha lo scopo di facilitare all'utente l'interazione con l'applicazione stessa.

CMDBuild prevede l'utilizzo di widget sotto forma di “pulsanti” posizionabili su schede dati o su schede di avanzamento di processi. I pulsanti aprono finestre di tipo “popup” tramite cui inserire se richiesto informazioni aggiuntive e visualizzare poi l'output della funzione richiamata.