

DocWay4: Configurazione MongoDB/Elasticsearch

DocWay4 (e gli altri applicativi ad esso legati) supportano l'utilizzo di database MongoDB/Elasticsearch in sostituzione di eXtraWay. Viene garantita (per quanto possibile) la piena compatibilità fra le 2 configurazioni (eXtraWay o MongoDB/Elasticsearch). Al fine di minimizzare l'integrazione all'interno di tutte le applicazioni legate a DocWay, la maggior parte degli interventi di sviluppo è stata realizzata a livello di [broker.jar](#).

Solo le funzionalità di eXtraWay richieste da DocWay sono state riportate sull'implementazione MongoDB/Elasticsearch, in particolare:

- Caricamento e salvataggio di documenti in base al numero fisico (assegnato da MongoDB);
- Esecuzione di query in linguaggio nativo eXtraWay. Sono state fatte verifiche solo su query utilizzate su DocWay, non tutte le funzionalità di eXtraWay sono supportate (es. query su bacino di adiacenza);
- Mantenimento delle selezioni derivanti da ricerca tramite cache applicativa. Questo significa che non esiste una unica cache condivisa da tutti gli applicativi, ma ogni istanza di applicativo (ogni singolo broker) gestisce la propria cache di ricerche;
- Assegnazione di Seriali;
- Gestione delle regole di univocità;
- Gestione di gerarchie;
- Vocabolari;
- Gestione Thesauri (non tutte le relazioni supportate da eXtraWay, ma solo quelle richieste da DocWay);
- Gestione file con salvataggio su Storage Server. Per ottimizzazione delle prestazioni non viene invocato il servizio REST dello Storage ma viene invocato il salvataggio dei file tramite codice JAVA (configurazione Storage Server su ogni applicativo).

Requisiti

Gli unici requisiti software richiesti dall'implementazione MongoDB/Elasticsearch rispetto a quella eXtraWay sono elencati di seguito:

- **MongoDB ver. 4.2.0** - Non è possibile utilizzare una versione precedente di MongoDB perché, per la gestione dei seriali sequenziali, vengono utilizzate le transazioni multi-documento introdotte da questa specifica versione;
- **Elasticsearch ver. 6.4.2** - Versione sulla quale è stato fatto lo sviluppo (corrispondeva alla versione più recente in fase di avvio della riscrittura del Broker);
- **Storage Server**.

Progetti

Progetto	Descrizione	Versione Richiesta
DocWay4	WebApp della view di DocWay (e ACL)	4.50.0
DocWay4-service	WebApp della business logic di DocWay (e ACL)	4.50.0
common4	Libreria common di DocWay	4.50.0
acl4	Business Logic di ACL	4.50.0
xdocway4	Business Logic di DocWay	4.50.0
apps	Dipendenza di DocWay che richiede connessioni tramite broker	4.50.0
Xway4Jasper	Integrazione di Jasper Reports all'interno di Docway	4.50.0
IRModel	Dipendenza di DocWay che richiede connessioni tramite broker	3.0.0
broker	Libreria utilizzate per l'esecuzione di comandi su eXtraWay o MongoDB/Elasticsearch	3.0.0
xway2elastic	Libreria di basso livello necessaria all'esecuzione dei comandi su Elasticsearch	2.0.0
3diws	WebApp dei WS di DocWay	3.9.0
extraway_jws	Business Logic dei 3diWS	3.9.0
extrawayxml-jws	Business Logic dei 3diWS PXML	3.9.0
DocwayWS	WS realizzati adhoc per Ader (estensione dei 3diWS)	2.2.0
DocwayIgiWS	WS di integrazione con IGI (Ader)	1.1.0
MSA	Processo MSA	3.1.0
Console MSA	Console di amministrazione di MSA	1.1.0
Dw4EquiProcesses	Applicazione di esecuzione di comandi da lanciare tramite CRON (es. reingegnerizzazione di comandi lanciati da console tramite Stored Procedure LUA, Equinotifier)	0.0.1
docway-fca	FCA per DocWay	6.1.0
docway-fcs	FCS per DocWay	6.1.0
Storage Server	Storage Server per gestione dei file	0.1.8
Libs core Storage Server	Librerie utilizzate dallo Storage Server, in strg-client-embeded è gestito il salvataggio su file system	0.1.8



Progetto	Descrizione	Versione Richiesta
Importer da eXtraWay	Comando di importazione di record da eXtraWay, allineamento dei seriali e rimappatura dei file (il percorso dei file su filesystem viene mantenuto inalterato ma vengono aggiunti i riferimenti al database dello Storage Server)	2.0.0
Mirroring da eXtraWay	Processo di Mirroring da eXtraWay	0.0.1
MongoElastic Tools	"Porting" della console eXtraWay per esecuzione di attività di manutenzione su archivio MongoDB/Elasticsearch	1.0.0
eXtraWay Tools	console eXtraWay per esecuzione di attività di manutenzione	1.0.0
xwJMeterSampler	Test di carico del broker (confronto fra eXtraWay e MongoDB/Elasticsearch) realizzato con progetto JMeter	0.0.1
docwayAccessWS	WebServices realizzati e utilizzati da Sogin (wrapper dei 3diWS)	3.0.0
mongodb2elastic-reindex	Libreria utilizzata dalla console per il reindex da MongoDB ad Elasticsearch	0.0.1
SoginSAPDocWS	Servizi esposti su SAP per integrazione SoginDoc (chiamata "terminaAssociazione")	2.0.0

In grassetto sono indicate applicazioni che possono essere installate come webapp o processi sul server.

Configurazione

it.highwaytech.broker.properties

Attraverso il file di properties del broker è possibile configurare il database da utilizzare:

- eXtraWay;
- MongoDB/Elasticsearch.

Di seguito il set di properties da configurare per l'utilizzo dell'applicazione su MongoDB/Elasticsearch.

```
#####
# Configurazione registrazione cancellazione record su eXtraWay
#####

# Se abilitata vengono registrate le cancellazioni su extraway nella nuova ud delinfo che va
aggiunta
# ai conf di tutti gli archivi (true/false, default = true)
xway.enableDeleteRegister=false

#####
# Configurazione MongoDB / Elasticsearch
#####

mongoelastic.enable=false

# Possibilita' di abilitare l'indicizzazione asincrona di documenti in fase di salvataggio
(true/false, default = false)
mongoelastic.index.async=false

##### MONGODB

# URI di connessione all'host MongoDB
#mongodb.uri=mongodb://localhost:27017/?safe=true&w=1
mongodb.uri=mongodb://127.0.0.1:27017,127.0.0.1:27018,127.0.0.1:27019/?replicaSet=rsdev

# Elenco di database supportati
mongodb.dbs=xdocwaydoc,xdocwayproc,acl

# Definizione di eventuali alias di nomi di database. Tipicamente utilizzati dai 3diWS per
apportare modifiche
# "non consone" a documenti (vengono ignorati tutti i controlli sul formato del documento di
DocWay).
# Il formato di definizione e' il seguente:
# dbalias.N=ALIAS_DBNAME|MONGODB_DBNAME
# dove:
# - N rappresenta il numero progressivo di definizione del path
# - ALIAS_DBNAME rappresenta il nome dell'alias del nome del database
# - MONGODB_DBNAME rappresenta l'effettivo nome del database su mongodb
# Esempio:
```



```
# dbalias.1=aliasdw4name|xdocwaydoc
# dbalias.2=aliasaclname|acl

# Numero massimo di tentativi di esecuzione di un salvataggio in transazione in caso di conflitti
dovuti ad
# accessi concorrenti
mongodb.transactions.max_attempts=100

# Tempo di attesa (in millisecondi) fra 2 tentativi di accesso in transazione ad un documento
(conflitto di accesso
# al record fra thread concorrenti)
mongodb.transactions.conflict_sleep=15

# Lunghezza del codice di nrecord da associare ai record salvati
mongodb.nrecord.length=10

# Validita' del LOCK su un documento (espresso in ore)
mongodb.doc_lock.max_time=24

##### ELASTICSEARCH

# Elenco di host elasticsearch: host1:port1,host2:port2,...,hostN:portN
elastic.hosts=127.0.0.1:9200
elastic.threads.count=
elastic.sniff=false
#elastic.username=
#elastic.password=
elastic.connectionRequestTimeout=30000
elastic.maxRetryTimeout=30000
elastic.socketTimeout=30000

# Gestione SSL
# N.B.: In caso di abilitazione di xPack su SSL occorre configurare il keystore contenente il
certificato per la connessione ai nodi di elasticsearch
elastic.ssl=false
#elastic.keystorePath=
#elastic.keystorePassword=
#elastic.keystoreType=

# Definisce se occorre forzare il refresh dell'indice dopo ogni attivita' di
salvataggio/cancellazione (true / false, default = false). False equivale al refresh dopo 1s.
elastic.forceRefresh=false

# Directory all'interno della quale registrare eventuali errori di indicizzazione riscontrati su
elasticsearch. Se non viene specificata alcuna directory, verra'
# creata ed utilizzata una directory 'elasticerrors' all'interno della directory dei temporanei.
#elastic.indexError.tempDir=
# Tempo (in minuti) di sleep del thread di che si occupa di ritentare il salvataggio degli indici
su Elasticsearch per i quali era stato
# riscontrato errore (default = 1 min)
elastic.indexErrorsJob.sleep=1

# Configurazione della relazione fra un archivio eXtraWay e il relativo indice su Elasticsearch.
Viene definito per ogni nome di archivio eXtraWay il nome (o alias) dell'indice
# su Elasticsearch e l'implementazione da adottare per la conversione da XML a JSON dei documenti
# xway2elastic.[XWAY_DBNAME].indexName=[ELASTIC_INDEXNAME]
# xway2elastic.[XWAY_DBNAME].mapping=[ELASTIC_INDEXNAME]
xway2elastic.xdocwaydoc.indexName=xdocwaydoc
xway2elastic.xdocwaydoc.mapping=it.tredi.xway2elastic.parser.DocWayDoc2Json
xway2elastic.xdocwayproc.indexName=xdocwayproc
xway2elastic.xdocwayproc.mapping=it.tredi.xway2elastic.parser.DocWayProc2Json
xway2elastic.acl.indexName=acl
xway2elastic.acl.mapping=it.tredi.xway2elastic.parser.Acl2Json

# Per ogni indice di Elasticsearch e' possibile definire un elenco di nomi di indici sui quali
```



```
devono essere riportate tutte le operazioni di salvataggio e cancellazione. Utile
# in fase di reindex a caldo in modo da garantire l'allineamento degli indici in caso di continuo
inserimento di documenti da parte degli operatore (il reindex su Elasticsearch utilizza
# le ricerche tramite scroll, quindi fa una fotografia dell'indice in quel momento e lo replica su
un secondo)
# xway2elastic.[ELASTIC_INDEXNAME].reindex=[ELASTIC_INDEX1],[ELASTIC_INDEX2],...,[ELASTIC_INDEXn]

#observer cleaning routine interval in secondi
observersCheckIntervalSeconds=600
#eta massima in secondi dei observer con data della chiusura in secondi [6 ore]
observersMaxAgeSeconds=21600
#eta massima in secondi dei observer senza data della chiusura in secondi [12 ore]
observersEndedMaxAgeSeconds=43200
```

Per abilitare l'implementazione su MongoDB/Elasticsearch occorre attivare la property **mongoelastic.enable**. Si consiglia di non abilitare l'indicizzazione asincrona dei documenti (*mongoelastic.index.async*) in scenari con un elevato numero di salvataggi in modo da non sovraccaricare le risorse del server Tomcat (e produrre disallineamenti fra i record salvati e quelli indicizzati).

Nel caso si vogliano definire degli alias ai database definiti in 'mongodb.dbs' occorre configurare le properties '**dbalias.n**':

```
dbalias.1=avepa|xdocwaydoc
```

L'esempio sopra permette di collegarsi al database xdocwaydoc tramite l'alias avepa. Questa tecnica viene principalmente utilizzata sui 3diWS in modo da evitare alcuni controlli software svolti dai WS in caso di salvataggi su archivio DocWay.

mongodb.config.json

File di configurazione di MongoDB attraverso il quale devono essere replicate le configurazioni eXtraWay relative a:

- storage di File;
- thesauri;
- UD Type definite sull'archivio;
- regole di univocità;
- gestione di seriali.

Configurazione Storage Server

```
"storageConfiguration": {
  "mongodbUri": "mongodb://127.0.0.1:27017,127.0.0.1:27018,127.0.0.1:27019/?replicaSet=rsdev",
  "database": "content_storage",
  "mediaCollection": "media",
  "filesystemRoot": "/opt/3di.it/platform/storageserver/media-storage",
  "filesystemRelativePathSaveFormat": "yyyy/MM/dd/HH",
  "mimeTypeAlwaysOctetStream": false,
  "mimeTypeOnlyByName": false,
  "showSlowSaveAction": true,
  "slowSaveActionMillis": 100,
  "slowSaveActionByteForMillis": 200,
  "showSlowReadAction": true,
  "slowReadActionMillis": 100,
  "slowReadActionByteForMillis": 200
}
```

I principali parametri della configurazione sono i seguenti:

- **mongodbUri** - URI MongoDB da utilizzare per la connessione allo Storage Server;
- **database** - Nome del database associato allo Storage Server;
- **filesystemRoot** - Percorso assoluto alla directory *montata* contenente i file caricati sullo storage. L'unica implementazione dello Storage supportata è quella con la registrazione dei file su FileSystem (come richiesto da specifiche ADER). Questo comporta che il file system sul quale vengono memorizzati i file debba essere condiviso con tutte le applicazioni che devono accedere ad esso (ogni istanza di broker);
- **filesystemRelativePathSaveFormat** - In caso di DatabaseConfiguration.storageSaveFileLikeExtraway=false indica il format che dalla data corrente recuperata dal objectid del medium relativo all'allegato che si stà salvando ricava il path da aggiungere alla root in cui verrà salvato il file, se lasciata vuota verrà utilizzato il default yyyy/MM/dd/HH;
- **mimeTypeAlwaysOctetStream** - Se impostato a true il mimetype viene sempre impostato a "application/octet-stream" in fase di salvataggio di un file sullo storage;
- **mimeTypeOnlyByName** - Se mimeTypeAlwaysOctetStream=false e mimeTypeOnlyByName è impostato a true il mimetype viene ricavato dal nome del file;



- **showSlowSaveAction** - Utilizzato per loggare eventuali salvataggi lenti di file sullo storage. Oltre a questo parametro occorre settare i timeout di `slowSaveActionMillis` e `slowSaveActionByteForMillis`. Il primo indica il tempo limite impiegato per il salvataggio al di sopra del quale verificare il log dell'attività, mentre il secondo il tempo minimo di salvataggio di 1 Byte (al di sopra di 200ms per Byte viene loggato il messaggio);
- **showSlowReadAction** - Simile a `showSlowSaveAction`, ma relativo al caricamento di file dallo Storage Server.

Definizione di Thesauri

```
"thesauroDataDefinitions": [  
  {  
    "uniqueName": "thxdocwaydocproc",  
    "database": "thesauroxdocwaydocproc",  
    "leftTrim": true,  
    "rightTrim": true,  
    "thesauroDefinitions": [  
      {  
        "key": 1,  
        "nome": "classif"  
      },  
      {  
        "key": 2,  
        "nome": "arrivo"  
      }  
    ]  
  }  
]
```

La porzione di JSON precedente mostra un estratto della configurazione dei Thesauri di eXtraWay su MongoDB. Di seguito alcune informazioni sulla configurazione:

- **uniqueName** - Corrisponde al nome univoco associato al thesauro;
- **database** - Nome del database MongoDB utilizzato per la registrazione dei thesauri;
- **thesauroDefinitions** - Definizione del thesauro. I valori di questa sezione devono essere recuperati e allineati al file di configurazione dei Thesauri su eXtraWay (file `.dsp`, sezione `KeyLink`).



N.B.: La gestione dei Thesauri non è stata implementata completamente su MongoDB. Sono state gestite le sole relazioni utilizzate in DocWay.

Per l'esportazione dei Thesauri da eXtraWay e la successiva importazione su MongoDB si devono utilizzare:

- la console eXtraWay (eXtraWay Tools) per l'esportazione da eXtraWay;
- la console MongoDB/Elasticsearch (MongoElastic Tools) per l'importazione su MongoDB.

Configurazioni del database

```
"databaseConfigurations": [  
  {  
    "name": "xdocwaydoc",  
    "storageSaveFileLikeExtraway": true,  
    "storageSaveFileLikeExtrawayPathOnRoot": "xdocwaydoc.file",  
    "dbConfiguration": {  
      "uniqueIndexOnProxyCollectionForShardingXmlRecordCollection": false,  
      "thesauroConfigurations": [  
        {  
          "definitionUniqueName": "thxdocwaydocproc",  
          "definitionKey": 1  
        },  
        ...  
        {  
          "definitionUniqueName": "thxdocwaydocproc",  
          "definitionKey": 5  
        }  
      ],  
      "entityConfigurations": [  
        {
```



```
"type": "doc",
"sequentials": [
  {
    "path": "/doc/@num_prot",
    "length": 7,
    "required": "false",
    "serialId": "num_protocollo"
  },
  {
    "path": "/doc/repertorio/@numero",
    "length": 7,
    "required": "false",
    "serialId": "num_repertorio"
  }
],
"uniqueRule": "[XML,/doc/@nrecord] or [XML,/doc/@num_prot]
or[XML,/doc/extra/raccomandata/@numero]",
"unique": [
  {
    "uniquePartials": [
      {"path": "/doc/@num_prot", "exists": "false"}
    ]
  },
  {
    "uniquePartials": [
      {"path": "/doc/repertorio/@numero", "exists": "false"}
    ]
  }
]
},
{
  "type": "fascicolo",
  "serials": [
    {
      "path": "/fascicolo/@numero",
      "length": 5,
      "serialId": "num_fascicolo"
    },
    {
      "path": "/fascicolo/@num_pos",
      "length": 8,
      "required": "false",
      "serialId": "fasc_pers_num_pos"
    }
  ],
  ...
}
]
}
}
```

La configurazione del database prevede i seguenti parametri:

- **name** - Nome del database (identico al nome su eXtraWay);
- **storageSaveFileLikeExtraway** - Indica se salvare gli allegati come faceva eXtraWay. Se true gli allegati verranno salvati come avveniva in extraway nelle cartelle cioè per un attachId 11232653.ext nel path filesystemRoot/storageSaveFileLikeExtrawayPathOnRoot/001123/11232653.ext. Il default è true;
- **storageSaveFileLikeExtrawayPathOnRoot** - Percorso relativo dalla root del file system dello Storage Server Prefisso per la gestione dei file dell'archivio (gestione file analoga e compatibile con eXtraWay);
- **uniqueIndexOnProxyCollectionForShardingXmlRecordCollection** - Necessario al calcolo dei seriali in caso di sharding della collection XmlRecord. Se si pensa che il database dovrà essere configurato in sharding questa chiave deve essere impostata a TRUE dal primo istante, altrimenti può essere lasciata a FALSE (non vengono utilizzate le proxy collection per il calcolo dei seriali);
- **thesauroConfigurations** - Thesauri utilizzati sul database;
- **entityConfigurations** - Definizione delle entity utilizzate (UDType di eXtraWay)



- **type** - Tipo di entity, corrisponde al valore di UD Type di eXtraWay;
- **sequentials** - Definizione dei seriali di sequenza (per i quali non devono essere lasciati buchi nella numerazione)
- **serials** - Definizione di seriali "semplici". E' sufficiente che il valore sia univoco, sono ammessi buchi nella numerazione
- **uniqueRule** - Regola di univocità in formato eXtraWay
- **unique** - Regola di univocità espressa in formato JSON per controlli su MongoDB

Di seguito vengono riportati i parametri di configurazione dei seriali:

- **path** - XPath XML associato al seriale;
- **prefix** - Eventuale prefisso da assegnare al seriale (ad esempio PE per le Persone Esterne);
- **length** - Lunghezza del seriale;
- **required** - TRUE se il valore è richiesto sul documento, FALSE altrimenti (seriale opzionale);
- **serialId** - Identificativo del seriale.



Se occorre impostare lo sharding sul database MongoDB è necessario impostare a TRUE il valore di **uniqueIndexOnProxyCollectionForShardingXmlRecordCollection**. Questa attività può essere fatta solo in fase di prima definizione dell'ambiente (la configurazione non potrà essere variata "a caldo" su MongoDB).

elasticsearch.config.json

Attraverso il file elasticsearch.config.json vengono configurate tutte le informazioni necessarie all'esecuzione di query sui documenti indicizzati su Elasticsearch. Di seguito un estratto del file JSON di configurazione:

```
{
  "databaseConfigurations": [
    {
      "db": "xdocwaydoc",
      "title": "XML,/doc/classif/@cod XML,/doc/oggetto XML,/doc/rif_esterni/rif/nome
XML,/doc/autore XML,/doc/@bozza XML,/doc/@nrecord XML,/doc/@data_prot XML,/doc/@num_prot
XML,/fascicolo/@numero XML,/raccoglitore/oggetto XML,/doc/scadenza/@data_scadenza
XML,/doc/repertorio/@numero UD,/xw/@UdType XML,/fascicolo/oggetto \"|sort|\" \"^\" XML,/doc/@tipo
\"^\" \"|epn|\" \"^\" XML,/doc/@nrecord \"^|\" XML,/doc/@num_prot \"^|\"
XML,/doc/repertorio/@numero \"^|\" XML,/doc/@data_prot \"^|\" XML,/doc/rif_esterni/rif/nome
XML,/doc/autore XML,/doc/minuta/mittente/@nome_persona \"^ (~\" XML,/doc/minuta/mittente/@nome_uff
\"~^\" XML,/doc/proposta/proponente/@nome_persona \"^ (~\" XML,/doc/proposta/proponente/@nome_uff
\"~^\" XML \"^|\" XML,/doc/oggetto \"^|\" \"^_DIFF_~\" XML,/doc/prot_differito/@data_arrivo \"^|\"
XML,/fascicolo/@numero \"^|\" XML,/fascicolo/oggetto \"^|\" XML,/fascicolo/@stato \"^|bozza=\"
XML,/doc/@bozza \"^|\" XML,/doc/repertorio \"^|annullato=\" XML,/doc/@annullato \"^|\"
XML,/raccoglitore/oggetto \"^|\" XML,/raccoglitore/@stato \"^|\" XML,/seduta/@data_convocazione
\"^|\" XML,/seduta/@straordinaria \"^|*COS*\" XML,/seduta/organo/@cod \"^|*SS*\"
XML,/seduta/@stato \"^|\" XML,/doc/societa \"^|\" XML,/doc/extra/@agli_atti \"^|D!~\"
XML,./rif_interni/rif/@diritto \"^|DP!~\" XML,./rif_interni/rif/@cod_persona \"^|DU!~\"
XML,./rif_interni/rif/@cod_uff \"^|DF!~\" XML,./rif_interni/rif/@cod_fasc \"^|RC!~\"
XML,/doc/repertorio/@cod \"^|V!~\" XML,/doc/visibilita/@tipo \"^|VA!~\"
XML,/doc/visibilita/@fino_al \"^|F!~\" XML,/doc/files/xw:file/@name(first) \"^|I!~\"
XML,/doc/immagini/xw:file/@name(first) \"^|R!~\" XML,/raccoglitore/@pubblico \"^|FC!~\"
XML,/fascicolo/classif/@cod \"^|VL!~\" XML,/doc/@validato \"^|P!~\" XML,/doc/personalView/@cod
\"^|DT!~\" XML,/doc/@tipo \"^|U!~\" UD,/xw/@UdType \"^|S!~\" XML,/doc/@cod_amm_aoo \"^|S!~\"
XML,/fascicolo/@cod_amm_aoo \"^|S!~\" XML,/raccoglitore/@cod_amm_aoo \"^|PS!\" XML,/doc/@personale
\"^|DUN!~\" XML,./rif_interni/rif/@nome_uff \"^|0xSS!\" XML,/doc/odg_seduta/@stato \"^|0xP0!~\"
XML,/doc/proposta/@cod_organo \"^|SCP!~\" XML,/doc/storia/*/@cod_persona \"^|SVD!~\"
XML,/doc/storia/*/@visto_da \"^|SPR!~\" XML,/doc/storia/creazione/@cod_oper \"^|SPPR!~\"
XML,/doc/storia/protocollazione/@cod_oper \"^|SPRU!~\" XML,/doc/storia/creazione/@cod_uff_oper
\"^|SPPRU!~\" XML,/doc/storia/protocollazione/@cod_uff_oper \"^|RS!~\" XML,/raccoglitore/@stato
\"^|NR!~\" XML,/raccoglitore/@nrecord \"^|0xST!~\" XML,/doc/proposta/@delib_seduta_stante
\"^|DS!~\" XML,/doc/scadenza/@data_scadenza \"^|TRASH!~\" XML,/doc/@cestino \"^|ST!~\"
XML,/doc/@stato \"^|DPN!~\" XML,./rif_interni/rif/@nome_persona \"^|\"
xml,/doc/mezzo_trasmissione/@cod \"^|\" xml,/doc/mezzo_trasmissione/@costo \"^|SPRN!~\"
XML,/doc/storia/creazione/@oper \"^|SPRNU!~\" XML,/doc/storia/creazione/@uff_oper \"^|CONSID!~\"
XML,/doc/extra/conservazione/@id \"^|CONSERV!~\" XML,/doc/extra/conservazione/@servizio
\"^|DEP!~\" XML,/doc/extra/deposito/@stato \"^|DEPM!~\" XML,/doc/extra/deposito_minuta/@stato
\"^|\" XML,/doc/extra/richiedente/nome
XML,/doc/extra/tipologie_repertorio/tipologia_repertorio",
```



```

    "alias": [
      {
        "name": "doc_ordinelicenza",
        "path": "/doc/ordine/licenza"
      },
      {
        "name": "doc_pubblicazionepreavvisodal",
        "path": "/doc/pubblicazione/@preavviso_dal"
      },
      {
        "name": "doc_xlinkhref",
        "path": "/doc/xlink/@href"
      },
      ...
    ],
    "fieldsDouble": [
      { "path": "/doc/storia/creazione/@oper" },
      { "path": "/doc/storia/creazione/@uff_oper" },
      ...
    ],
    ...
  ]
}

```

Di seguito la descrizione delle principali configurazioni:

- **db** - Nome dell'indice su Elasticsearch (o meglio dell'alias). Deve corrispondere al nome del database su MongoDB;
- **title** - Regola di composizione del titolo per l'archivio. Deve essere recuperata dal file conf di archivio su eXtraWay;
- **alias** - Elenco degli alias di ricerca definiti sull'archivio. Anche questi valori sono definiti nel file conf dell'archivio eXtraWay;
- **fieldsDouble** - Elenco dei field configurati come double su archivio extraway, per i quali viene automaticamente attivata una modalità di ricerca ad-hoc in caso ad esempio di lookup.

cache query

Una delle principali differenze fra l'implementazione eXtraWay e quella MongoDB/Elasticsearch riguarda la gestione delle query.

Su eXtraWay ogni volta che si fa una ricerca viene fatta una “fotografia” dell'archivio e vengono restituiti come risultati i soli documenti trovati in quell'istante (selezione di documenti). Su MongoDB ed Elasticsearch, invece, la query di ricerca viene eseguita ogni volta che si richiede una pagina di documenti (eventuali documenti nuovi possono essere aggiunti ai risultati della ricerca ed altri record possono non essere più presenti perché eliminati). Ad ogni invocazione della pagina dei titoli, eXtraWay si aspetta come parametro di input l'identificativo della selezione, mentre Elasticsearch (come anche MongoDB) la query con relativa regola di sort. Per garantire la retrocompatibilità del codice e non dover variare la logica applicativa si è scelto di emulare (fin quanto possibile) il comportamento delle selezioni direttamente sul broker. Ogni volta che viene fatta una query in formato eXtraWay si procede come segue:

- Viene convertita la query dal formato eXtraWay a quello JSON di Elasticsearch;
- Viene convertita l'eventuale regola di sort in formato Elasticsearch;
- Viene eseguita la query su Elasticsearch;
- Viene generato un identificativo univoco di selezione e associato alla query (ed eventuale regola di sort);
- Viene memorizzata su una cache locale lo stato della ricerca (associato all'identificativo della selezione generato).

Ad ogni richiesta di titoli vengono recuperate dalla cache le informazioni sulla query da eseguire in base all'identificativo di selezione ricevuto e si procede con la chiamata su Elasticsearch.

Al momento si sta utilizzando *Apache Commons JCS* per la gestione della CACHE locale delle selezioni. Per maggiori dettagli si rimanda alla [documentazione ufficiale](#)

Il file da configurare per la gestione della cache è **cache.ccf**.

Database

MongoDB

La creazione dei database MongoDB è gestito direttamente dall'applicazione (a livello di broker) così come la definizione di tutti gli indici necessari.

Di seguito vengono riportati alcuni dettagli su collection o struttura dei dati.

Le collection definite sul database sono 3:

- **xmlrecords** - All'interno della collection vengono registrati tutti i record XML dell'archivio;
- **hierarchy** - Collection utilizzata per il mantenimento delle relazioni di gerarchia fra record dell'archivio (es. gerarchia di



strutture interne in ACL);

- **serials** - Collection utilizzata per la generazione dei seriali definiti sull'archivio.

Delle 3 collection, sicuramente quella con una struttura più complessa è xmlrecords. Di seguito viene analizzata la struttura JSON definita per la memorizzazione dei documenti XML su MongoDB.

```
{
  "_id" : ObjectId("5f86f107593fd2116fb5a29f"),
  "content" : "<?xml version='1.0' encoding='UTF-8'>\n<doc
xmlns:xw='http://www.3di.it/ns/xw-200303121136' anno='2020' annullato='no' bozza='no'
cod_amm_aoo='3DINBOL' data_prot='20200904' nrecord='00310774'
num_prot='2020-3DINBOL-0000028' sensibile='no' tipo='arrivo'>...</doc>",
  "insDate" : ISODate("2020-10-14T12:37:27.550Z"),
  "joinField" : "parent",
  "nrecord" : "00310774",
  "physDoc" : 200,
  "type" : "doc",
  "unique" : {
    "doc__num_prot" : "2020-3DINBOL-0000028"
  },
  "updDate" : ISODate("2020-10-14T12:37:27.550Z"),
  "version" : 0,
  "lock" : {
    "time" : ISODate("2020-10-20T12:31:44.523Z"),
    "lockKey" : "nbijxojk2477538dd5e21324e7cecd3ff9ee508",
    "userId" : "sstagni"
  }
}
```

Di seguito la descrizione dei campi che compongono il record:

- **_id** - Identificativo del record su MongoDB (ObjectID);
- **content** - Contenuto XML del record;
- **type** - Tipologia di record (UD Type);
- **insDate** - Data e Ora di creazione del record;
- **updDate** - Data e Ora di ultima modifica del record;
- **version** - Numero di versione del record;
- **nrecord** - NRecord associato al documento XML (importato da eXtraWay o gestito come seriale su MongoDB);
- **physDoc** - Numero fisico del documento (non importato da eXtraWay). Viene generato come seriale su MongoDB e, a differenza dell'implementazione eXtraWay, non varia;
- **joinField** - Valore fisso 'parent' utilizzato per il join con i file allegati al record (ricerca per contenuto estratto dai file);
- **unique** - Elenco di tutti i campi del record soggetti a regole di univocità
- **lock** - Blocco opzionale che identifica un documento bloccato in modifica da un operatore
 - **time** - Data e Ora di blocco del documento
 - **lockKey** - Chiave di lock del documento
 - **userId** - Utente che ha bloccato il documento

Elasticsearch

A differenza di MongoDB, gli indici su Elasticsearch devono essere generati obbligatoriamente prima dell'avvio dell'applicazione. Questa attività consiste nella definizione della struttura dati e della modalità di indicizzazione dei vari campi che compongono l'archivio.

Nel progetto **archivi_conf_xml** è presente una **directory** con i file di mapping di default per gli archivi docway:

- [xdocwaydoc](#)
- [xdocwayproc](#)
- [acl](#)



Si consiglia di creare gli indici utilizzando un nome differente da quello del database MongoDB (banalmente è sufficiente aggiungere al nome dell'archivio il suffisso con la data corrente). In questo modo, l'archivio xdocwaydoc avrà come indice associato xdocwaydoc_20201020. A questo indice andrà poi associato un alias avente lo stesso nome dell'archivio. In questo modo sarà possibile effettuare un reindex dell'archivio e spostare l'alias xdocwaydoc sul secondo indice creato.



Migrazione da eXtraWay

La procedura di migrazione dati da eXtraWay a MongoDB/Elasticsearch può essere fatta secondo 2 differenti modalità:

1. *Procedura di Importazione.* Set di attività che permettono (a sistema spento) di migrare tutti i dati da eXtraWay a MongoDB/Elasticsearch e riattivare l'applicazione DocWay e i servizi configurati sui nuovi Database;
2. *Procedura di Mirroring.* Utilizzo dell'applicazione su eXtraWay e migrazione dei dati "real time" per allineamento di MongoDB/Elasticsearch. Con questa modalità il passaggio da eXtraWay a MongoDB/Elasticsearch si può realizzare con tempi di stop del sistema più ridotti (minori disservizi per gli utenti).

Importazione

Di seguito gli step da seguire per migrare DocWay da eXtraWay a MongoDB/Elasticsearch tramite la procedura di importazione dati:

1. Installazione/Configurazione di MongoDB/Elasticsearch (definizione repliche, sharding, etc.);
2. Stop dell'applicazione DocWay e di tutti i servizi a supporto (3diWS, MSA, etc.);
3. Creazione su Elasticsearch degli indici relativi agli archivi. Su [archivi_conf.xml](#) è presente una configurazione di default che va comunque adattata alla specifica configurazione del conf dell'archivio eXtraWay utilizzato dal cliente (es. aggiunta di eventuali campi extra gestiti sull'installazione del cliente e che devono essere mappati in un modo particolare su Elasticsearch);
4. Avvio da shell del processo di [importazione dati da eXtraWay a MongoDB/Elasticsearch](#) (comando `xway2elastic`, si consiglia di provare varie configurazioni su un campione di dati al fine di trovare la combinazione ottimale di thread in modo da minimizzare i tempi di importazione);
5. Al termine dell'importazione occorre procedere con la "migrazione" dei file, ovvero la registrazione su archivio MongoDB dello Storage Server di tutti i file registrati su eXtraWay (comando `fileimport` su progetto [xway2elastic-import](#));
6. Allineamento dei seriali fra i 2 sistemi (comando `ser2mongo` su progetto [xway2elastic-import](#));
7. Configurazione di tutti gli applicativi e servizi installati presso il cliente (file broker, file di configurazione di MongoDB ed Elasticsearch, gestione cache, etc.)
 1. In questa fase occorre montare la directory dei file dello Storage Server come directory di rete su tutti i server sui quali sono installati servizi connessi al database tramite broker. Questo è necessario perché ogni singolo broker include un client dello Storage Server che si occupa in autonomia di registrare il file su file system e il record sull'archivio dello Storage Server (per ovvie esigenze di performance ed eliminazione di colli di bottiglia).
8. Stop del servizio eXtraWay. In questo modo in caso di mancato aggiornamento di alcuni applicativi non verranno effettuate scritture sul vecchio database;
9. Avvio di tutti i servizi;
10. Test di funzionamento dell'ambiente.

Mirroring

La procedura di mirroring è più articolata rispetto a quella di import visto che la include parzialmente. Per identificare i documenti da registrare in fase di mirror viene fatta una ricerca su eXtraWay di tutti i documenti inseriti/modificati dall'ultima esecuzione del thread di mirror. Per come è implementato eXtraWay questi dati vengono resettati alla data e ora correnti in caso di reindex dell'archivio, quindi l'esecuzione del mirror comporterebbe il recupero dell'intero archivio al primo avvio del thread. Di seguito è riportato l'iter consigliato di attività.

Avvio del servizio di Mirroring

1. Installazione/Configurazione di MongoDB/Elasticsearch (definizione repliche, sharding, etc.);
2. Stop dell'applicazione DocWay e di tutti i servizi a supporto (3diWS, MSA, etc.);
3. Creazione su Elasticsearch degli indici relativi agli archivi. Su [archivi_conf.xml](#) è presente una configurazione di default che va comunque adattata alla specifica configurazione del conf dell'archivio eXtraWay utilizzato dal cliente (es. aggiunta di eventuali campi extra gestiti sull'installazione del cliente e che devono essere mappati in un modo particolare su Elasticsearch);
4. Avvio da shell del processo di [importazione dati da eXtraWay a MongoDB/Elasticsearch](#) (comando `xway2elastic`, si consiglia di provare varie configurazioni su un campione di dati al fine di trovare la combinazione ottimale di thread in modo da minimizzare i tempi di importazione);
5. Avvio della procedura di "migrazione" dei file, ovvero la registrazione su archivio MongoDB dello Storage Server di tutti i file registrati su eXtraWay (comando `fileimport` su progetto [xway2elastic-import](#));
6. Esecuzione del servizio di mirror sull'archivio eXtraWay. Nella configurazione iniziale indicare la data e ora di avvio del mirror alla data corrente;



Per evitare che vengano registrati erroneamente dei documenti sugli archivi di destinazione del mirroring si sconsiglia l'installazione di una copia dell'applicativo che punti a MongoDB/Elasticsearch. Per controllare l'avanzamento/stato dell'attività occorre controllare i log del servizio ed eventualmente consultare i database MongoDB ed Elasticsearch tramite query adhoc.



Switch su MongoDB/Elasticsearch

1. Stop dell'applicativo DocWay e servizi a corredo (nessuna nuova scrittura su eXtraWay);
2. Verifica dai log del servizio di mirroring che tutte le attività risultino completate (completo allineamento degli archivi fra i 2 ambienti);
3. Stop del servizio di mirroring;
4. Stop del servizio eXtraWay;
5. Allineamento dei seriali fra i 2 sistemi (comando *ser2mongo* su progetto [xway2elastic-import](#));
6. Configurazione di tutti gli applicativi e servizi installati presso il cliente (file broker, file di configurazione di MongoDB ed Elasticsearch, gestione cache, etc.)
 1. In questa fase occorre montare la directory dei file dello Storage Server come directory di rete su tutti i server sui quali sono installati servizi connessi al database tramite broker. Questo è necessario perché ogni singolo broker include un client dello Storage Server che si occupa in autonomia di registrare il file su file system e il record sull'archivio dello Storage Server (per ovvie esigenze di performance ed eliminazione di colli di bottiglia).
7. Avvio di tutti i servizi;
8. Test di funzionamento dell'ambiente.