ElasticSearch River

Il meccanismo che si occupa di indicizzare i dati dei nodi in modalità pull è implementato attraverso un plugin di ElasticSearch di tipo River. In ES un river è identificato da due coordinate nome e tipo. Il tipo è il plugin è il nome del plugin, nell'applicazione `rest`, ed il nome identifica l'istanza del river. In una installazione di ES possono esserci più istanze dello stesso tipo river sotto nomi diversi ed ES si occupa di far si che ogni istanza (identificata dalla coppia nome-type) sia gestita come singleton nel cluster ES.

Nel setup dell'applicazione sarà presente un'istanza di river indipendente per ogni nodo da indicizzare. Il nome (una stringa arbitraria) sarà l'identificativo del nodo che indicizza.

Nella documentazione seguente sono usate le seguenti variabili

- `$SERVER_URL` fa riferimento al base path dell'applicazione server così come configurata in JNDI
- `$ES_URL` fa riferimento al path base della servlet che gestisce le richieste per ElasticSearch (quindi il suo valore sarà `$SERVER_URL/search`)

Ad un river è associato un documento ES (quindi un documento json) che può contenere settings e/o stato per il river in questione. Il documento è accessibile all'url `/_river/<nome>/_meta` e può essere visualizzato con il comando

```
curl -XGET $ES_URL/_river/$NAME/_meta
```

L'istanza del river viene fatta partire creando tale documento passandogli il json che contiene il tipo ed eventuali settings per il nodo in questione, ad esempio per far partire il river di tipo `rest` con il nome `rest_river` il comando da eseguire è

```
curl -XPUT $ES_URL/_river/rest_river/_meta -d '{
  "type": "rest",
  "settings": { "foo": "bar", "timestamp": 1375373914000 }
}'
```

Per fermare e deregistrare un river è sufficiente cancellare il documento ad esso associato con il comando

```
curl -XDELETE $ES_URL/_river/rest_river/
```

Infine lo status nel cluster di una determinata istanza (ad esempio su quale nodo gira) è accessibile con il comando

```
curl -XGET $ES_URL/_river/rest_river/_status
```

Eventuali impostazioni dell'istanza possono essere cambiate a runtime aggiornando il documento `_meta` ad esso associato. Le nuove impostazioni hanno effetto nell'applicazione nel giro di un minuto.

RestRiver

L'implementazione del river per l'indicizzazione dei nodi interni è chiamata RestRiver e si occupa della indicizzazione dei dati presenti in un nodo che espone i servizi REST documentati nel progetto.

Il nodo da indicizzare va configurato tramite l'interfaccia json descritta in precedenza.

Strategie di indicizzazione

Il river può usare varie strategie per recuperare i dati dal nodo di cui è responsabile. Queste strategie sono le seguenti

- `oneshot` : Recupera tutti i dati in una singola esecuzione e poi si disattiva
- `all` : Recupera tutti i dati ad intervalli periodici configurabili tramite settings e con due possibili modalità, tutti i dati in un'unica chiamata rest oppure usando la paginazione.
- `versioned` : Il nodo che usa questa strategia deve esporre un endpoint che ritorna il numero di versione corrente dei dati come stringa (ad esempio uno sha1 di tutto il dataset). Ad ogni retrieve completo dei dati viene associato il numero di versione corrente e prima del successivo retrieve dei dati il plugin richiede la versione corrente. Successivamente i dati vengono richiesti per una nuova indicizzazione se, e solo se, la versione corrente è cambiata rispetto alla versione salvata localmente associata al dataset.
- `timestamped` : Il nodo che utilizza questa strategia deve supportare nell'operazione di retrieve un parametro timestamp e ritornare nella risposta solamente i record aggiunti, modificati o aggiornati dopo l'istante temporale passato come parametro della chiamata.

Documento `_meta` per RestRiver

Segue un esempio di documento `_meta` associato ad una istanza del river

```
{
    "type": "rest",
    "settings": {
        "source_name": "Idra",
        "base_url": "http://localhost:8080/node",
        "target_index": "idra_info",
        "strategy": "oneshot",
        "poll_interval": 15000,
        "paged": false,
        "page_size": 1000
    },
    "state": {
        "updated_at": 12345565948954,
        "current_version": "6cda5831c75f2715dac9ad37d290233698361cbc"
    }
}
```

Non tutti i campi sono sempre presenti o valorizzati, di seguito la descrizione dei campi del documento:

- `type` : metadato interno di ES che indica il tipo di river
- `settings` : oggetto che mantiene le impostazioni per l'istanza
- `source` : informazioni sulla sorgente da indicizzare
- `source.name` : nome della sorgente
- `source.base_url` : url di base degli endpoint rest
- `strategy` : informazioni e configurazioni della strategia da utilizzare
- `strategy.name` : nome della strategia utilizzata
- `strategy.poll_interval` : intervallo in secondi tra una indicizzazione e la successiva
- `strategy.paged` : se il nodo supporta la paginazione e questo valore è true il fetch verrà eseguito usando la paginazione
- `strategy.pagesize` : se la paginazione è attiva indica la dimensione delle pagine da usare
- `target.index` : l'indice interno dove memorizzare i dati per il nodo

L'oggetto state è utilizzato dal river per mantenere le informazioni sullo stato di indicizzazione del nodo. Tale oggetto non va modificato ed eventuali modifiche verranno sovrascritte dal river stesso durante l'esecuzione.

Configurazione predefinita nodo interno

L'applicazione nodo espone i webservices di base pertanto viene indicizzato utilizzando le seguenti settings

```
{
  "type": "rest",
  "settings": {
    "source_name": "idra.info",
    "base_url": "<JNDI value for idraRest/nodeServiceBaseUrl>",
    "target_index": "node_idra",
    "strategy": "all",
    "poll_interval": 86400,
    "paged": true,
    "page_size": 1000
  }
}
```

Tali impostazioni sono precaricate dall'applicazione stessa ad ogni avvio in modo che l'indicizzazione del nodo interno venga effettuata una volta al giorno per ricevere gli ultimi aggiornamenti. Al momento tale impostazione è preconfigurata all'interno dell'applicazione stessa. Secondo necessità sarà possibile esternalizzare tale configurazione.