



Tot plugin AQTIVA

**4.4.2**

<b>Control de versiones</b>	<b>2</b>
<b>Modelo de integración</b>	<b>2</b>
Plantilla Dataset	3
Plantilla AQTIVA_RULE	3
<b>Credenciales requeridas</b>	<b>5</b>
<b>Despliegue</b>	<b>5</b>
<b>Configuración</b>	<b>6</b>
Tot	7
<b>Funcionamiento</b>	<b>7</b>
Plugin Aqativa	7
Última fecha de ejecución	7
Sincronización de metadatos entre Aqativa y Anjana	7
Sincronización de resultados	8
Endpoints disponibles	8
Tot	9
Endpoints disponibles	9
Kerno	9
Actualización de metadatos	9
Actualización de resultados	10
Endpoints disponibles	11

## Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	30/03/2023	Anjana Producto	Anjana Producto	Creación del documento
2.0	26/04/2023	Anjana Producto	Anjana Producto	Renombrado del documento a versión correcta

# Modelo de integración

El proceso consta de la petición de forma periódica a Aqtvia de las reglas vivas existentes desde una determinada fecha y los resultados de las mismas. Se enviará a Anjana esa información para actualizar los datasets asociados a los dataformats informados así como las reglas asociadas a ellos.

Para que el proceso funcione correctamente se deben crear unos atributos predefinidos en la plantilla del dataset y crear un nuevo tipo de entidad Aqtvia\_Rule también con unos atributos predefinidos en el que se almacenarán los datos relativos a las reglas de Aqtvia (metadatos y resultados).

## Plantilla Dataset

Los atributos que hay que añadir a la plantilla del dataset son los siguientes:

Label	Nombre Atributo	Tipo	Valor
AqtvialdDataFormat	AqtvialdDataFormat	INPUT_TEXT	Contendrá el identificador del dataformat
AqtviaDataFormatName	AqtviaDataFormatName	INPUT_TEXT	Contendrá el nombre del dataformat
AqtviaQRulesArray	AqtviaQRulesArray	ARRAY_ENTITY	Contendrá un array de ARIs con las reglas asociadas
AqtviaKPILastUpdateDate	AqtviaKPILastUpdateDate	INPUT_TEXT	Fecha de Última Ejecución del KPI de resultados a nivel de datasource
AqtviaDataQualityDatasourceLevel	AqtviaDataQualityDatasourceLevel	TEXT_AREA	Contendrá los resultados a nivel del datasource

NOTA: Los label de esta tabla son nombres recomendados para el entendimiento del funcionamiento de la integración.

## Plantilla AQTIVA\_RULE

Para poder contener la información relativa a las reglas ejecutadas se necesita crear la entidad nativa **AQTIVA\_RULE** que contendrá el metadato relativo a las mismas junto con los resultados aplicados. Para ello creamos una plantilla con los siguiente atributos

Label	Nombre Atributo	Tipo	Valor
AqtvialdProject	AqtvialdProject	INPUT_TEXT	Id del proyecto en Aqtvia
AqtviaProjectName	AqtviaProjectName	INPUT_TEXT	Nombre del proyecto en Aqtvia
AqtvialdQualityPoint	AqtvialdQualityPoint	INPUT_TEXT	Id del Quality Point en Aqtvia

AqtvQualityPointName	AqtvQualityPointName	INPUT_TEXT	Nombre del Quality Point en Aqtv
AqtvIdDataSource	AqtvIdDataSource	INPUT_TEXT	Id del Datasource en Aqtv
AqtvDataSourceName	AqtvDataSourceName	INPUT_TEXT	Nombre del Datasource en Aqtv
AqtvIdQualityRule	AqtvIdQualityRule	INPUT_TEXT	Id de la regla en Aqtv
AqtvQualityRuleName	AqtvQualityRuleName	INPUT_TEXT	Nombre compuesto de la regla (Proyecto, Quality Point, DataSource)
AqtvIdDataFormat	AqtvIdDataFormat	INPUT_TEXT	Id del dataformat en Aqtv
AqtvAnjanaDatasets	AqtvAnjanaDatasets	ARRAY_ENTITY	Dataset/s sobre el/los que se Ejecuta la Regla
AqtvDataFormatSource	AqtvDataFormatSource	INPUT_TEXT	Tipo Dataset
AqtvQualityRuleActive	AqtvQualityRuleActive	INPUT_CHECKBOX	Indica si la regla está activa para hacer una simulación
AqtvQualityRuleDimension	AqtvQualityRuleDimension	INPUT_TEXT	Dimensión
AqtvQualityRuleColumns	AqtvQualityRuleColumns	INPUT_TEXT	Columnas sobre las que Aplica
AqtvQualityRuleConditions	AqtvQualityRuleConditions	INPUT_TEXT	Condiciones de la Regla
AqtvQualityRuleCreationDate	AqtvQualityRuleCreationDate	INPUT_DATE	Fecha de Creación de la Regla en Aqtv
AqtvQualityRuleWarnThType	AqtvQualityRuleWarnThType	INPUT_TEXT	Tipo de Umbral de Warning
AqtvQualityRuleWarnThValue	AqtvQualityRuleWarnThValue	INPUT_DECIMAL	Valor de umbral de Warning
AqtvQualityRuleErrType	AqtvQualityRuleErrType	INPUT_TEXT	Tipo de Umbral de error
AqtvQualityRuleErrThValue	AqtvQualityRuleErrThValue	INPUT_DECIMAL	Valor umbral de error
AqtvRecordIdRecordLastExec	AqtvRecordIdRecordLastExec	INPUT_TEXT	Id Registro de Ejecución Aqtv
AqtvLastResultSynchronizationDateAnjana	AqtvLastResultSynchronizationDateAnjana	INPUT_DATE	Fecha de Última Sincronización
AqtvRuleExecutionResults	AqtvRuleExecutionResults	TEXT_AREA	Resultado en representación json. Uso interno
AqtvRuleExecutionResultsView	AqtvRuleExecutionResultsView	ENRICHED_TEXT_AREA	Presentación de los resultados del atributo AqtvRuleExecutionResults
finishDate	finishDate	INPUT_DATE	Fecha de expiración para la regla

NOTA: Los label de esta tabla son nombres recomendados para el entendimiento del funcionamiento de la integración.

Recomendamos que los atributos arriba indicados se configuren todos con las siguientes restricciones:

- NO EDITABLES para evitar la edición errónea de un atributo ya que los mismos se actualizan de forma automática.
- No permitir la deprecación manual del objeto AQTIVA\_RULE para evitar tener inconsistencias en la bbdd y lo enviado por Aqiva.

El plugin no tiene ninguna ARI definida ya que no se va a poder hacer extracción, sampleo ni gobierno activo de estructuras, la comunicación se desencadena directamente desde el Plugin hacia Anjana con los datos a actualizar.

## Credenciales requeridas

Las credenciales para acceder a Aqiva están almacenadas en el fichero de configuración en base64 y serán proporcionadas por Aqiva. El plugin irá a buscar la siguiente configuración:

```
totplugin:  
  aqiva:  
    auth:  
      client-id: <clientId a sustituir>  
      secret: <secret a sustituir>
```

## Despliegue

Se ha de seguir el manual genérico del despliegue de plugins.

Doc: Anjana Data 4.4 - DOC - Tot despliegue de plugins

Se necesita la instalación de certificado de Aqiva:

- Se obtiene de la URL que hay configurada en el YAML de la API de Aqiva que se configura en el apartado [Configuración](#).
- Se sube a la máquina contenedora de la instancia del plugin.
- Se registra el certificado subido con el comando “keytool”, teniendo en cuenta que en el primer parámetro hay que poner la versión de Java que usa (sacada del descriptor de servicio del plugin):

```
keytool -trustcacerts -keystore  
"<path_to_jdk>/jre/lib/security/cacerts" -storepass  
changeit -importcert -alias <alias_to_be_stored> -file  
"<path_to_certificate/certificate.crt"
```

- El certificado usa DNS y no IP, por lo que la propiedad “totplugin.aqiva.auth.url” marcada en [Configuración](#) debe de ser un DNS.

# Configuración

Aquí se incluye el detalle de la configuración específica del plugin. Para revisar la configuración común mirar el documento Anjana Data 4.4 - DS - Configuración técnica de Portal y microservicios

Configuraciones específicas:

- **scheduling:** Configuración del batch de sincronización contra Aqiva. Por ejemplo de lunes a viernes entre las 07:00 y las 22:00 cada minuto

```
0 */1 7-22 * * MON-FRI
```

- **url:** Url del API de Aqiva
- **auth.url:** Url para la autenticación de Aqiva. No es un flujo oAuth2 sino simplemente un endpoint que nos da un token.
- **enable-certs-validation:** Flag que indica si el plugin hace la validación de los certificados en una conexión segura (https). **Por defecto está a true y en producción NO se debe de cambiar este valor..**
- **synchronization.delta:** Determina un delta que se añadirá a la fecha de última ejecución del plugin de cara a hacer las peticiones a Aqiva. Si no se especifica no se añadirá ningún valor.
- **synchronization.unit:** Determina la unidad de tiempo (milisegundos, segundos etc) que se aplicará al delta. Si no está especificado no se añadirá ningún valor a la fecha de última ejecución. El formato que debe de tener es específico y si es correcto fallará el proceso, siendo los posibles valores (existen más pero no se consideran útiles):
  - MILLIS
  - SECONDS
  - MINUTES
  - HOURS
  - DAYS
  - HALF\_DAYS (Equivalente a 12h)

Configuración de ejemplo:

```
logging:
  level:
    com.anjana: DEBUG

server:
  port: 15020

totplugin:
  location: http://localhost:15020/plugin/aqiva/api/v1
  server:
    url: http://localhost:15000/tot/
    keep-alive-seconds: 300
  aris:
```

```
- ari: "anja:totplugin:extract:/jdbc/aqtvadata/devQA"
aqtiva:
  enable-certs-validation: false
  url: https://aqtvatasdemoapp.westeurope.cloudapp.azure.com:4443/anjana
  auth:
    url:
      http://aqtvatasdemoapp.westeurope.cloudapp.azure.com:2018/public/auth/login
      client-id: YW5qYW5hLnRlc3RAYXF0aXZhLmlh
      secret: QW5qYW5hQXF0aXZhMjAyM190ZXN0Xyox
  scheduling: 0 */1 7-22 * * MON-FRI
  synchronization:
    delta: 5
    unit: SECONDS # ChronoUnit format: MILLIS, SECONDS, MINUTES, HOURS, DAYS
```

## Tot

Tot persiste el momento de la última ejecución correcta para poder hacer posteriormente una consulta más acotada de los datos de Aqtvata. Para ello se realiza una conexión a una bbdd perteneciente a Tot con las siguientes características:

- schema: tot
- tabla: plugin\_execution
  - name como nvarchar(255) Primary Key
  - date como timestamp

# Funcionamiento

## Plugin Aqtvata

El plugin va a disponer de un batch que se ejecutará según la configuración especificada. El proceso que ejecuta consta de 3 partes:

### Última fecha de ejecución

El proceso obtendrá la última fecha de ejecución sobre la que realizar la petición a Aqtvata. Para ello hace una petición a un endpoint en Tot que le da esta información. Si la respuesta de Tot es correcta se comprueba el delta configurado y, si existiera, se aplica siendo ésta la fecha resultante a hacer la petición.

Si la fecha devuelta por Tot no fuera correcta, ya sea porque no está establecida todavía o porque se ha producido un error, la fecha resultante será la fecha actual menos un año.

### Sincronización de metadatos entre Aqtvata y Anjana

Pide los datos vivos de Aqtvata desde una fecha determinada y pide todos los datos de Anjana relacionados con Aqtvata. Esta fecha de petición de los datos debe ser una fecha antigua para que Aqtvata nos de todas las reglas vivas que existen. Se ha definido una propiedad de

configuración donde se puede establecer esta fecha y si no existiera se utilizará 1-ene-1970. Una vez obtenidos procesa los datos de ambos orígenes para calcular aquellas reglas que hay que expirar, crear o modificar.

Una vez procesada toda esta información se enviará a Anjana para su sincronización de metadatos. Si no se produce ningún error se pasa a la sincronización de resultados.

## Sincronización de resultados

A partir de los metadatos a sincronizar para obtener las reglas que hay que tener en cuenta, se inicia el proceso pidiendo en un primer momento los resultados a Aqiva a partir de la fecha obtenida en el primer paso. De los resultados a aplicar (excluyendo aquellas reglas que en la sincronización de metadatos han resultado a expirar) generamos 2 tipos de resultados:

- Resultado de datasources a nivel de dataformat. Agrupando por dataformat, datasource y qualityPoint todas las reglas tendrán el mismo valor así que se sincroniza en Anjana el correspondiente a la regla más nueva en Aqiva.
  - Resultado a nivel de reglas. Añadimos el listado de reglas a sincronizar.

Si hubiera algún problema podemos verificar los logs de Aqiva donde se puede identificar claramente el proceso aquí mencionado

```
2023-03-30 15:01:00.010 [PLUGIN-AQTVIA] DEBUG 20520 --- [ scheduling-1] c.a.t.p.a.connector.AqtviaConnector : Requesting for the last execution date persisted
2023-03-30 15:01:00.038 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.a.service.BatchAqtviaService : Init synchronization with Anjana. Requesting data from 2023-03-26T13:00:00
2023-03-30 15:01:00.172 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Init Metadata Synchronization process
2023-03-30 15:01:00.172 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 1. Getting date to query Aqtvia.
2023-03-30 15:01:00.172 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Date to query Aqtvia: 2023-03-26T13:00:00.830Z
2023-03-30 15:01:00.172 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 2. Getting all aqtvia live records from 2023-03-26T13:00:00.830Z
2023-03-30 15:01:00.249 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 3. Obtained a total of 1 live rules
2023-03-30 15:01:00.250 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 4. Getting all anjana data related with Aqtvia
2023-03-30 15:01:00.335 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 5. Obtained a total of 1 data from anjana
2023-03-30 15:01:00.335 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 6. Finding data to expire
2023-03-30 15:01:00.336 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 7. Finding rules to create or modify
2023-03-30 15:01:00.340 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 8. Finding metadata to expire
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : End Metadata Synchronization process
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] DEBUG 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Time elapsed for Aqtvia metadata synchronization: 1142ms
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Init Result Synchronization process
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 1. Getting date to query Aqtvia.
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Date to query Aqtvia: 2023-03-26T13:00:00.830Z
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] DEBUG 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 2. Initialize data from metadata synchronization process
2023-03-30 15:01:01.180 [PLUGIN-AQTVIA] DEBUG 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 3. Getting all aqtvia results from 2023-03-26T13:00:00.830Z
2023-03-30 15:01:01.307 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 4. Obtained a total of 1 records
2023-03-30 15:01:04.663 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 5. 1 rules to process after filtering wrong values in rules
2023-03-30 15:01:04.663 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 6. Filtering non live rules from Anjana
2023-03-30 15:01:04.664 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 7. 1 rules to process after filtering non live rules in Anjana
2023-03-30 15:01:04.664 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 8. 5 Finding live results over grouping by dataSources.
2023-03-30 15:01:04.664 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Grouping by dateFormat, dataSource and qualityPoint and get those with max timestamp
2023-03-30 15:01:04.664 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 9. 1 dataFormats to be applied after processing
2023-03-30 15:01:04.666 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 10. 6. Finding live results over grouping by rules.
2023-03-30 15:01:04.666 [PLUGIN-AQTVIA] DEBUG 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Grouping by dateFormat and Quality Rule and get those with max timestamp
2023-03-30 15:01:04.666 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : 11. 1 rules to be applied after processing
2023-03-30 15:01:04.666 [PLUGIN-AQTVIA] DEBUG 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : Time elapsed for Aqtvia results synchronization: 3486ms
2023-03-30 15:01:05.989 [PLUGIN-AQTVIA] INFO 20520 --- [ scheduling-1] c.a.t.p.aqtvia.service.AqtviaService : End Result Synchronization process
```

## Endpoints disponibles

- POST /api/aqtv/synchronize/rules. Se encarga de realizar el proceso completo de sincronización del metadatado de Aqtv. Por un lado sincroniza el metadatado actualizando y expirando según los datos de Anjana y Aqtv y para finalizar sincroniza los resultados según esos datos de actualización.
  - POST /synchronize/rules/metadata. Se encarga de sincronizar los resultados del metadatado en Anjana según el payload pasado de sincronización de metadatado.
  - GET /synchronize/metatada. Se encarga de sincronizar el metadatado en Anjana con los datos de Aqtv según el filtro pasado (fecha de inicio del proceso).

## Tot

Tot va a actuar como un mero proxy entre el plugin de Aqtiva y Kerno. El único trabajo que hace es que cuando se termina la petición de los resultados, si todo ha ido bien (no se ha producido ninguna excepción), persiste en base de datos la fecha del momento actual.

## Endpoints disponibles

- GET /internal/v4/anjana/aqtiva/lastExecution. Se encarga de obtener la última ejecución del plugin de Aqtiva.
- POST /internal/v4/anjana/aqtiva/synchronize. Se encarga de obtener el metadatado de Aqtiva almacenado en Anjana.
- PATH /internal/v4/anjana/aqtiva/synchronize. Se encarga de dirigir la petición a Kerno de la actualización del metadatado.
- /internal/v4/anjana/aqtiva/result. Se encarga de dirigir la petición a Kerno para actualizar los resultados de los objetos en Anjana relativos a Aqtiva. También se encarga de guardar la fecha de última actualización cuando no se ha producido ningún error.

## Kerno

Kerno va a procesar los datos que le envíe el plugin según cada proceso.

## Actualización de metadatos

En este caso tiene dos funciones, por un lado obtener el metadato existente en Anjana que esté relacionado con algún dataformat (esto es, aquellos datasets que tengan un atributo dataformat en su plantilla con valor), y por otro, hacer efectiva esa actualización deprecando objetos y creando reglas necesarias.

- Cuando tenga que expirar datasources obtendrá los datasets en los que está relacionado y eliminará ese resultado de la lista de resultados del dataset.
- Cuando tenga que expirar dataformats se obtendrán los datasets asociados y se actualizarán los atributos que contienen la relación entre datasets y reglas
  - AqtivaQRulesArray del dataset se elimina la regla a expirar
  - AqtivaAnjanaDatasets de la regla se deja a vacío
- Cuando tenga que expirar dataformats se obtendrán los datasets asociados y se borrarán el valor de ciertos atributos del dataset:
  - AqtivaIdDataFormat
  - AqtivaDataFormatName
  - AqtivaKPILastUpdateDate
  - AqtivaDataQualityDatasourceLevel
- Las reglas a expirar se ponen en estado DEPRECATED y se asigna al atributo finishDate (fecha de expiración) el momento actual de ejecución para que se expire cuando toque (siguiente batch de expiración). Si este atributo no estuviera asignado en la template se mostrará un log por consola indicando la incidencia pero se permitirá guardar la regla como deprecada.
- De las reglas a crear se comprueba si existe o no. Si existe se actualiza los datos con la información que venga del plugin y si no existe se crea una regla nueva asociándose al dataset correspondiente.

## Actualización de resultados

- Se procesan los resultados a nivel de datasource y a nivel de regla. Para los datasources se obtienen los datasets asociados a los dataformats a actualizar y se añade en el atributo de resultados los distintos valores pasados. Este proceso añade o modifica, no tiene que borrar ninguno.
- Se procesan los resultados a nivel de regla. Para cada regla se agrupan los resultados según el dataformat e identificador de la regla mostrando el listado de resultados de las ejecuciones para esa regla. Existe una particularidad con estos resultados y es que se utiliza un atributo que contendrá el valor “técnico” de los resultados y otro atributo que contendrá el valor visual del mismo. Este proceso añade o modifica, no tiene que borrar ningún resultado.

Si hubiera algún problema podemos verificar los logs de Kerno donde se puede identificar claramente el proceso aquí mencionado:

```

2023-03-30 15:50:00.265 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.loggableDispatcherServlet : POST "/internal/v4/metadata/aqtiva/synchronize", parameters={}
2023-03-30 15:50:00.268 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.d.service.AqtivaService : 1. Find if any dataformat is related with more than one dataset (APPROVED or DEPRE
2023-03-30 15:50:00.279 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.d.service.AqtivaService : 2. Exclude those dataformats that are related with different objects.
2023-03-30 15:50:00.279 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.d.service.AqtivaService : DataFormats excluded because they are related with different datasets:
2023-03-30 15:50:00.279 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.d.service.AqtivaService : 3. Find all AQTIVA_RULE in Anjana excluding dataformat repeated in datasets
2023-03-30 15:50:00.314 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.d.service.AqtivaService : 38 rules from Aqtiva in Anjana
2023-03-30 15:50:00.315 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.d.service.AqtivaService : Processing dataformat 605da85ab48d5a01a3ff8861
2023-03-30 15:50:00.324 [KERNO] DEBUG 18884 --- [nio-8081-exec-4] c.a.c.e.s.loggableDispatcherServlet : Completed 200 OK
2023-03-30 15:50:00.325 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.loggableDispatcherServlet : PATCH "/internal/v4/metadata/aqtiva/synchronize", parameters={}
2023-03-30 15:50:00.326 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Expiring Aqtiva data in Anjana
2023-03-30 15:50:00.326 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : 0 rules to be expired: []
2023-03-30 15:50:00.326 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Creating or modifying Aqtiva rules in Anjana
2023-03-30 15:50:00.326 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Getting all active attribute definitions for AQTIVA_RULE
2023-03-30 15:50:00.326 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Getting datasets related with dataFormat 605da85ab48d5a01a3ff8861
2023-03-30 15:50:00.365 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : 1 datasets found related with dataFormat 605da85ab48d5a01a3ff8861
2023-03-30 15:50:00.365 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Getting rules in Anjana related with the AqtivaQualityRuleName [test_integracion_a
2023-03-30 15:50:00.379 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : 1 rules found
2023-03-30 15:50:00.379 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Rule test_integracion_aqtiva-QP1_test_anjana2-DS1.test_anjana-sample-renfe-trips-5
2023-03-30 15:50:00.380 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.d.service.AqtivaService : Updating AqtivaAnjanaDatasets attribute from rule 165281 adding value [ANJA:OBJECT:EN
2023-03-30 15:50:00.407 [KERNO] DEBUG 18884 --- [nio-8081-exec-5] c.a.c.e.s.loggableDispatcherServlet : Completed 200 OK
2023-03-30 15:50:00.430 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.e.generic.service.EntityService : Init getObjectsToIndex
2023-03-30 15:50:01.012 [KERNO] INFO 18884 --- [nervaExecutor-4] org.hibernate.SQL_SLOW : SlowQuery: 432 milliseconds. SQL: 'HikariProxyPreparedStatement@1971357218 wrapping
2023-03-30 15:50:01.203 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL --> http://zeus/internal/v1/auth/users/organizational-unit
2023-03-30 15:50:01.203 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event EXPIRED SimpleKey [Europe/SPA/Finance,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.287 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event CREATED SimpleKey [Europe/SPA/Finance,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.350 [KERNO] DEBUG 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event EXPIRED SimpleKey [Europe/SPA/Accounting,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.355 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL --> http://zeus/internal/v1/auth/users/organizational-unit
2023-03-30 15:50:01.393 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event CREATED SimpleKey [Europe/SPA/Accounting,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.442 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event EXPIRED SimpleKey [Europe/SPA/Finance,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.444 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL --> http://zeus/internal/v1/auth/users/organizational-unit
2023-03-30 15:50:01.474 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event CREATED SimpleKey [Europe/SPA/Finance,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.481 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event EXPIRED SimpleKey [Europe/SPA/Market,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.481 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL --> http://zeus/internal/v1/auth/users/organizational-unit
2023-03-30 15:50:01.517 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event CREATED SimpleKey [Europe/SPA/Market,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.554 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.e.generic.service.EntityService : End getObjectsToIndex time elapsed: 1124ms
2023-03-30 15:50:01.555 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.e.generic.service.EntityService : Init getObjectsToIndex
2023-03-30 15:50:01.584 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event EXPIRED SimpleKey [Europe/SPA/Finance,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.588 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL --> http://zeus/internal/v1/auth/users/organizational-unit
2023-03-30 15:50:01.612 [KERNO] INFO 18884 --- [e [..default..]-1] c.a.core.nero.config.CacheEventLogger : custom Caching event CREATED SimpleKey [Europe/SPA/Finance,ORGANIZATIONAL_UNIT_OWNER]
2023-03-30 15:50:01.615 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.e.generic.service.EntityService : End getObjectsToIndex time elapsed: 60ms

```

```

2023-03-30 15:50:01.620 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.k.c.s.MinervaOperationsService : Bulk indexation
2023-03-30 15:50:01.620 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL -> http://minerva/internal/v1/indexer/KERNO/index/group
2023-03-30 15:50:02.185 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL -> http://minerva/internal/v1/indexer/KERNO/index/group
2023-03-30 15:50:02.493 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.k.batch.service.LineageService : Generating lineage for entities [1551, 165281]
2023-03-30 15:50:02.493 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.c.k.batch.service.LineageService : Init processing doDeleteAndSave
2023-03-30 15:50:02.585 [KERNO] INFO 18884 --- [nervaExecutor-4] c.a.c.k.batch.service.LineageService : End processing doDeleteAndSave
2023-03-30 15:50:02.586 [KERNO] DEBUG 18884 --- [nervaExecutor-4] c.a.core.audit.service.LoggerAspect : Indexing audit
2023-03-30 15:50:02.586 [KERNO] DEBUG 18884 --- [nervaExecutor-4] com.anjana.core.eureka.EurekaConnector : Call URL -> http://minerva/internal/v1/audit/index
2023-03-30 15:50:05.409 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.a.s.LoggableDispatcherServlet : PATCH "/internal/v4/metadata/aqtvia/result", parameters={}
2023-03-30 15:50:05.410 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : Processing results for datasources
2023-03-30 15:50:05.410 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : 1 dataformats with results to be updated ([605da85ab48d5a01a3ff8861])
2023-03-30 15:50:05.426 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : 1 datasets found to update its results
2023-03-30 15:50:05.426 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : 1 datasets found after filtering by state
2023-03-30 15:50:05.427 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : Aqtvia Result for DATASET 1551 updated. Old value: test_integracion_aqtvia.QP1_te...
demo_vueling_2-QP1-DS_ESTIMATION_446f0-sample-renfe-trips-500k.csv-null_rule_idAndcompanyAndoriginAnddestinationAn..._1:108.90
ndemo_vueling_2-QP1-DS_ESTIMATION_446f0-sample-renfe-trips-500k.csv-null_rule_idAndcompanyAndoriginAnddestinationAn..._2 : 500
test_integracion_aqtvia.QP1_test_anjana2.DS1_test_anjana:83.232666 New value test_integracion_aqtvia.QP1_test_anjana:74.849
demo_vueling_2-QP1-DS_ESTIMATION_446f0-sample-renfe-trips-500k.csv-null_rule_idAndcompanyAndoriginAnddestinationAn..._1:108.90
ndemo_vueling_2-QP1-DS_ESTIMATION_446f0-sample-renfe-trips-500k.csv-null_rule_idAndcompanyAndoriginAnddestinationAn..._2 : 500
test_integracion_aqtvia.QP1_test_anjana2.DS1_test_anjana:83.232666
2023-03-30 15:50:05.448 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : Processing results for rules
2023-03-30 15:50:05.505 [KERNO] INFO 18884 --- [nio-8081-exec-6] org.hibernate.SQL.SLOW : SlowQuery: 29 milliseconds. SQL: 'HikariProxyPreparedStatement@108944415 wrapping...
2023-03-30 15:50:05.547 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.k.e.s.d.service.AqtviaService : 0 rules found to update its results
2023-03-30 15:50:05.558 [KERNO] DEBUG 18884 --- [nio-8081-exec-6] c.a.c.a.s.LoggableDispatcherServlet : Completed 200 OK
2023-03-30 15:50:05.569 [KERNO] DEBUG 18884 --- [nervaExecutor-5] c.a.c.k.e.generic.service.EntityService : Init getObjectsToIndex
2023-03-30 15:50:06.054 [KERNO] INFO 18884 --- [nervaExecutor-5] org.hibernate.SQL.SLOW : SlowQuery: 369 milliseconds. SQL: 'HikariProxyPreparedStatement@2146658986 wrapping...
2023-03-30 15:50:06.379 [KERNO] DEBUG 18884 --- [nervaExecutor-5] c.a.c.k.e.generic.service.EntityService : End getObjectsToIndex time elapsed: 810ms
2023-03-30 15:50:06.384 [KERNO] DEBUG 18884 --- [nervaExecutor-5] c.a.c.k.c.s.MinervaOperationsService : Bulk indexation
2023-03-30 15:50:06.384 [KERNO] DEBUG 18884 --- [nervaExecutor-5] com.anjana.core.eureka.EurekaConnector : Call URL -> http://minerva/internal/v1/indexer/KERNO/index/group
2023-03-30 15:50:08.446 [KERNO] DEBUG 18884 --- [nervaExecutor-5] c.a.c.k.batch.service.LineageService : Generating lineage for entities [1551]
2023-03-30 15:50:08.475 [KERNO] INFO 18884 --- [nervaExecutor-5] c.a.c.k.batch.service.LineageService : Init processing doDeleteAndSave
2023-03-30 15:50:08.503 [KERNO] INFO 18884 --- [nervaExecutor-5] c.a.c.k.batch.service.LineageService : End processing doDeleteAndSave
2023-03-30 15:50:08.503 [KERNO] DEBUG 18884 --- [nervaExecutor-5] c.a.core.audit.service.LoggerAspect : Indexing audit
2023-03-30 15:50:08.504 [KERNO] DEBUG 18884 --- [nervaExecutor-5] com.anjana.core.eureka.EurekaConnector : Call URL -> http://minerva/internal/v1/audit/index

```

## Endpoints disponibles

- POST /internal/v4/metadata/aqtvia/synchronize. Se encarga de obtener el metadatado relativo a Aqtvia que existe en Anjana. Se puede pasar un filtro pensado para dar soporte a la paginación aunque en esta versión no se está usando. Se ha generado de base para futuros desarrollos si hiciera falta.
- PATCH /internal/v4/metadata/aqtvia/synchronize. Se encarga de realizar la sincronización del metadatado de Aqtvia en Anjana según los valores pasados creando las reglas necesarias, asociándolo a los datasets y deprecando los objetos necesarios.
- PATCH /internal/v4/metadata/aqtvia/result. Se encarga de actualizar los resultados (a nivel de datasource en los datasets y a nivel de distintas ejecuciones en reglas) del metadatado de Aqtvia existente en Anjana según los valores pasados.