E4.1 SERVICIOS CONTENIDO

ESPACIO DE DATOS LINGÜÍSTICO (SER 15/23 OTT)

Resumen

Este documento describe los principales servicios de contenido del conector INESData que se usan en el despliegue del espacio de datos lingüístico EDL. Los servicios de contenido están relacionados con la localización, almacenamiento y transferencia de ficheros asociados a los activos en el espacio de datos. Se incluye servicios que dan soporte a ficheros almacenados como recursos HTTP o en un sistema de almacenamiento en la nube que provee INESData. Además, se describe los servicios que permiten transferir ficheros usando el protocolo HTTP o a un sistema de almacenamiento en la nube soportado por INESData. En esta primera versión del entregable se incluyen los servicios que hasta la fecha están disponibles en el repositorio GitHub de INESData.

Andrés García-Silva

26/06/2024

Expert.ai Expert.ai Language Technology Research Lab

Historia

revisión	Fecha	Descripción	Autor
0.1	20/06/2024	Primera versión del documento	Andrés García-Silva
0.2	31/07/2024	Revisión del documento	José Manuel Gómez

Contenido

1 Introducción

El espacio de datos lingüístico EDL se apoya en los componentes INESData para el despliegue de espacios de datos. Los componentes de INESDATA ofrece servicios de contenido que dan soporte a la localización, almacenamiento y transferencia de conjuntos de datos. La interfaz de usuario del espacio de datos lingüístico ofrece estos servicios de contenido al usuario final cuando se crean activos o se transfieren activos en el marco de negociaciones finalizadas. La interfaz se apoya en el conector de INESDATA y en servicios de almacenamiento que provee INESDATA como el sistema de almacenamiento Min.io.

En este documento se describen los principales servicios que ofrecen los componentes de INESDATA para la localización, almacenamiento y transferencia de ficheros. Los datos de un activo en INESData pueden estar disponibles en i) una dirección http, ii) almacenados previamente en un sistema de almacenamiento minio en la nube de INESData, o iii) almacenarse al momento de la creación del activo en el sistema de almacenamiento minio.

Estos ficheros pueden a su vez ser transferidos al usuario consumidor usando i) un servicio HTTP o ii) a un sistema de almacenamiento minio en la nube de INESData que esté asociada al usuario consumidor.

En la versión actual del conector de INESData algunos tipos de transferencia están en desarrollo como por ejemplo la transferencia de ficheros usando HTTP y la transferencia de ficheros almacenados en minio al mismo tiempo que la creación del activo. Se espera que en la siguiente versión del conector soporte todos los tipos de transferencias que serán reportados en la siguiente versión de este entregable.

2 Datos almacenados en la nube de INESDATA (MinIO-S3)

El espacio de datos lingüístico ofrece la posibilidad a los usuarios de almacenar y compartir los activos en la nube de INESDATA. Cada conector tiene una instancia de Min.io que se ejecuta en la nube de INESDATA.

MinIO es una plataforma de almacenamiento de objetos que se puede implementar en infraestructuras de nube privadas, públicas o híbridas. Está diseñado para ofrecer almacenamiento compatible con Amazon S3, lo que significa que las aplicaciones que funcionan con S3 pueden integrarse con MinIO sin cambios significativos. Algunas características importantes de MinIO son:

- Compatibilidad con S3: MinIO implementa el API de Amazon S3, lo que permite una fácil integración con aplicaciones y herramientas que ya utilizan S3.
- Rendimiento Elevado: MinIO está optimizado para un alto rendimiento, con capacidades para manejar grandes volúmenes de datos y realizar operaciones rápidas de lectura/escritura.
- Escalabilidad Horizontal: MinIO permite la escalabilidad horizontal, lo que significa que se pueden añadir más nodos al clúster para aumentar la capacidad de almacenamiento y mejorar el rendimiento.
- Almacenamiento Distribuido: MinIO proporciona almacenamiento distribuido, permitiendo la replicación para asegurar la disponibilidad y durabilidad de los datos.
- Seguridad: Ofrece características avanzadas de seguridad, incluyendo encriptación en tránsito y en reposo, así como control de acceso basado en políticas (IAM) detallado.
- Simplicidad y Facilidad de Uso: MinIO es conocido por su simplicidad y facilidad de uso, con una interfaz de usuario intuitiva y una configuración sencilla.

Los datos de configuración del minio del conector se encuentran en el fichero Docker-compose.yml que se usa para desplegar el espacio de datos. En el siguiente ejemplo se definen dos instancias minio para el conector 1 y 2 respectivamente. Las consolas de administración se pueden acceder en el puerto 9001 y 9011 y las API en los puertos 9010 y 900 respectivamente para cada conector cuando se accede desde fuera de los contenedores. Si el acceso es interno entre contenedores los puertos que dan acceso a la API es el 9000 en ambos casos y se debe usar la ruta interna de los contenedores: minio-c1 y minio-c2.

```
minio-c1:
    image: minio/minio
    ports:
      - '9000:9000'
      - '9001:9001'
    volumes:
      - mdata-c1:/data
    environment:
      - MINIO_ROOT_USER=inesdata
      - MINIO_ROOT_PASSWORD=inesdata
      - MINIO_DEFAULT_BUCKETS=bucket-connector-1
    command: server --console-address ":9001" /data
    healthcheck:
      test: timeout 5s bash -c ':> /dev/tcp/127.0.0.1/9000' || exit 1
      interval: 30s
      timeout: 20s
      retries: 3

minio-c2:
    image: minio/minio
   ports:
      - '9010:9000'
      - '9011:9011'
    volumes:
      - mdata-c2:/data
    environment:
      - MINIO_ROOT_USER=inesdata
      - MINIO_ROOT_PASSWORD=inesdata
      - MINIO_DEFAULT_BUCKETS=bucket-connector-2
    command: server --console-address ":9011" /data
    healthcheck:
      test: timeout 5s bash -c ':> /dev/tcp/127.0.0.1/9000' || exit 1
      interval: 30s
      timeout: 20s
      retries: 3
```

3 Datos HTTP

La forma más sencilla de compartir datos en el espacio de datos lingüísticos es tener disponibles los datos en una dirección URL desde donde se puedan descargar.

Al crear un activo en el espacio de datos usando el servicio web (POST /management/v3/assets) el usuario proveedor tiene que especificar en la dirección de los datos (dataAddress) que el activo está disponible como datos de tipo HTTP (HttpData) y la URL donde se encuentran. A continuación, un ejemplo del documento json que se envía con esta información al crear el activo:

```
{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@id": "c1-http-asset1",
  "properties": {
    "name": "c1-http-asset1",
    "contenttype": "application/json"
  },
  "dataAddress": {
    "type": "HttpData",
    "name": "c1-http-asset1",
    "baseUrl": "https://jsonplaceholder.typicode.com/posts",
    "proxyPath": "true"
  }
}
```

Además, el proveedor del activo debe crear al menos una política de acceso y de contratación, y una definición de contrato donde se apliquen las políticas al activo que se desea compartir vía HTTP.

Por otra parte, el usuario consumidor debe iniciar la negociación del contrato que aparece en su catálogo con los activos disponibles en el espacio de datos. Está negociación se inicia con una llamada POST al servicio http://localhost:29193/management/v2/contractnegotiations y enviando un fichero json con la información del activo que se quiere negociar, y la política de contratación:

```
POST /management/v2/contractnegotiations HTTP/1.1
Host: localhost:29193
Content-Type: application/json
Authorization: Bearer Access Token
Content-Length: 881

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
    "odrl": "http://www.w3.org/ns/odrl.jsonld"
  },
  "@type": "ContractRequest",
  "counterPartyAddress": "http://connector-c1:19194/protocol",
  "protocol": "dataspace-protocol-http",
  "policy": {
    "@context": "http://www.w3.org/ns/odrl.jsonld",
    "@id": "PolicyId-from-federated-catalog",
    "@type": "Offer",
    "odrl:permission": {
        "odrl:action": {
            "odrl:type": "USE"
        },
```

```
        "odrl:constraint": {
            "odrl:leftOperand": "POLICY_EVALUATION_TIME",
            "odrl:operator": {
                "@id": "odrl:lt"
            },
            "odrl:rightOperand": "2024-12-31T23:59:59+01:00"
        }
    },
    "assigner": "connector-c1",
    "target": "{{asset-id}}"
  }
}
```

Sí la política de contratación se valida automáticamente entonces la contratación debería finalizar de forma exitosa. Esto se puede comprobar usando el servicio de consulta de estado de negociación: http://localhost:29193/management/v2/contractnegotiations/{id-contract}. La negociación debe estar en estado finalizada.

3.1 Transferencia de ficheros usando HTTP

Cuando la negociación ha finalizado con éxito se puede realizar la transferencia del activo. Para esto hay que hacer una llamada post al servicio en 29193/managemente/v2/transferprocesses enviando un documento json donde se indique el activo que se quiere transferir y el id del acuerdo del contrato que se generó en la negociación.

```
POST /management/v2/transferprocesses HTTP/1.1
Host: localhost:29193
Content-Type: application/json
Authorization: Bearer Access token
{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@type": "TransferRequestDto",
  "connectorId": "connector-c2",
  "counterPartyAddress": "http://connector-c1:19194/protocol",
  "contractId": "{{contract-agreement-id}}",
  "assetId": "{{asset-id}}",
  "protocol": "dataspace-protocol-http",
  "transferType": "HttpData-PULL"
}
```

El estado de la transferencia se puede ver haciendo una llamada get al siguiente servicio pasando el id de la transferencia:

```
GET /management/v2/transferprocesses/{{transferId}} HTTP/1.1
Host: localhost:29193
Authorization: Bearer Access Token
```

Nota Importante: En la versión actual del conector de INESData solo es posible descargar el fichero incluido en el activo al utilizar el servicio web y el token de acceso que se puede visualizar en los logs del http-proxy (Ver ejemplo del log en la **Error! Reference source not found.**). Sin embargo, dado que la información solo aparece en el log no es posible para el conector consumidor recuperarla. Por tanto, no es posible automatizar aún la descarga del fichero para el conector consumidor. Se espera que en una próxima versión del conector INESData se provea un servicio web para poder recuperar la información que indique donde se puede descargar el fichero.

Figura 1. Log del http-proxy donde aparece la url y el token de acceso para descargar el fichero asociado a un activo transferido en el espacio de datos.

3.2 Transferencia de ficheros a Minio S3

Para iniciar la transferencia de ficheros al sistema de almacenamiento minioS3 del conector consumidor se debe especificar en el campo dataDestination los datos de acceso cómo la llave de acceso y la llave secreta, e información de la localización incluyendo el servicio de minio y el nombre del bucket donde se almacenará.

```
POST /management/v2/transferprocesses HTTP/1.1
Host: localhost:29193
Content-Type: application/json
Authorization: Bearer

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@type": "TransferRequestDto",
  "connectorId": "connector-c2",
  "counterPartyAddress": "http://connector-c1:19194/protocol",
  "contractId": "{{contract-agreement-id}}",
  "assetId": "{http-asset-id}",
  "protocol": "dataspace-protocol-http",
  "transferType": "AmazonS3",
  "dataDestination": {
    "properties": {
      "type": "AmazonS3",
      "region": "us-east-1",
      "bucketName": "bucket-connector-2",
```

```
      "accessKeyId": "wE..",
      "secretAccessKey": "Jl.."
    },
    "endpointOverride": "http://minio-c2:9000",
    "type": "AmazonS3"
  },
  "managedResources": true
}
```

Cuando el estado de la transferencia pase a Finalizada quiere decir que podremos ver y descargar el fichero del minio del conector consumidor (por ejemplo, http://localhost:9011/).

4 Datos en Minio

El espacio de datos lingüístico soporta la creación de activos con ficheros almacenados en minio y la transferencia de estos usando HTTP o a un sistema de almacenamiento minio. Además, el activo puede referenciar un fichero ya almacenado en minio o puede subirlo al momento de la creación del activo a minio.

4.1 Creación de activo que referencia un fichero en minio

Para crear un activo que referencia un fichero almacenado previamente en minio se requiere una llamada POST al servicio /management/v3/assets y se pasa un documento json con la información del activo. En la información del activo se especifica la localización del activo y debe incluirse la información de ubicación minio (incluyendo el nombre del bucket). A continuación, un ejemplo de llamada al servicio desde el conector del usuario proveedor:

```
POST /management/v3/assets HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access token

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@id": "{{asset-id}}",
  "properties": {
    "name": "{{asset-name}}",
    "contenttype": "text/plain"
  },
  "dataAddress": {
    "type": "AmazonS3",
    "region": "us-east-1",
    "keyName": "{{file-name}}",
    "bucketName": "{{bucket-name}}"
  }
}
```

Como cualquier otro activo el usuario proveedor debe crear y asociar una política de acceso y de contrato en un contrato que incluya el activo.

4.2 Creación de activos y carga de fichero simultánea a minio S3

Para crear activos en el espacio de datos lingüísticos y almacenar el fichero asociado en la nube de INESData se hace una llamada post al servicio /management/s3assets. Se pasa un documento json con la información del activo incluyendo los metadatos y la información de la ubicación (data Address). La información de la ubicación debe incluir el tipo INESDataStore y la carpeta donde se quiere almacenar el activo en minio. Además, se debe pasar el fichero que se subirá a minio. Tanto el json como el fichero se den pasar como datos de un formulario de múltiples partes (multipart/form-data).

A continuación, una llama de ejemplo para crear un activo y subir el fichero relacionado a la nube de INESData:

```
POST /management/s3assets HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access Token
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary7MA4YWxkTrZu0gW

------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="json"
Content-Type: application/json

{
    "@context": {
        "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
    },
    "@id": "{{Asset-id}}",
    "properties": {
        "name": "{{Asset-name}}",
        "contenttype": "text/plain"
    },
    "dataAddress": {
        "type": "INESDataStore",
            "folder": ""
    }
}
------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file";
filename="/C:/Users/agarcia/Downloads/fibra.pdf"
Content-Type: application/pdf

(data)
------WebKitFormBoundary7MA4YWxkTrZu0gW—
```

Si la creación del activo ha sido correcta el servicio devuelve el código 200 y un json confirmando la creación:

```
{
    "@type": "IdResponse",
    "@id": "c1-internalS3-asset-1",
    "createdAt": 1718968110719,
    "@context": {
        "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
        "edc": "https://w3id.org/edc/v0.0.1/ns/",
        "odrl": "http://www.w3.org/ns/odrl/2/"
    }
}
```

Para verificar que el fichero se ha subido a la nube de INESData se puede visitar la web de administración de minio asociada al conector proveedor. Además, el usuario proveedor debe crear un contrato donde se asignen políticas de acceso y de contratación al activo.

En la versión actual del conector de INESDATA no está soportada la transferencia de activos con ficheros subidos simultáneamente al minio. Sin embargo, se espera que en una próxima versión del conector de INESDATA está transferencia sea soportada.

4.3 Transferencia de ficheros usando HTTP

La transferencia de ficheros almacenados en minio usando http es igual a la descrita en la sección 3.1. Además, también aplica la restricción que en la versión actual del conector de INESData solo es posible obtener la url de descarga a partir de los logs del http-proxy y por tanto no se puede automatizar la descarga por parte del conector consumidor.

4.4 Transferencia de ficheros a Minio S3

La transferencia de ficheros almacenados en minio a un sistema de almacenamiento minio del conector consumidor es igual al proceso que se describió en la sección 3.2.

5 Conclusiones

Se ha presentado los servicios de contenidos que usa el espacio de datos lingüísticos y que ofrece el conector de INESData. Estos servicios permiten localizar, almacenar y transferir ficheros asociados a los activos creados en los espacios de datos. Los ficheros pueden almacenarse como un recurso HTTP o en un sistema de almacenamiento minio que ofrece la nube de INESData. Estos ficheros pueden transferirse vía HTTP o directamente a un sistema de destino minio asociado al conector del usuario consumidor del activo. Algunos servicios aún están en desarrollo, como la descarga de ficheros cuando la transferencia se usa HTTP o cuando el fichero se ha almacenado en minio al crearse el activo. Se espera que en la próxima versión del conector de INESData se soporte estas transferencias que serán reportadas en la próxima versión del entregable.