



Inventory interceptor

Control de versiones	2
Modelo de integración	2
Interfaces de entrada y salida	3
Inclusión y activación	4
Inclusión en classpath (kerno)	4
Aplicación de configuración necesaria	5
Artefactos	5
Librería	5
Interceptor	7

Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Anjana Producto	Anjana Producto	Creación del documento

Modelo de integración

Se habilita en kerno un punto en el cual se pueden transformar valores de atributos incluidos en la plantilla.

La clase transformadora recibirá todo el objeto completo y deberá devolver ese mismo objeto pero con los valores de atributos transformados como se quiera y necesite.

No se puede añadir un atributo nuevo ni eliminar uno de los que ya contenga el objeto, en tal caso la ejecución fallará. Sólo se puede modificar el valor de un atributo, ningún otro parámetro, si no la ejecución fallará.

Interfaces de entrada y salida

El DTO de entrada y salida es el mismo, `com.anjana.core.customers.model.EntityFormDTO`, y está disponible en la librería pública `customers-lib`.

Dentro de este DTO se encuentra la propiedad `entityAttributes`, esta propiedad contiene una lista de `AttributeDefinitionDTO`. En este segundo DTO donde se puede hacer transformaciones es en la propiedad `value`. El resto de atributos no se pueden modificar pero pueden ser de utilidad para la transformación que se necesite hacer.

Para más información se puede consultar la siguiente imagen con la información más detallada del DTO y todas sus propiedades:

```

EntityFormDTO v {
  idObject      integer($int32)
                example: 1
                The object id
  idParent      integer($int32)
                example: 1
                The id parent
  referenceId    integer($int32)
                example: 654
                The reference id
  organizationalUnit string
                example: HHQ/LEGAL
                The organizational unit
  entityAttributes*
                v [
                The entity attributes
                AttributeDefinitionDTO v {
                description:
                    The state
                id      integer($int32)
                    example: 7980
                    The id
                name    string
                    example: colour
                    The name
                shortDescription string
                    The description
                isCustom boolean
                    example: true
                    The isCustom flag
                type    string
                    The type
                Enum:
                value    > Array [ 29 ]
                label    > {...}
                    string
                    example: module
                    The label
                placeholder string
                    example: data here
                    The place holder
                }
                }
}

```

Inclusión y activación

Inclusión en classpath (kerno)

Cambio en orden/entrypoint kerno

```
java -Xmx4G -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5005
-javaagent:/opt/common/xjar-agent-hibernate.jar -jar "/opt/keruo/keruo.jar"
--spring.cloud.config.uri=http://edusaserver:8888 --spring.profiles.active=default
--spring.cloud.config.failFast=true --spring.main.allow-bean-definition-overriding=true
```

Aplicación de configuración necesaria

Para habilitar la transformación de atributos se debe añadir la propiedad: `anjana.transformation.enabled = true`
Además se tiene que marcar cual es la paquetería y clase elegidas para hacer transformaciones con la propiedad (ejemplo):

```
anjana.transformation.className =
com.anjana.inventory.example.DescriptionCalculation
```

El jar de hermes-extra se ha introducido en el classLoader por código, por lo que es necesario incluirlo en la propiedad:

```
anjana.transformation.jarLocation =
"/opt/inventory-interceptor-example/inventory-interceptor-example.jar"
```

Si kerno se despliega en Windows, la ruta debe tener el siguiente formato:

```
"C://work/inventory-interceptor-example/target/inventory-interceptor-examp
le.jar"
```

Artefactos

Librería

Para nuevos artefactos se usará la librería provista.

```
<dependency>
<groupId>com.anjana.core</groupId>
<artifactId>customers-library</artifactId>
<version>4.4.0</version>
</dependency>
```

 **Browse** /  devmvn

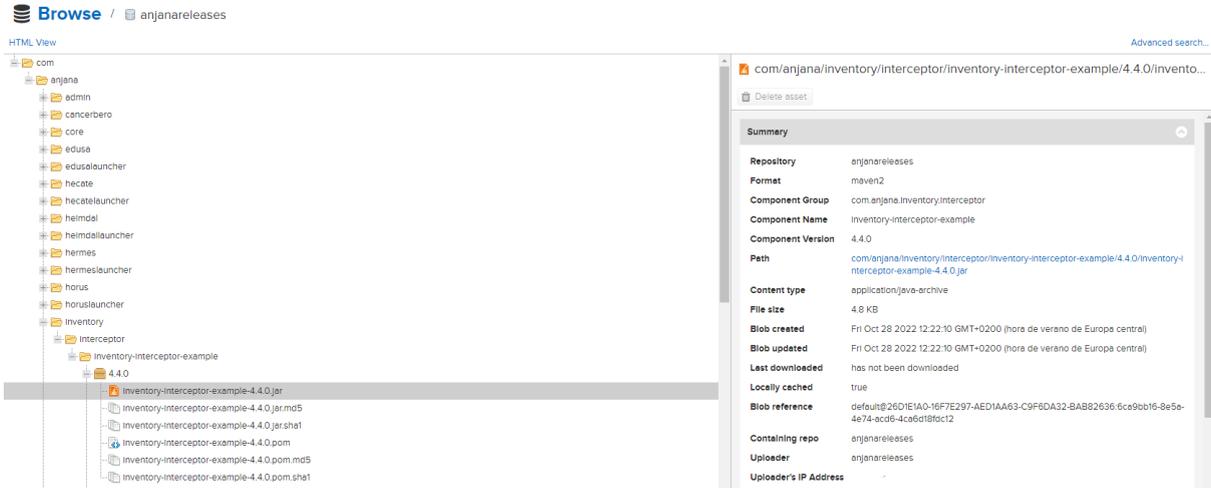
HTML View



- com
 - anjana
 - admin
 - anjana-starter
 - bgapi
 - bgapi_deps
 - cancerbero
 - core
 - core-library
 - customers-library
 - 4.2.1
 - 4.2.4
 - 4.2.5
 - 4.2.6
 - 4.2.7
 - 4.2.8
 - 4.2.11
 - 4.3.0
 - 4.3.1
 - 4.3.2-rc
 - 4.3.2
 - 4.4.0
 - customers-library-4.4.0.jar
 - customers-library-4.4.0.jar.md5
 - customers-library-4.4.0.jar.sha1
 - customers-library-4.4.0.pom
 - customers-library-4.4.0.pom.md5
 - customers-library-4.4.0.pom.sha1

Interceptor

Se provee de un interceptor base para su uso o modificación, se proveerán las fuentes de dicho artefacto de ejemplo a petición del cliente.



The screenshot shows a Maven repository browser interface. On the left, a tree view displays the directory structure under 'com', with 'Inventory-Interceptor-Example 4.4.0' selected. On the right, a summary panel provides details for the selected artifact.

Summary	
Repository	anjana/releases
Format	maven2
Component Group	com.anjana.inventory.interceptor
Component Name	inventory-interceptor-example
Component Version	4.4.0
Path	com/anjana/inventory/interceptor/inventory-interceptor-example/4.4.0/inventory-interceptor-example-4.4.0.jar
Content type	application/java-archive
File size	4.8 KB
Blob created	Fri Oct 28 2022 12:22:10 GMT+0200 (hora de verano de Europa central)
Blob updated	Fri Oct 28 2022 12:22:10 GMT+0200 (hora de verano de Europa central)
Last downloaded	has not been downloaded
Locally cached	true
Blob reference	default@26D1E1A0-16F7E297-AED1AA63-C9F9DA32-BAB82636-6ca9bb16-9e5e-4e74-acd6-4ca5d18fd12
Containing repo	anjana/releases
Uploader	anjana/releases
Uploader's IP Address	-