

Componente Java para la integración con DMSAS

Dirección General de Tecnologías de la Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud
Consejería de Sanidad, Presidencia
y Emergencias

Contenido

- Dirección General de Tecnologías de la Información y Comunicaciones
 - Áreas de Gobierno Tecnológico de SSII y del Dato
- 1. Introducción
- 2. Historial de Cambios
- 3. Integración de la librería
 - Configuración
- 4. Funcionalidades
 - Autenticación
 - Búsqueda

Resumen



- **Versión:** v01r17
- **Fecha publicación:**
31 may 2019
- **Entrada en vigor desde:**
31 may 2019

blocked URL

Cumplimiento normativo



Las normas expuestas son de obligado cumplimiento. La STIC podrá estudiar los casos excepcionales los cuales serán gestionados a través de los responsables del proyecto correspondiente y autorizados por el Área de Gobernanza de la STIC. Asimismo cualquier aspecto no recogido en estas normas deberá regirse en primera instancia por las guías técnicas correspondientes al esquema nacional de seguridad y esquema nacional de interoperabilidad según correspondencia y en su defecto a los marcos normativos y de desarrollo software establecidos por la Junta de Andalucía, debiendo ser puesto de manifiesto ante la STIC.

La STIC se reserva el derecho a la modificación de la norma sin previo aviso, tras lo cual, notificará del cambio a los actores implicados para su adopción inmediata según la planificación de cada proyecto.

En el caso de que algún actor considere conveniente y/o necesario el incumplimiento de alguna de las normas y /o recomendaciones, deberá aportar previamente la correspondiente justificación fehaciente documentada de la solución alternativa propuesta, así como toda aquella documentación que le sea requerida por la STIC para proceder a su validación técnica.

Contacto Arquitectura: l-arquitectura.stic@juntadeandalucia.es

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabezas de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	v01r17	Fecha publicación	31 may 2019	Fecha entrada en vigor	31 may 2019
Alcance					
Versión inicial sobre el componente que permite la integración con DMSAS fácilmente haciendo mínimo JDK 1.7.					

1. Introducción

Para la integración con dmsas recomendamos el uso de la siguiente librería **dmsasApiClient**. Esta librería permitirá ser desarrollada y mantenida por la STIC, permitiendo al usuario comunicarse con el servidor LDAP de DMSAS de forma fácil y sin necesidad de conocer los aspectos técnicos de dicha comunicación.

Actualmente la librería tiene las siguientes funcionalidades:

- **Autenticación**
 - Permite realizar la autenticación con dmsas.
- **Busqueda**
 - Permite realizar la búsqueda de usuarios en dmsas.

De cara a priorizar futuras funcionalidades, si algún proveedor tiene la necesidad de alguna funcionalidad adicional se deberá de comunicar a l-arquitectura.stic.sspa@juntadeandalucia.es

2. Historial de Cambios

- **1.1.3.2:**
 - Modificación para hacer serializables los beans expuestos por DmsasApiClient-cdi.
 - Modificación para hacer serializable DmsasUser.
- **1.1.3.1**
 - Optimizada la búsqueda evitando que realice búsquedas en diferentes nodos
 - Limitado el numero de resultados por defecto a 20
- **1.1.2.1**
 - Añadida posibilidad de establecer el dominio y compatibilidad con la libreria de sasconfiguration
- **1.1.1.1**
 - Inyeccion con CDI de los servicios.
- **1.1.0.1**
 - Búsqueda en DMSAS de usuarios
- **1.0.0.1**
 - Versión inicial
 - Autenticación con DMSAS

3. Integración de la librería

Para incluir la librería en un proyecto maven es necesario realizar los siguientes pasos:

1. Agregar el repositorio de la Junta de Andalucía. Es posible realizarlo de dos formas diferentes.

-
- Especificando el repositorio en la configuración global.

```
        <project>
    ...
    <repositories>
      <repository>
        <id>Repo_SAS</id>
        <name>Repo_SAS</name>
        <url>http://calidad.sas.junta-andalucia.es/artifactory/sas-internal</url>
      </repository>
    </repositories>
    ...
  </project>
```

- Otra opción es especificar un perfil en concreto en \${user.home}/.m2/settings.xml

```

        <settings>
        ...
        <mirrors>
            <mirror>
                <id>Repo_SAS</id>
                <name>Repo_SAS</name>
                <url>http://snap.java.lib.repository.dev.alm.sas.junta-andalucia.
es</url>
                <mirrorOf>*</mirrorOf>
            </mirror>
        </mirrors>
        <profiles>
        <profile>
            <activation>
                <activeByDefault>true</activeByDefault>
            </activation>
            <repositories>
            <repository>
                <id>nexus-public</id>
                <name>Nexus Public Repository</name>
                <url>http://snap.java.lib.repository.alm.pre.sas.junta-andalucia.
es</url>
                <releases>
                    <enabled>true</enabled>
                </releases>
                <snapshots>
                    <enabled>true</enabled>
                    <updatePolicy>always</updatePolicy>
                </snapshots>
            </repository>
            </repositories>
        </profile>
        </profiles>
        ...
    </settings>

```

Hay que tener en cuenta, que se deberá de tener activo el perfil.

2. Especificar la dependencia en el POM del proyecto que requiere la funcionalidad. Se especificara "-cdi" si se quiere hacer uso de CDI e inyecciones.

```

        ...
    <dependencies>
        <!--DMSAS-->
        <dependency>
            <groupId>es.ja.csalud.sas.componentescomunes.dmsasapiclient</groupId>
            <artifactId>dmsasApiClient[-cdi]</artifactId>
            <version>x.x.x.x</version>
        </dependency>
        ...
    </dependencies>

```

Configuración

Puede ser usada mediante CDI o de forma tradicional creando los objetos requeridos por cada constructor. **Con CDI es indispensable especificar como se producirán las inyecciones** del objeto de configuración.

Actualmente existe un objeto de configuración:

- **DmsasApiConfig**, gestiona la configuración general de la librería. Ejemplo:

```
...
@Produces
public DmsasApiConfig getDmsasApiConfig() {
    final DmsasApiConfig dmsasApiConfig = new DmsasApiConfig();
    dmsasApiConfig.setUrl("ldap://x.x.x.x:xxx/DC=dmsas,DC=sda,DC=sas,DC=junta-andalucia,
DC=es");
    dmsasApiConfig.setSearchDN("OU=VirgenRocio,OU=AtencionEspecializada");
    dmsasApiConfig.setLdapDomain("dmsas-test")
    return dmsasApiConfig;
}
...
```

- Haciendo uso de la librería de configuración sasconfiguration:
Ejemplo:

```

@Inject
SasConfigService sasConfigService;
...
@Produces
public DmsasApiConfig getDmsasApiConfig() {
    return new DmsasApiConfig(sasConfigService);
}
...
```

Los campos que se deberán de establecer en la configuración serán:

- es.ja.csalud.sas.componentescomunes.dmsasapiclient.config.url
- es.ja.csalud.sas.componentescomunes.dmsasapiclient.config.searchDN
- es.ja.csalud.sas.componentescomunes.dmsasapiclient.config.domain

Los parametros para los diferentes entornos son:

parametro	valor PRO	valor PRE
Url	ldap://sclldapd.dmsas.sda.sas.junta-andalucia.es:389/DC=dmsas,DC=sda,DC=sas,DC=junta-andalucia,DC=es	ldap://ldap.dmsas-test.sda-test.sas.junta-andalucia.es/DC=dmsas-test,DC=sda-test,DC=sas,DC=junta-andalucia,DC=es
searchDn	vacio (se puede rellenar si se quiere limitar a un grupo especifico)	vacio (se puede rellenar si se quiere limitar a un grupo especifico)
ldapDomain	dmsas	dmsas-test

Este objeto serán usados por defecto en todas las inyecciones de la librería.

Si no se usa CDI para inyectar los servicios se podrá hacer uso de la factoria DmsasFactory que facilita la creacion de los servicios.

4. Funcionalidades

Autenticación

Para poder realizar la autenticación y autorización con Dmsas se deberá de hacer uso del servicio `DmsasAuthentication`. Mediante el uso del método `.loginAuthentication()` que devuelve un objeto `DmsasUser`, se podrá realizar la autenticación con Dmsas. Si la autenticación no se ha podido realizar con éxito se lanzará una excepción de tipo `DmsasException`.

Ejemplo:

```
...
@Inject
DmsasAuthentication dmsasAuthentication;

public DmsasUser login(String username, String password) {
    return dmsasAuthentication.loginAuthentication(username, password);
}
...
```

Búsqueda

Para poder realizar la Búsqueda en Dmsas se deberá de hacer uso del servicio `DmsasSearch`. Mediante el uso del método `.userSearch()` o `.usersSearch()` que devuelve un objeto `DmsasUser` o una lista de estos, será necesario pasarle como parámetros el usuario que realizará la operación de búsqueda, y otro objeto de usuario con los parámetros de búsqueda. Si la búsqueda no se ha podido realizar con éxito se lanzará una excepción de tipo `DmsasException`.

Ejemplo:

```
... //con cdi
@Inject
DmsasSearch dmsasSearch;
... //sin cdi
DmsasFactory dmsasFactory = new DmsasFactory(apiConfig);
DmsasSearch dmsasSearch=dmsasFactory.getDmsasSearch();
... //búsqueda
DmsasUser result=dmsasSearch.userSearch(petitioner, searchCriteria);
...
```