REGIONE AUTONOMA DELLA SARDEGNA

SISaR

ENGINEERING

Servizio di Transizione

Titolo: XMPI/JBF/JAUF
Codice: AREAS_AP_01_JBF
Profilo: Analista Programmatore

REGIONE AUTONOMA DELLA SARDEGNA

SISaR

ENGINEERING

ARGOMENTI DI QUESTE SLIDE

- Oggetti comuni e OSGI
- Integrazioni LDAP e SSO (oggetti comuni)
- Webservices ABF (eventi)
- Rilascio delle modifiche al DB

Introduzione a JBF-XMPI

Oggetti comuni e OSGi

LDAP e SSO

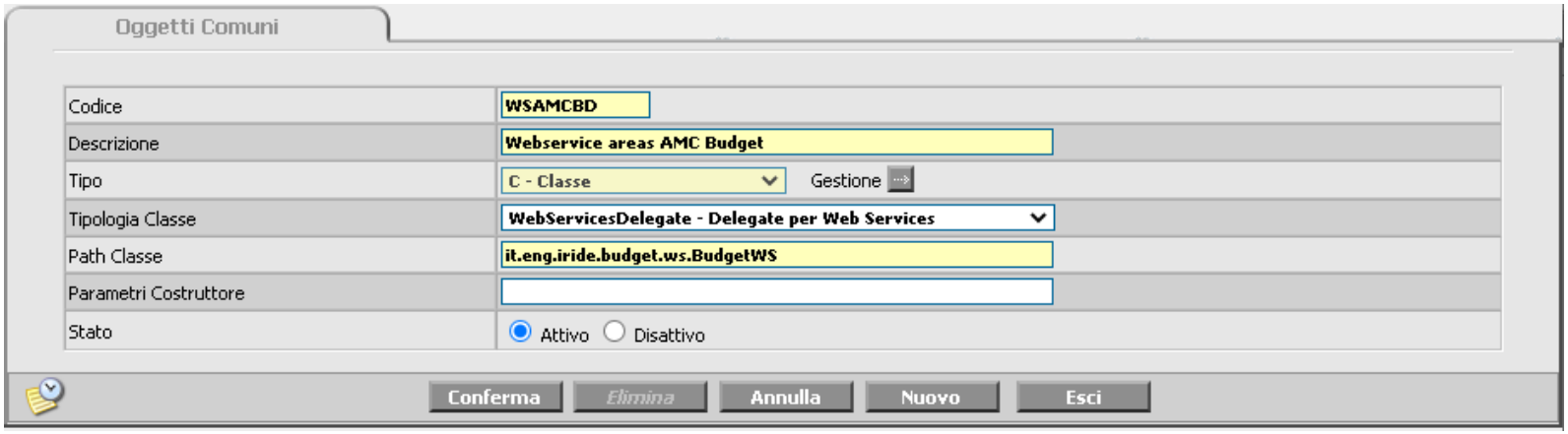
WebService ABF

Modifiche al DB

Plugin / Oggetti comuni

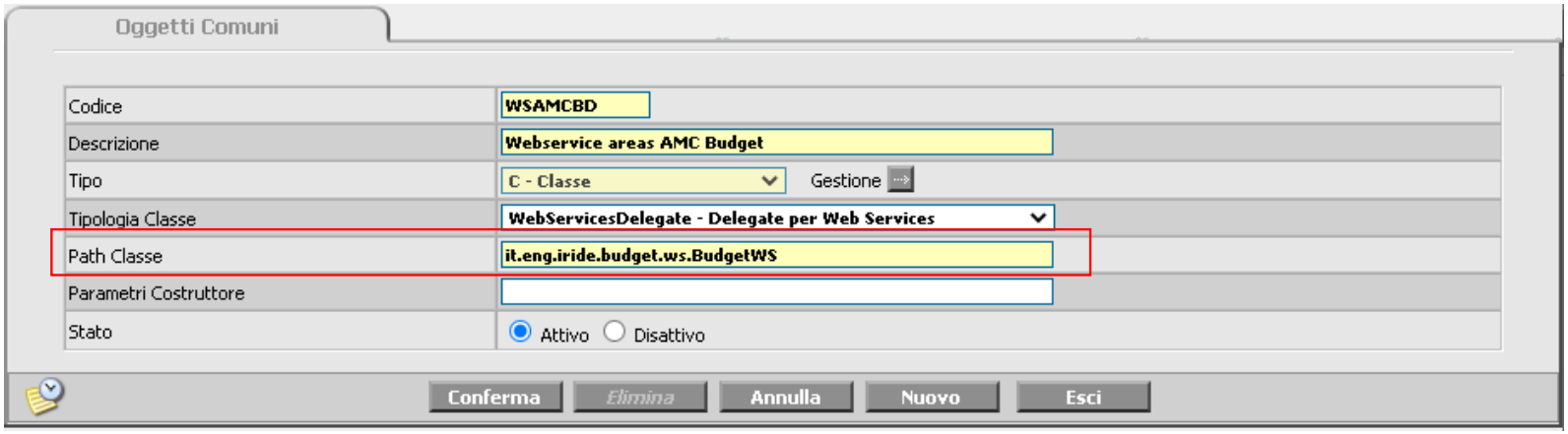
Alcune componenti software sono rilasciate in modalità **plugin** allo scopo di estendere le funzionalità base di AREAS. Il principale vantaggio del plugin è la facilità di installazione (non richiede l'aggiornamento dell'intero sistema).

I plugin di AREAS sono gestiti tramite il menu **Configurazione>>Avanzate>>Oggetti comuni**. Ciascun plugin e tipizzato in modo da istruire il sistema su come il plugin debba essere utilizzato



Plugin / Oggetti comuni

Tecnicamente un oggetto comune non è altro che un jar al cui interno è presente una classe "**entry-point**" che implementa una particolare interfaccia. Questa classe viene poi dichiarata nel campo **Path classe** in fase di definizione dell'oggetto comune.



Plugin / Oggetti comuni

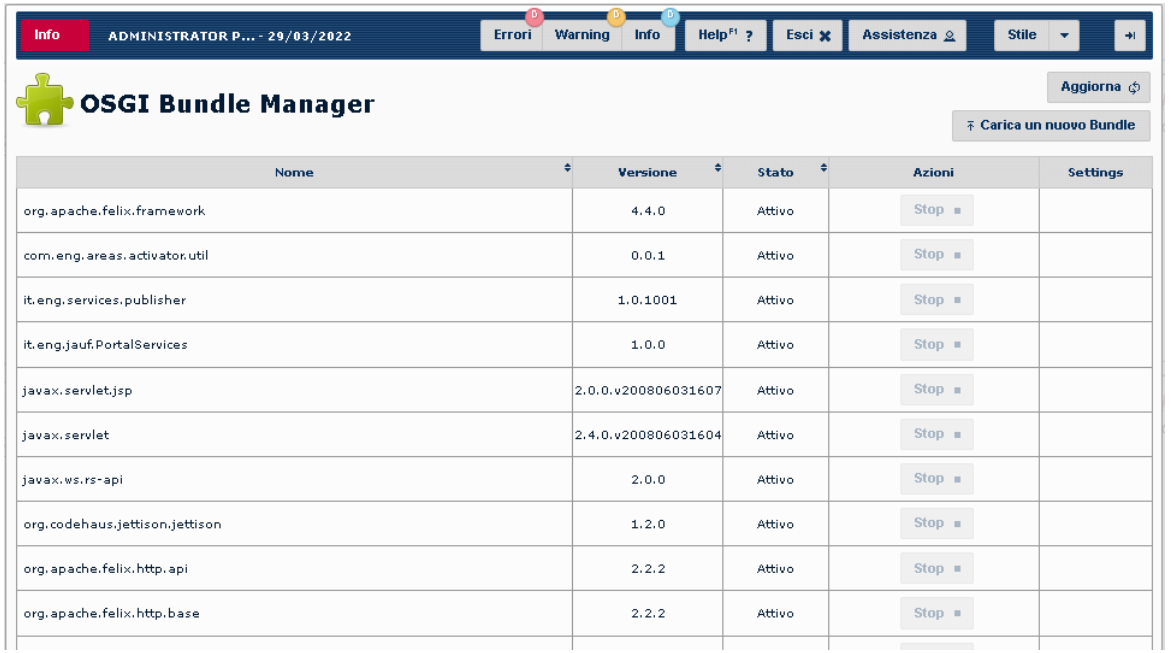
È possibile fare il deploy di diverse tipologie di plugin. Ciascuna tipologia richiede l'implementazione di una diversa interfaccia. Tra le tipologie a disposizione abbiamo:

- **OpenSSO**, per l'integrazione con sistemi SSO terzi
- **LDAP**, per l'integrazione con server LDAP
- **Ticket provider SSO**, per gestire il rilascio di token di autenticazione (chiamate di contesto)
- **Ricerca remota**, per integrare l'anagrafica pazienti con MPI terzi
- **Delegate WebService**, per l'implementazione di un servizio web service
- **Agent plugin**, per l'implementazione di un agent

Plugin / OSGI Manager

OSGi è un framework definito dalla OSGi Alliance. I plugin OSGi sono anche denominati **bundle** ed è possibile installarli tramite il menu **Configurazione>>Avanzate>>OSGI Manager**

Rispetto agli oggetti comuni si tratta di uno standard estremamente diffuso e non richiede il riavvio dell'applicazione (salvo necessità specifiche dei singoli bundle).



Plugin / OSGI Manager

A differenza degli oggetti comuni, inoltre, non è necessario l'implementazione di particolari interfacce o la dichiarazione preventiva della tipologia di plugin.

Ogni bundle espone una classe **Activator** che si occupa di gestire il proprio processo di start/stop. Il plugin potrà implementare in questa classe quanto necessario per interagire con il resto del sistema tramite opportune chiamate alla classe **PSGExt.osgi.OSGIFramework**.

```
public class Activator implements BundleActivator {

  public void start(BundleContext bundleContext) throws Exception {
    OSGIFramework.registerDynamicMapping("abfSign.SignTest",SignTestLayout.class, null);
  }
  public void stop(BundleContext bundleContext) throws Exception {
    OSGIFramework.unregisterDynamicMapping("abfSign.SignTest");
  }
```

Plugin / OSGI Manager

La classe **PSGExt.osgi.OSGIFramework** espone vari metodi per la registrazione di:

- **Controller**, OSGIFramework.registerDynamicMapping
- **Bean**, OSGIFramework.registerBean
- **Query**, OSGIFramework.registerQuery
- **Agent**, OSGIFramework.registerAgent

Scopo di questa classe è anche la dichiarazione della **boot delegation**: una lista di package che risiedono nei progetti standard (quindi non in plugin) ma che sono comunque “visibili” dai bundle (es: log4j, axis, dom4j e package dei progetti)

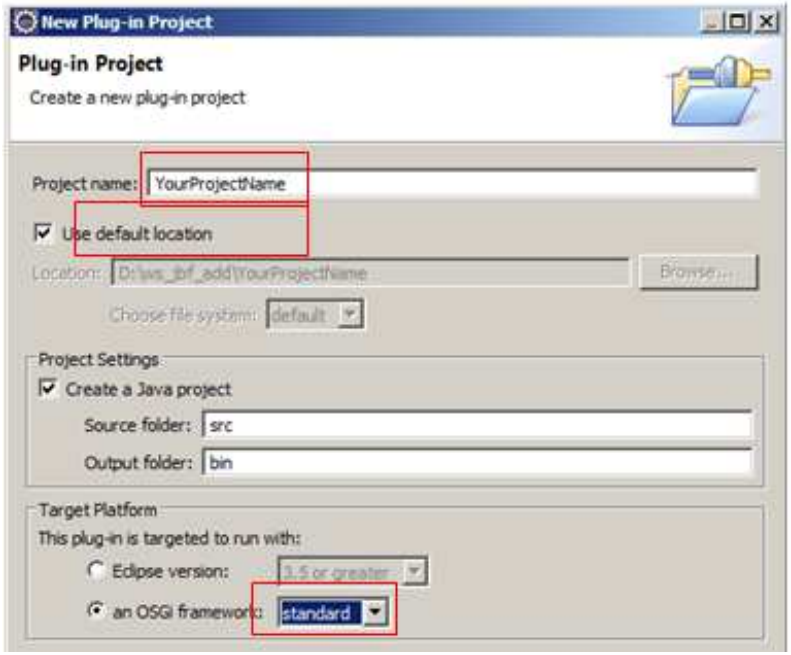
OSGI / Esercitazione

Proviamo a configurare insieme un nuovo plugin OSGi (solo struttura e scheletro)

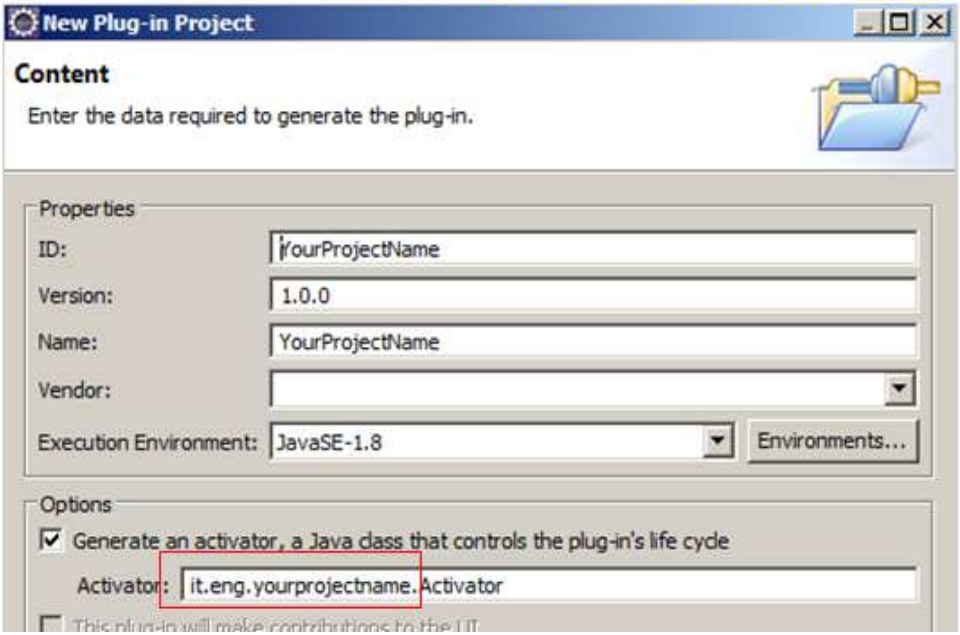
OSGI / Creazione di un plugin step by step

1 Create a new project and select the type "**Plug-in Project**"

2 Give a name and select "**standard**" in the "**an OSGi Framework**" field



3 In the **Activator** field, set the package **it.eng.name_of_your_project_in_lowercase.Activator**



OSGI / Creazione di un plugin step by step

4 Open the **MANIFEST.MF** file

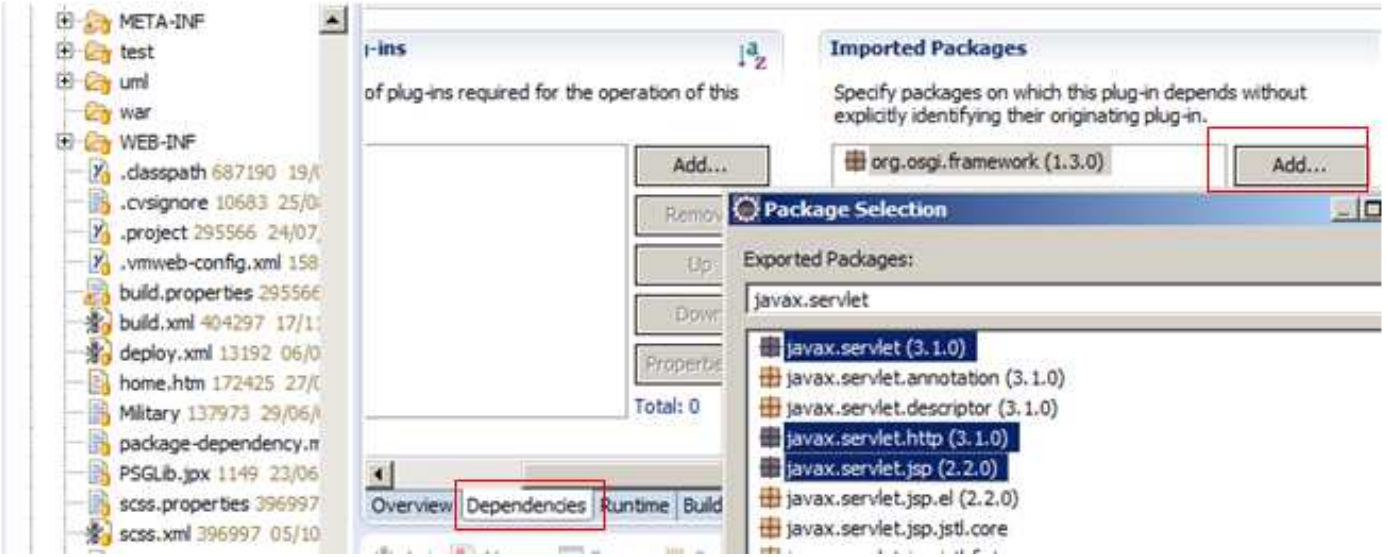
4.1 Select the "**Dependencies**" tab

4.2 Click "**Add**" button

4.3 Search the package **javax.servlet**

4.4 Select the packages **javax.servlet**, **javax.servlet.http**, **javax.servlet.jsp**

4.5 Click "**Ok**" button

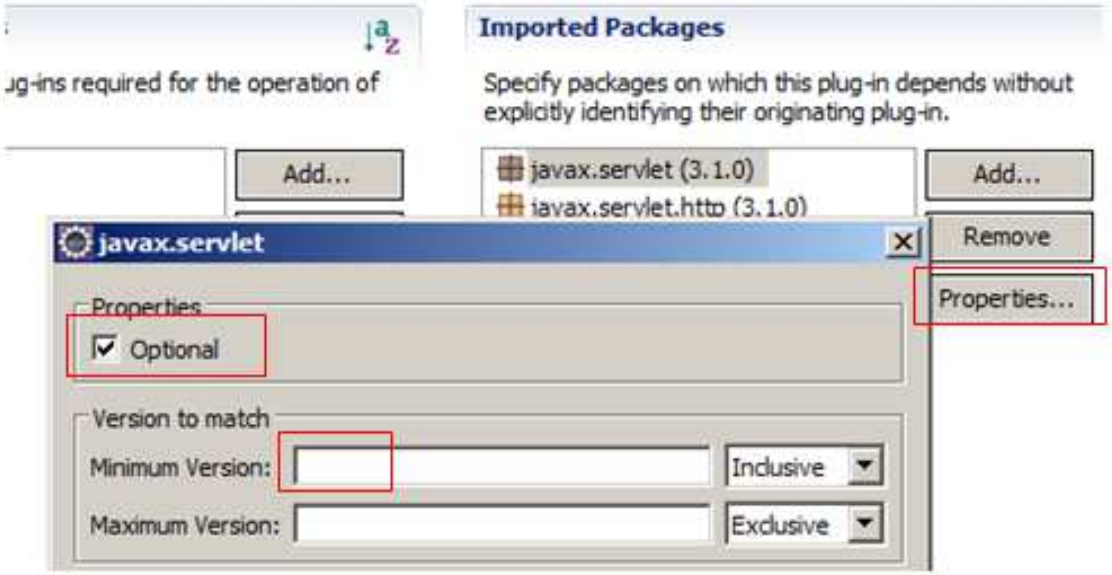


OSGI / Creazione di un plugin step by step

5 Select the packages you just added (one by one) and click the button "**Properties**"

5.1 Select the "**Optional**" check

5.2 Remove the value in the **Minimum Version** field



6 Click the "**export button**" in the right top corner and generate the jar file

You can now upload your jar via OSGIManager panel

Introduzione a JBF-XMPI

Oggetti comuni e OSGi

LDAP e SSO

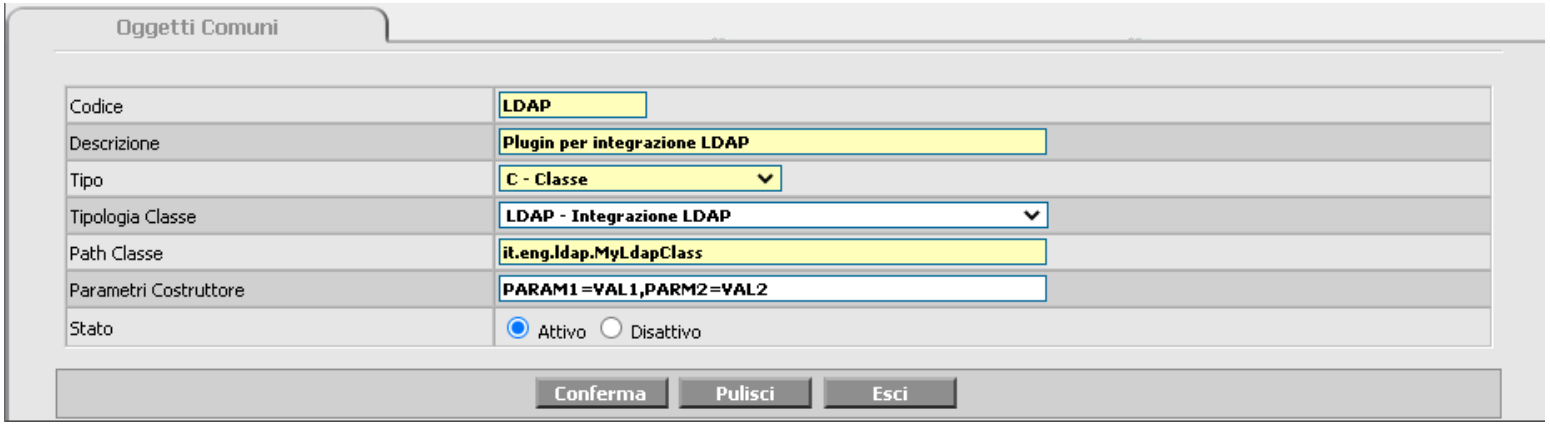
WebService ABF

Modifiche al DB

Plugin / LDAP e SSO

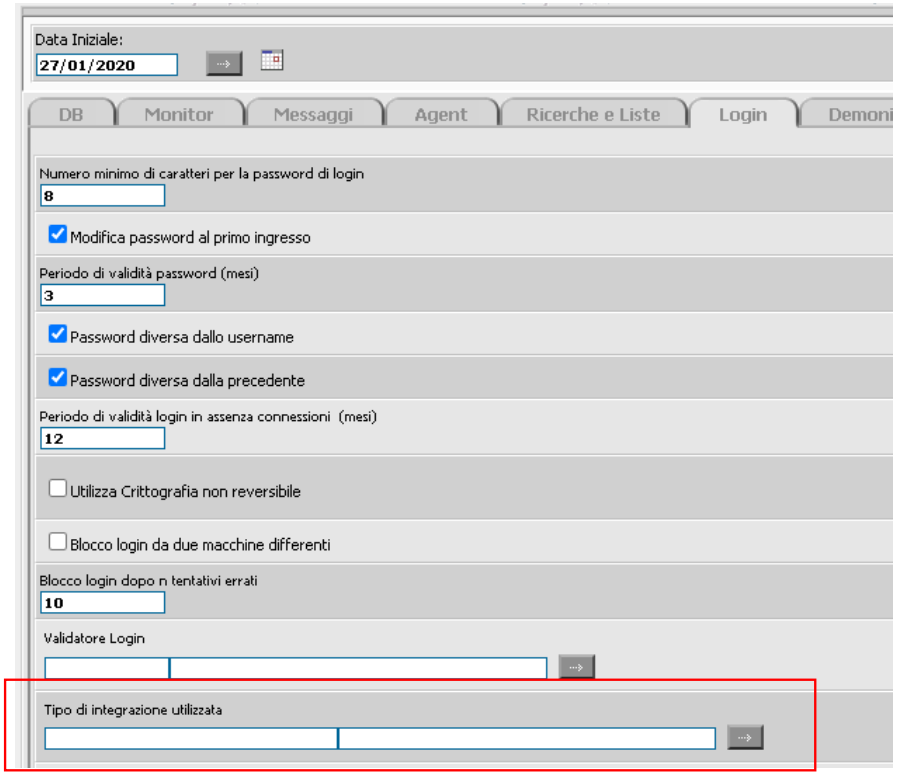
AREAS è in grado di delegare a terzi il processo di autenticazione utente integrandosi, ad esempio, con server **LDAP** o sistemi di **Single Sign On**. L'implementazione delle logiche necessarie all'integrazione avviene tramite il rilascio di un **oggetto comune**.

In questo caso il jar dovrà contenere una classe che implementa l'interfaccia **PSGExt.ldap.ILdapManager** per integrazioni via LDAP o **PSGExt.profile.w3.ExternalLogin** per SSO



Plugin / LDAP e SSO

Una volta installato l'oggetto comune è necessario "attivarlo" tramite un apposito parametro d'installazione sotto Sistema>>Login



Introduzione a JBF-XMPI

Oggetti comuni e OSGi

LDAP e SSO

WebService ABF

Modifiche al DB

WebService ABF / Introduzione

AREAS è un sistema aperto e permette l'accesso a tutti i suoi bean (che incapsulano le logiche di business) tramite opportuni **Web Service**.

Il servizio è di tipo **SOAP-RPC** (implementato tramite AXIS) in cui si chiama l'unico metodo call() che prende come parametro una stringa XML e restituisce una stringa XML. Il servizio effettua il parsing della stringa in input ed esegue il comando specificato.

Tutti i servizi, quindi, sono esposti attraverso un unico endpoint centrale:

http://miosito/contesto/**jbfServices/JBFService**

WebService ABF / Introduzione

Di seguito un esempio di chiamata al Web Service per il servizio standard di Open Session (login):

```
<?xml version="1.0" ?>
<!--
  Test con richiesta completa
-->
<Input>
  <OpenSession>
    <user>UTENTE</user>
    <password>PASSWORD</password>
    <azienda>AZ</azienda>
    <ufficio>UFFICIO</ufficio>
    <dataLavoro>20040101</dataLavoro>
  </OpenSession>
</Input>
```

WebService ABF / Introduzione

La chiamata precedente, inclusa all'interno di un envelope SOAP, diventa quindi così:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <call soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <xml xsi:type="xsd:string">&lt;?xml version=&quot;1.0&quot; ?&gt;&#xd;
&lt;!--&#xd;
  Test con richiesta completa&#xd;
--&gt;&#xd;
&lt;Input&gt;&#xd;
  &lt;OpenSession&gt;&#xd;
    &lt;user&gt;JUNIT&lt;/user&gt;&#xd;
    &lt;password&gt;JUNIT&lt;/password&gt;&#xd;
    &lt;azienda&gt;JU&lt;/azienda&gt;&#xd;
    &lt;ufficio&gt;JUNITUFF&lt;/ufficio&gt;&#xd;
    &lt;dataLavoro&gt;20040101&lt;/dataLavoro&gt;&#xd;
  &lt;/OpenSession&gt;&#xd;
&lt;/Input&gt;&#xd;
</xml>
  </call>
 </soapenv:Body>
</soapenv:Envelope>
```

WebService ABF / Flusso e servizi standard

All'interno di AREAS sono previsti una serie di servizi standard per l'accesso in modalità CRUD di tutti i suoi Bean e Query. In particolare:

• servizio di **OpenSession**, si occupa di aprire una sessione applicativa. In input riceve i dati di autenticazione e in output restituisce un SID (token) da utilizzare nelle successive chiamate

• servizio di **Find**, si occupa di eseguire una classe Query. In input è possibile indicare dei filtri e l'output è il risultato di ricerca della query

• servizio di **Get**, si occupa di caricare un singolo Bean. In input è possibile indicare l'identificativo del record e l'output è la lista completa delle proprietà del bean

• servizio di **Set**, si occupa di inserire/aggiornare un singolo Bean. In input è possibile indicare la lista delle proprietà da valorizzare sul bean e l'output è la lista completa delle proprietà e l'identificativo assegnato al bean

• servizio di **Delete**, si occupa di eliminare un singolo Bean. In input è possibile indicare l'identificativo del record e l'output è un messaggio di conferma o di errore

• servizio di **CloseSession**, si occupa di chiudere una sessione applicativa

NB: È possibile ottenere uno stub di tutti i servizi tramite il menu Configurazione>>Avanzate>>Help Webservice

WebService ABF / Configurazione

Per poter utilizzare un servizio, l'utente che apre la sessione (OpenSession) dovrà essere opportunamente abilitato. Le abilitazioni si basano sul concetto di **entità applicativa**, che rappresenta il sistema che vuole integrarsi con AREAS. Gli step completi da seguire sono:

1. Creazione di un'entità applicativa (Configurazione>>Sistema>>Demoni>>Entità applicativa)
2. Definizione del messaggio sul demone di tipo SI_WEBSERVICES (Configurazione>>Sistema>>Demoni>>Demone, tab "Messaggi")
3. Associazione dell'entità e abilitazione del messaggio appena creato (tab "Entità")

WebService ABF / Esercitazione

Proviamo ad utilizzare i servizi ABF tramite **SoapUI.** Vedremo sia la parte di configurazione e sia la parte di esecuzione

WebService ABF / Generazione di un evento

È possibile configurare AREAS in modo da generare dei messaggi in uscita al verificarsi di determinati **eventi**. Con eventi intendiamo l'inserimento, la modifica o la cancellazione di un oggetto su AREAS (Bean).

I vari eventi verranno accodati all'interno della tabella SI_DEMONEMSGQUEUE e conterranno il riferimento ad un XML con i dati completi del bean oggetto dell'evento.

La configurazione degli eventi che deve generare l'applicativo avviene nel punto di menu **Configurazione>>Sistema>>Demoni>>Demone**, tab "Messaggi". Gli eventi vanno poi abilitati alle entità applicative (tab Entità, Msg di output)

WebService ABF / Esercitazione

Vediamo come configurare un evento e il relativo messaggio di uscita generato sulla SI_DEMONEMSGQUEUE

Introduzione a JBF-XMPI

Oggetti comuni e OSGi

LDAP e SSO

WebService ABF

Modifiche al DB

Modifiche al DB / Introduzione

Tutto ciò che sviluppiamo deve essere rilasciato anche in termini di struttura e di oggetti sulla base dati. Esistono in AREAS due modalità per esportare le modifiche al DB:

- Tramite XML di versione
- Tramite la libreria opensource **Liquibase** (https://liquibase.org/)

Il primo metodo è utile **solo** in caso si preveda il rilascio di nuove versioni Major o minor. L'XML può essere creato tramite procedure guidate presenti all'url http://localhost:8080/areas/**PSGExt.MonitorSviluppatori.do**

Per tutte le altre casistiche è possibile creare una semplice modifica liquibase (**changelog**) seguendo i formalismi previsti dalla libreria.

Modifiche al DB / Liquibase in 4 passi

1. Individuare il proprio changelog master (HotDBModify.registerChangeLogMaster)
2. Aggiungere il proprio changeset (occhio al context)
3. Scrivere la propria modifica al DB secondo i formalismi disponibili nella documentazione ufficiale di liquibase (https://docs.liquibase.com/concepts/changelogs/changelog-formats.html)
4. Testare la propria modifica tramite il monitor liquibase (http://localhost:8080/areas/**hdm.ControlPanel.do**)

Modifiche al DB / Esercitazione

Proviamo a creare una nostra modifica liquibase per la creazione di una tabella e di un lookup.