

Web Services AMUE

Nous proposons ici quelques codes utilisés pour interagir avec Apogée et Siham au travers de leurs WebServices.

Au vu des particularités d'usage technique de ces WebServices, il a semblé intéressant d'initier cette page.

- [Web Service Apogée](#)
 - [Introduction](#)
 - [Contexte](#)
 - [Java : code initial avec la librairie client Apogée \(et donc code déprécié\)](#)
 - [Java : nouveau code](#)
 - [Generation de code depuis le wsdl](#)
 - [Code Java](#)
 - [Remarques](#)
- [Web Service Siham.](#)
 - [Contraintes Web Service Siham \(mars 2020\).](#)
 - [Accès concurrents](#)
 - [Session Timeout HRAccess](#)
 - [Problème de récupération en masse de dossiers agents](#)
 - [Exemple en python : détection des dossiers agents en erreurs](#)
 - [Exemple en Java : récupération des dossiers agents](#)
 - [Generation de code depuis le wsdl](#)
 - [Code Java](#)
 - [Descripteur Spring](#)
 - [Usage](#)
 - [Contrainte supplémentaire Web Service Siham \(février 2022 - SIHPRD.2.09.52\).](#)

Web Service Apogée

Intoduction

Le Web Service SOAP Apogée de l'AMUE a longtemps été livré avec un client jar.

Ce client ne sera plus livré à partir d'octobre 2020.

Le WSDL présenté par le WebService SOAP doit normalement permettre d'utiliser le WebService directement et de manière efficace, sans usage d'une librairie cliente spécifique.

Il est cependant à noter que le client JAVA du WS Apogée livré jusque là par l'AMUE permettait en fait de faciliter le maniement des structures de données du Web Service exposé.

Une conception plus légère du web service n'aurait pas nécessité de client spécifique, et permettrait un maniement aisé des méthodes exposées.

Contexte

On partage donc ici un retour d'expérience de la migration d'un code utilisant le client jar web service Apogée (webservicesclient-500la.jar) vers un code Java toujours n'utilisant plus cette librairie mais utilisant un code généré directement depuis un WSDL.

On se place dans le contexte d'une application Java qui permet de **modifier l'email d'un étudiant** en utilisant le web service Apogée.

Java : code initial avec la librairie client Apogée (et donc code déprécié)

```
...
public xxxx() {
    EtudiantMetierServiceInterface apogeeProxyEtu = new EtudiantMetierServiceInterfaceServiceLocator().
getEtudiantMetier(url);
    CoordonneesDT02 coordonnes = apogeeProxyEtu.recupererAdressesEtudiant_v2(codeEtu, annee, "N");
    CoordonneesMajDT0 coordonnesMaj = new CoordonneesMajDT0(coordonnees);
    coordonnesMaj.setEmail(newEmailPerso);
    apogeeProxyEtu.mettreAJourAdressesEtudiant(coordonneesMaj, codeEtu);
}
...
```

Java : nouveau code

Generation de code depuis le wsdl

Partie du pom.xml pour générer le client depuis le WSDL :

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>jaxws-maven-plugin</artifactId>
  <executions>
    <execution>
      <id>apogee-wsdl-etudiant-metier</id>
      <goals>
        <goal>wsimport</goal>
      </goals>
      <configuration>
        <wsdlUrls>
          <wsdlUrl>https://ws.univ-ville.fr/apo-ws/services/EtudiantMetier?wsdl</wsdlUrl>
        </wsdlUrls>
        <packageName>fr.univville.wsclient.apogee.etudiant</packageName>
      </configuration>
    </execution>
  </executions>
  <configuration>
    <verbose>true</verbose>
    <sourceDestDir>${basedir}/src/main/java</sourceDestDir>
    <args>
      <arg>-B-XautoNameResolution</arg>
    </args>
  </configuration>
</plugin>
```

Code Java

```
....

EtudiantMetierServiceInterface etudiantMetierService;
....

private EtudiantMetierServiceInterface getEtudiantMetierService() throws Exception {
    if(etudiantMetierService==null) {
        URL etudiantMetierWsUrl = new URL(wsdlUrl);
        EtudiantMetierServiceInterfaceService etudiantMetierServiceInterfaceService = new
EtudiantMetierServiceInterfaceService(etudiantMetierWsUrl);
        etudiantMetierService = etudiantMetierServiceInterfaceService.getEtudiantMetier();
    }
    return etudiantMetierService;
}

public xxxx() {
    CoordonneesDT02 coordonnes = getEtudiantMetierService().recupererAdressesEtudiantV2(codeEtu, annee,
"N");
    CoordonneesMajDTO coordonnesMaj = map2CoordonneesMajDTO(coordonnees);
    coordonnesMaj.setEmail(newEmailPerso);
    getEtudiantMetierService().mettreAJourAdressesEtudiant(coordonneesMaj, codeEtu);
}

private CoordonneesMajDTO map2CoordonneesMajDTO(CoordonneesDT02 coordonnees) {
    CoordonneesMajDTO coordonnesMaj = new CoordonneesMajDTO();
    coordonnesMaj.setAnnee(coordonnees.getAnnee());
    coordonnesMaj.setEmail(coordonnees.getEmail());
    coordonnesMaj.setNumTelPortable(coordonnees.getNumTelPortable());
    if(coordonnees.getTypeHebergement() != null) {
        coordonnesMaj.setTypeHebergement(coordonnees.getTypeHebergement().getCodTypeHebergement());
    }
    if(coordonnees.getAdresseAnnuelle()!=null) {
        AdresseMajDTO adresseAnnuelle = new AdresseMajDTO();
```

```

        if(coordonnes.getAdresseAnnuelle().getPays()!=null) {
            adresseAnnuelle.setCodPays(coordonnes.getAdresseAnnuelle().getPays().getCodPay());
        }
        if(coordonnes.getAdresseAnnuelle().getCommune()!=null) {
            CommuneMajDTO commune = new CommuneMajDTO();
            commune.setCodeInsee(coordonnes.getAdresseAnnuelle().getCommune().getCodeInsee());
            commune.setCodePostal(coordonnes.getAdresseAnnuelle().getCommune().getCodePostal());
            adresseAnnuelle.setCommune(commune);
        }
        adresseAnnuelle.setLibAd1(coordonnes.getAdresseAnnuelle().getLibAd1());
        adresseAnnuelle.setLibAd2(coordonnes.getAdresseAnnuelle().getLibAd2());
        adresseAnnuelle.setLibAd3(coordonnes.getAdresseAnnuelle().getLibAd3());
        adresseAnnuelle.setLibAde(coordonnes.getAdresseAnnuelle().getLibAde());
        adresseAnnuelle.setNumTel(coordonnes.getAdresseAnnuelle().getNumTel());
        coordonnesMaj.setAdresseAnnuelle(adresseAnnuelle);
    }
    if(coordonnes.getAdresseFixe() != null) {
        AdresseMajDTO adresseFixe = new AdresseMajDTO();
        if(coordonnes.getAdresseFixe().getPays()!=null) {
            adresseFixe.setCodPays(coordonnes.getAdresseFixe().getPays().getCodPay());
        }
        if(coordonnes.getAdresseFixe().getCommune()!=null) {
            CommuneMajDTO commune = new CommuneMajDTO();
            commune.setCodeInsee(coordonnes.getAdresseFixe().getCommune().getCodeInsee());
            commune.setCodePostal(coordonnes.getAdresseFixe().getCommune().getCodePostal());
            adresseFixe.setCommune(commune);
        }
        adresseFixe.setLibAd1(coordonnes.getAdresseFixe().getLibAd1());
        adresseFixe.setLibAd2(coordonnes.getAdresseFixe().getLibAd2());
        adresseFixe.setLibAd3(coordonnes.getAdresseFixe().getLibAd3());
        adresseFixe.setLibAde(coordonnes.getAdresseFixe().getLibAde());
        adresseFixe.setNumTel(coordonnes.getAdresseFixe().getNumTel());
        coordonnesMaj.setAdresseFixe(adresseFixe);
    }
    return coordonnesMaj;
}

```

Remarques

Au vu de la méthode `""CoordonneesMajDTO map2CoordonneesMajDTO(CoordonneesDTO2 coordonnes)""` qu'il a fallu créer, on comprend **ce qu'apportait le client web service et ce qu'il fait défaut maintenant ici**.

Les structures de données n'étant pas les mêmes lorsqu'on récupère la donnée et lorsqu'on souhaite la mettre à jour, on est obligé ici de 'transvaser' (transférer) la donnée d'une structure à l'autre (correspond à l'anti-pattern "Redundant data models").

Le code obtenu ci-dessus est lourd à mettre en place et à maintenir, notamment en cas d'évolutions des web services SOAP Apogée, structures des données comprises.

Notez que naturellement l'oubli de la reprise d'un champ dans la méthode `map2CoordonneesMajDTO` entraîne l'écrasement de ce champ dans Apogée puisque cela correspond à le 'setter' à null (le réinitialiser).

Web Service Siham.

Siham propose des Web Services SOAP pour récupérer les informations des personnels.

L'objectif est de partager le code permettant d'utiliser les webservices `ListeAgentsWebService` et `DossierAgentDateWebService` (et `DossierParametrageWebService` pour la partie Java)

Contraintes Web Service Siham (mars 2020).

Nous avons identifié plusieurs contraintes à l'usage des web services siham ; nous espérons que Siham évolue rapidement pour corriger ces problèmes.

Accès concurrents

Le Web Service Siham `DossierAgentDateWebService.RecupDonneesAgents` ne supporte pas les appels en accès concurrent : si 2 clients appellent ce web service siham en même temps, le web service plante.

En octobre 2019, l'AMUE nous signalait qu'une correction devait être apportée dans le prochain "Patch SIHAM" dédié aux corrections Web Services.

Cela dit, ce problème est actuellement bloquant, et au 14 février 2020, aucun correctif n'a été proposé par l'AMUE. De plus il apparaît que le WebService ListeAgentsWebService poserait également des difficultés de même nature.

Pour contourner ce problème, nous avons mis en place un goulot d'étranglement au niveau de notre frontal Apache pour que celui-ci ne laisse passer qu'une requête à la fois vers le serveur d'application Siham tout en gérant une file d'attente des requêtes.

Cette mise en oeuvre n'est pas naturelle dans le monde des applications web (les applications web supportent les appels/accès concurrents), aussi sur Apache (utilisé pour faire du ProxyPass en AJP), on a besoin de faire des paramétrages spécifiques et inhabituels.

On documente une solution d'un tel paramétrage ici (réalisé sur une centos 7.6) :

On configure apache pour utiliser le module worker et n'avoir qu'un seul process (server) à la fois ce qui permettra d'utiliser la directive max dans le proxypass :

conf.modules.d/00-mpm.conf

```
LoadModule mpm_worker_module modules/mod_mpm_worker.so

# ServerLimit à 1 pour pouvoir faire un max=1 dans le proxypass et faire ainsi goulot d'étranglement :
# 1 seule requête à la fois que le WS de siham qui ne supporte pas les appels concurrents
ServerLimit          1
StartServers         1
MaxRequestWorkers    512
ThreadsPerChild      512
MaxClients            512
ThreadLimit          512
# MaxConnectionsPerChild à 1 pour que si ça plante entre apache et tomcat,
# le fait que le thread qui gère cette requête/connexion soit dans un état 'incohérent' ne pose pas de pb :
# chaque thread n'étant utilisé qu'une fois pour 1 requête :
MaxConnectionsPerChild 1
```

Ainsi dans le virtualhost apache, on peut faire du proxypass en précisant qu'on ne laisse passer qu'une requête à la fois pour chacun des web services cibles :

```
# mise en place du goulot d'étranglement pour les WS siham ne supportant pas la concurrence
# conf en lien avec 00-mpm.conf
# flushpackets=on ... à voir l'utilité ?
ProxyPass /DossierAgentDateWebService ajp://sacha.univ-rouen.fr:8010/DossierAgentDateWebService retry=1
timeout=300 max=1 smax=50 flushpackets=on
ProxyPass /ListeAgentsWebService ajp://sacha.univ-rouen.fr:8010/ListeAgentsWebService retry=1 timeout=300 max=1
smax=50 flushpackets=on
ProxyPass / ajp://sacha.univ-rouen.fr:8010/ retry=1 timeout=3600
```

Session Timeout HRAccess

Le Web Service Siham DossierAgentDateWebService.RecupDonneesAgents tombe en erreur au bout d'un certain temps. L'erreur côté serveur mentionne une erreur de "VIRTUAL_SESSION_TIMEOUT" ; le web-service est alors Hors Service.

Pour contourner le problème, on a suivi les indication de la DA 75567 afin de résoudre le problème de timeout :

Ajout d'une ligne dans la table pp15 afin de modifier le timeout du serveur HRAccess

```
insert into pp15 (cdplph, idparm, valpar) select cdplph, 'OP_TIMEOUT', '999' from pp10 where cdpcom='1'
```

L'AMUE nous a signalé que "l'erreur "VIRTUAL_SESSION_TIMEOUT " est suivi dans la fiche 133967 prévue dans le patch SIHPRD 2.07.30 prévu dans le premier semestre 2020".

Problème de récupération en masse de dossiers agents

Lorsqu'on souhaite récupérer plusieurs dossiers agents (via le web service DossierAgentDateWebService) en une seule requête, si un dossier agent a un élément manquant (~ dossier incomplet) :

- a. celui-ci ne peut pas être récupéré

- b. et c'est l'ensemble de la requête qui tombe en erreur

Ce dernier point nous a amené à partager un code permettant :

1. de détecter les dossiers en erreur (dossier incomplet) et donc non récupérables par web-service
2. de récupérer l'ensemble des dossiers agents en
 - a. utilisant la possibilité de récupérer plusieurs dossiers en même temps
 - b. tout en prenant en compte le fait qu'un dossier parmi la liste peut être incomplet et faire planter l'ensemble de la requête.

Exemple en python : détection des dossiers agents en erreurs

```

# -*- coding: utf-8 -*-
from requests.auth import HTTPBasicAuth
from requests import Session
from zeep import Client
from zeep.transports import Transport
from time import time

user = 'siham'
password = 'hello'
LISTE_AGENTS_WSDL = 'https://siham.univ-ville.fr/ListeAgentsWebService/ListeAgentsWebService?WSDL'
DOSSIER_AGENT_WSDL = 'https://siham.univ-ville.fr/DossierAgentDateWebService/DossierAgentDateWebService?WSDL'

session = Session()
session.auth = HTTPBasicAuth(user, password)
transport = Transport(session=session)

client_liste_agents = Client(LISTE_AGENTS_WSDL, transport=transport)
print 'connexion1 ok ...'

client_dossier_agents = Client(DOSSIER_AGENT_WSDL, transport=transport)
print 'connexion2 ok ...'

debut = time()
agents = client_liste_agents.service.recupListeAgents(ParamRecupListeAgents={'nomUsuel': '%'})
print 'récup de ' + str(len(agents)) + ' agents en ' + str(time()-debut) + 'sec'

debut = time()
agents_matricules = []
agents_full = []
i = 0
n = 0
agents_erreurs = []
for agent in agents:
    i = i+1
    agents_matricules.append({'matricule':agent.matricule})
    if(i==20) :
        i = 0
        n = n+1
        try:
            agents_full += client_dossier_agents.service.RecupDonneesAgents(ParamAgentDateWS={'listeMatricules':
agents_matricules})
        except:
            for agent_matricule in agents_matricules:
                try:
                    agents_full += client_dossier_agents.service.RecupDonneesAgents(ParamAgentDateWS=
{'listeMatricules':agent_matricule})
                except:
                    agents_erreurs.append(agent_matricule)
            agents_matricules = []

if(len(agents_matricules)>0) :
    try:
        agents_full += client_dossier_agents.service.RecupDonneesAgents(ParamAgentDateWS={'listeMatricules':
agents_matricules})
    except:
        for agent_matricule in agents_matricules:
            try:
                agents_full += client_dossier_agents.service.RecupDonneesAgents(ParamAgentDateWS=
{'listeMatricules':agent_matricule})
            except:
                agents_erreurs.append(agent_matricule)

n = n+1

print 'récup complète de ' + str(len(agents_full)) + ' agents en ' + str(n) + ' requete(s) en ' + str(time()-
debut) + 'sec'
if(len(agents_erreurs)>0) :
    print str(len(agents_erreurs)) + ' agents en erreur : ' + str(agents_erreurs)

```

Exemple en Java : récupération des dossiers agents

Generation de code depuis le wsdl

Partie du pom.xml pour générer le client depuis le WSDL :

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>jaxws-maven-plugin</artifactId>
  <executions>
    <execution>
      <id>siham-wsdl-liste-agents</id>
      <goals>
        <goal>wsimport</goal>
      </goals>
      <configuration>
        <wsdlUrls>
          <wsdlUrl>https://siham.univ-ville.fr/ListeAgentsWebService
/ListeAgentsWebService?wsdl</wsdlUrl>
        </wsdlUrls>
        <packageName>fr.univville.siham.listeagents</packageName>
      </configuration>
    </execution>
    <execution>
      <id>siham-wsdl-dossier-agent</id>
      <goals>
        <goal>wsimport</goal>
      </goals>
      <configuration>
        <wsdlUrls>
          <wsdlUrl>https://siham.univ-ville.fr/DossierAgentDateWebService
/DossierAgentDateWebService?wsdl</wsdlUrl>
        </wsdlUrls>
        <packageName>fr.univville.siham.dossieragent</packageName>
      </configuration>
    </execution>
    <execution>
      <id>siham-wsdl-parametrage</id>
      <goals>
        <goal>wsimport</goal>
      </goals>
      <configuration>
        <wsdlUrls>
          <wsdlUrl>https://siham.univ-ville.fr/DossierParametrageWebService
/DossierParametrageWebService?wsdl</wsdlUrl>
        </wsdlUrls>
        <packageName>fr.univville.siham.parametrage</packageName>
      </configuration>
    </execution>
  </executions>
  <configuration>
    <xauthFile>${basedir}/src/main/resources/siham/authFile</xauthFile>
    <verbose>true</verbose>
    <sourceDestDir>${basedir}/src/main/java</sourceDestDir>
  </configuration>
</plugin>
```

Avec le fichier src/main/resources/siham/authFile :

```
https://siham:hello@siham.univ-ville.fr/ListeAgentsWebService/ListeAgentsWebService?wsdl
https://siham:hello@siham.univ-ville.fr/DossierAgentDateWebService/DossierAgentDateWebService?wsdl
https://siham:hello@siham.univ-ville.fr/DossierParametrageWebService/DossierParametrageWebService?wsdl
```

Code Java

```
import java.net.Authenticator;
import java.net.PasswordAuthentication;
import java.net.URL;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.xml.ws.BindingProvider;
import javax.xml.ws.handler.MessageContext;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.multipart.MultipartException;

import fr.univville.siham.dossieragent.DossierAgentDateWebService;
import fr.univville.siham.dossieragent.Individu;
import fr.univville.siham.dossieragent.Matricule;
import fr.univville.siham.dossieragent.ParamAgentDateWS;
import fr.univville.siham.dossieragent.ReferentielService;
import fr.univville.siham.dossieragent.WebServiceException_Exception;
import fr.univville.siham.listeagents.IListeAgentsWebService;
import fr.univville.siham.listeagents.ListeAgentsWS;
import fr.univville.siham.listeagents.ParamListeAgents;
import fr.univville.siham.listeagents.ResultatsListeAgents;
import fr.univville.siham.listeagents.WebServiceException_Exception;
import fr.univville.siham.parametrage.DossierParametrageWebService;
import fr.univville.siham.parametrage.ListeNomenclatures;
import fr.univville.siham.parametrage.ListeUO;
import fr.univville.siham.parametrage.NomenclatureService;
import fr.univville.siham.parametrage.ParamNomenclature;
import fr.univville.siham.parametrage.ParamStructure;
import fr.univville.siham.parametrage.Repertoire;
import fr.univville.siham.parametrage.Structures;

public class SihamService {

    private static final int NB_AGENTS_PER_REQUEST = 20;

    protected final Logger log = LoggerFactory.getLogger(this.getClass());

    private IListeAgentsWebService listeAgentsWebService;

    private DossierAgentDateWebService dossierAgentDateWebService;

    private DossierParametrageWebService parametrageWebService;

    private String url;

    private String username;

    private String password;

    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public void setPassword(String password) {
```

```

        this.password = password;
    }

    private Map<String, List<String>> getWsHeaders() {
        Map<String, List<String>> headers = new HashMap<String, List<String>>();
        headers.put("Username", Collections.singletonList(username));
        headers.put("Password", Collections.singletonList(password));
        return headers;
    }

    private IListeAgentsWebService getListeAgentsWebService() {
        if(listeAgentsWebService==null) {
            try {
                initAuthenticator();
                URL listeAgentsWsUrl = new URL(url + "/ListeAgentsWebService
/ListeAgentsWebService?wsdl");
                ListeAgentsWS listeAgentsWS = new ListeAgentsWS(listeAgentsWsUrl);
                listeAgentsWebService = listeAgentsWS.getListeAgentsWebServiceImplPort();
                Map<String, Object> req_ctx = ((BindingProvider)listeAgentsWebService).
getRequestContext();
                req_ctx.put(MessageContext.HTTP_REQUEST_HEADERS, getWsHeaders());
                log.info("Siham listeAgentsWebService OK");
            } catch(Exception ex) {
                throw new RuntimeException("Exception during initialization of siham ws
ListeAgentsWebService", ex);
            } finally {
                resetAuthenticator();
            }
        }
        return listeAgentsWebService;
    }

    private DossierAgentDateWebService getDossierAgentDateWebService() {
        if(dossierAgentDateWebService==null) {
            try {
                initAuthenticator();
                URL dossierAgentWsUrl = new URL(url + "/DossierAgentDateWebService
/DossierAgentDateWebService?wsdl");
                ReferentielService referentielService = new ReferentielService
(dossierAgentWsUrl);
                dossierAgentDateWebService = referentielService.
getDossierAgentDateWebServiceImplPort();
                Map<String, Object> req_ctx_ref = ((BindingProvider)dossierAgentDateWebService).
getRequestContext();
                req_ctx_ref.put(MessageContext.HTTP_REQUEST_HEADERS, getWsHeaders());
                log.info("Siham dossierAgentDateWebService OK");
            } catch(Exception ex) {
                throw new RuntimeException("Exception during initialization of siham ws
DossierAgentDateWebService", ex);
            } finally {
                resetAuthenticator();
            }
        }
        return dossierAgentDateWebService;
    }

    private DossierParametrageWebService getDossierParametrageWebService() {
        if(parametrageWebService==null) {
            try {
                initAuthenticator();
                URL parametrageWsUrl = new URL(url + "/DossierParametrageWebService
/DossierParametrageWebService?wsdl");
                NomenclatureService nomenclatureService = new NomenclatureService
(parametrageWsUrl);
                parametrageWebService = nomenclatureService.
getDossierParametrageWebServiceImplPort();
                Map<String, Object> req_ctx_par = ((BindingProvider)parametrageWebService).
getRequestContext();
                req_ctx_par.put(MessageContext.HTTP_REQUEST_HEADERS, getWsHeaders());
                log.info("Siham parametrageWebService OK");
            } catch(Exception ex) {

```

```

        throw new RuntimeException("Exception during initialization of siham ws
DossierAgentDateWebService", ex);
    } finally {
        resetAuthenticator();
    }
}
return parametragewebService;
}

private void initAuthenticator() {
    Authenticator.setDefault(new Authenticator() {
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(
                username,
                password.toCharArray());
        }
    });
}

private void resetAuthenticator() {
    Authenticator.setDefault(null);
}

protected List<String> getAgentsMatricules(String nomUsuel) {
    List<String> matricules = new ArrayList<String>();
    ParamListeAgents paramRecupListeAgents = new ParamListeAgents();
    paramRecupListeAgents.setNomUsuel(nomUsuel);
    try {
        List<ResultatsListeAgents> agents= getIListeAgentsWebService().recupListeAgents
(paramRecupListeAgents);
        for(ResultatsListeAgents agent : agents) {
            matricules.add(agent.getMatricule());
        }
        log.debug(agents.size() + " matricules d'agents récupérés depuis Siham");
    } catch (WebServiceException_Exception e) {
        throw new MultipartException("Erreur lors de la récupération agents depuis web-service
siham", e);
    }
    return matricules;
}

public Individu getSihamIndividu(String matricule) {
    Map<String, Individu> sihamIndividus = getSihamIndividus(Arrays.asList(new String[]
{matricule}));
    return sihamIndividus.get(matricule);
}

public synchronized Map<String, Individu> getSihamIndividus(List<String> matricules) {
    Map<String, Individu> sihamIndividus = new HashMap<String, Individu>();
    ParamAgentDateWS paramAgentDateWS = new ParamAgentDateWS();
    int i = 0;
    int n = 0;
    List<String> matriculesErreurs = new ArrayList<String>();
    for(String matricule: matricules) {
        i++;
        n++;
        Matricule matriculeObj = new Matricule();
        matriculeObj.setMatricule(matricule);
        paramAgentDateWS.getListeMatricules().add(matriculeObj);
        if(i==NB_AGENTS_PER_REQUEST || n==matricules.size()) {
            try {
                List<Individu> agents = getDossierAgentDateWebService().
recupDonneesAgents(paramAgentDateWS);
                for(Individu agent: agents) {
                    sihamIndividus.put(agent.getDonneesPersonnelles().
getMatricule(), agent);
                }
            } catch (WebserviceException_Exception e) {
                log.info("Erreur lors de la récupération agents depuis web-service

```

```

siham", e);

        for(Matricule agentMatricule : paramAgentDateWS.getListeMatricules()) {
            ParamAgentDateWS agentParamAgentDateWS = new ParamAgentDateWS();
            agentParamAgentDateWS.getListeMatricules().add(agentMatricule);
            try {
                List<Individu> agents = getDossierAgentDateWebService().
recupDonneesAgents(agentParamAgentDateWS);
                for(Individu agent: agents) {
                    sihamIndividus.put(agent.
getDonneesPersonnelles().getMatricule(), agent);
                }
            } catch (WebserviceException_Exception ee) {
                log.warn("Erreur lors de la récupération de cet agent
depuis web-service siham : " + agentMatricule.getMatricule(), ee);
                sihamIndividus.put(agentMatricule.getMatricule(), new
SihamIndividuAgentInError());
                matriculesErreurs.add(agentMatricule.getMatricule());
            }
        }
        paramAgentDateWS = new ParamAgentDateWS();
        i = 0;
    }
    }
    if(matriculesErreurs.size() > 0) {
        log.error("Matricules en erreur lors de la récupération de ces agents depuis web-
service siham : " + matriculesErreurs);
    }
    log.info(sihamIndividus.size() + " agents récupérés depuis Siham dont " + matriculesErreurs.
size() + " en erreur.");
    return sihamIndividus;
}

public Map<String, Individu> getSihamIndividuFromNomUsuel(String nomUsuel) {
    List<String> matricules = getAgentsMatricules(nomUsuel);
    Map<String, Individu> sihamIndividus = getSihamIndividus(matricules);
    return sihamIndividus;
}

public List<Structures> getSihamStructures() {
    List<Structures> structures;
    try {
        ParamStructure paramStructure = new ParamStructure();
        paramStructure.setCodeAdministration("");
        paramStructure.setDateObservation("");
        paramStructure.getListeUO().add(new ListeUO());
        structures = getDossierParametrageWebService().recupStructures(paramStructure);
    } catch (fr.univille.siham.parametrage.WebserviceException_Exception e) {
        throw new RuntimeException("Erreur lors de la récupération des structures depuis web-
service siham", e);
    }
    log.info(structures.size() + " structures récupérées depuis Siham");
    return structures;
}

public List<Repertoire> getSihamNomenclaturesRepertoires() {
    List<Repertoire> repertoires;
    try {
        ParamNomenclature paramNomenclature = new ParamNomenclature();
        paramNomenclature.setCodeAdministration("");
        paramNomenclature.setDateObservation("");
        paramNomenclature.getListeNomenclatures().add(new ListeNomenclatures());
        repertoires = getDossierParametrageWebService().recupNomenclaturesRH(paramNomenclature);
    } catch (fr.univille.siham.parametrage.WebserviceException_Exception e) {
        throw new RuntimeException("Erreur lors de la récupération des répertoires de
nomenclatures depuis web-service siham", e);
    }
    log.info(repertoires.size() + " répertoires de nomenclatures récupérées depuis Siham");
    return repertoires;
}

```

```
}
```

Classe SihamIndividuAgentInError :

```
import fr.univrouen.multipass.domain.siham.dossieragent.Individu;

public class SihamIndividuAgentInError extends Individu {

}
```

Descripteur Spring

En spring, on pourra initialiser le service ainsi :

```
<bean id="sihamService" class="fr.univville.siham.SihamService">
    <property name="url" value="https://siham.univ-ville.fr/">
        <property name="username" value="siham"/>
        <property name="password" value="hello"/>
    </property>
</bean>
```

Usage

```
List<Structures> sihamStructures = sihamService.getSihamStructures();
// ...

List<Repertoire> sihamRepertoiresNomenclatures = sihamService.getSihamNomenclaturesRepertoires();
// ...

Map<String, Individu> sihamIndividus = sihamService.getSihamIndividuFromNomUsuel("%");
for(String matricule : sihamIndividus.keySet()) {
    Individu sihamIndividu = sihamIndividus.get(matricule);
    if(sihamIndividu instanceof SihamIndividuAgentInError) {
        // l'individu siham est en erreur - ne rien faire -> ne pas le supprimer notamment ;- )
    } else {
        // ...
    }
}
```

Contrainte supplémentaire Web Service Siham (février 2022 - SIHPRD.2.09.52).

Suite à la mise à jour des WebService Siham en janvier/février 2022, une nouvelle contrainte est apparue sur la récupération des matricules

La méthode recupListeAgents de /ListeAgentsWebService/ListeAgentsWebService?wsdl ne supporte pas l'absence de certains paramètres. Il est nécessaire de les positionner à vide, sous peine d'avoir une erreur du type

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <soap:Fault><faultcode>soap:Server</faultcode><faultstring>Fault occurred while processing.<
    /faultstring></soap:Fault>
    </soap:Body>
</soap:Envelope>
```

Ainsi la requête minimale est maintenant la suivante :

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sih="http://siham.amue.fr">
  <soapenv:Header/>
  <soapenv:Body>
    <sih:recupListeAgents>
      <ParamRecupListeAgents>
        <listeTypeContrat>
          <codeTypeContrat></codeTypeContrat>
          <modeGest></modeGest>
        </listeTypeContrat>
        <temEtat>A</temEtat>
        <temoinValide></temoinValide>
      </ParamRecupListeAgents>
    </sih:recupListeAgents>
  </soapenv:Body>
</soapenv:Envelope>

```

En Java, on a ainsi modifié notre code ainsi, suite à cette mise à jour :

```

...
ParamListeAgents paramRecupListeAgents = new ParamListeAgents();
paramRecupListeAgents.setTemEtat("A");

/*
HACK WS SIHAM
Depuis la mäj de février 2022, ces paramètres, même vides, doivent figurer dans l'appel SOAP
Sinon ça plante.
*/
paramRecupListeAgents.setTemoinValide("");
ListeTypeContrat dummyListeTypeContrat = new ListeTypeContrat();
dummyListeTypeContrat.setCodeTypeContrat("");
dummyListeTypeContrat.setModeGest("");
paramRecupListeAgents.getListeTypeContrat().add(dummyListeTypeContrat);
/*
FIN HACK WS SIHAM
*/

try {
    List<ResultatsListeAgents> agents= getIListeAgentsWebService().recupListeAgents(paramRecupListeAgents);
    ...
}

```