package com.webank.wedatasphere.dss.appconn.core.impl;

import com.webank.wedatasphere.dss.appconn.core.AppConn

import com.webank.wedatasphere.dss.appconn.core.excepti

import com.webank.wedatasphere.dss.appconn.core.excepti

import com.webank.wedatasphere.dss.appconn.core.ext.Onl

import com.webank.wedatasphere.dss.standard.app.sso.req

import com.webank.wedatasphere.dss.standard.common.core

import com.webank.wedatasphere.dss.standard.common.core

import com.webank.wedatasphere.dss.standard.common.desc

import com.webank.wedatasphere.dss.standard.common.exce

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.Arrays;

import java.util.List;

import java.util.Objects;

import java.util.stream.Collectors;

public abstract class AbstractAppConn implements AppCon

private AppDesc appDesc;

private ListAppStandard appStandards;

protected final ListString appStandardMethodHeader

protected final Logger logger = LoggerFactory.getLo

@Override

public ListAppStandard getAppStandards() {

if(appStandards == null) {

synchronized (appStandardMethodHeader) {

if(appStandards == null) {

try {

init();

} catch (AppConnErrorException e) {

throw new AppConnWarnException(

}

}

}

}

return appStandards;

}

protected abstract void initialize();

@Override

public void init() throws AppConnErrorException {

initialize();

appStandards = Arrays.stream(getClass().getDecl

String methodName = method.getName();

if(appStandardMethodHeader.stream().anyMatc

AppStandard.class.isAssignableFrom(meth

try {

return (AppStandard) method.invoke(

} catch (ReflectiveOperationException e

logger.warn(methodName + " execute

}

}

return null;

}).filter(Objects::nonNull).collect(Collectors.

if(this instanceof OnlySSOAppConn) {

SSORequestService ssoRequestService = ((Onl

for(AppStandard appStandard : appStandards)

if(appStandard instanceof AppIntegratio

((AppIntegrationStandardSSORequestS

}

try {

appStandard.init();

} catch (AppStandardErrorException e) {

throw new AppConnErrorException(e.g

}

}

}

}

@Override

public AppDesc getAppDesc() {

return appDesc;

}

@Override

public void setAppDesc(AppDesc appDesc) {

this.appDesc = appDesc;

}

}

package com.webank.wedatasphere.dss.appconn.core.impl;

import com.webank.wedatasphere.dss.appconn.core.excepti

import com.webank.wedatasphere.dss.appconn.core.ext.Onl

import com.webank.wedatasphere.dss.standard.app.sso.SSO

import com.webank.wedatasphere.dss.standard.app.sso.SSO

import com.webank.wedatasphere.dss.standard.app.sso.ori

import com.webank.wedatasphere.dss.standard.app.sso.use

import com.webank.wedatasphere.dss.standard.app.sso.use

import com.webank.wedatasphere.dss.standard.common.util

import com.webank.wedatasphere.dss.standard.sso.utils.S

import org.apache.commons.collections4.CollectionUtils;

import org.slf4j.LoggerFactory;

public abstract class AbstractOnlySSOAppConn extends Ab

private SSOIntegrationStandard SSO\_INTEGRATION\_STAN

@Override

public final void init() throws AppConnErrorExcepti

SSOIntegrationStandardFactory ssoIntegrationSta

AppStandardClassUtils.getInstance(getAp

ssoIntegrationStandardFactory.init();

SSO\_INTEGRATION\_STANDARD = ssoIntegrationStanda

LoggerFactory.getLogger(AbstractOnlySSOAppConn.

ssoIntegrationStandardFactory.getClass(

if(CollectionUtils.isNotEmpty(getAppDesc().getA

getAppDesc().getAppInstances().forEach(appI

SSOUserService ssoUserService = SSO\_INT

if(ssoUserService instanceof SSOUserSer

((SSOUserServiceImpl) ssoUserServic

}

});

}

SSOHelper.setSSOBuilderService(SSO\_INTEGRATION\_

super.init();

}

@Override

public final SSOIntegrationStandard getOrCreateSSOS

return SSO\_INTEGRATION\_STANDARD;

}

}

package com.webank.wedatasphere.dss.appconn.loader.claz

import java.net.URL;

import java.net.URLClassLoader;

public class AppConnClassLoader extends URLClassLoader

public AppConnClassLoader(URL[] urls, ClassLoader p

super(urls, parent);

}

@Override

public Class? loadClass(String name) throws ClassNo

return loadClass(name, false);

}

@Override

protected Class? loadClass(String name, boolean res

return super.loadClass(name, resolve);

}

}

package com.webank.wedatasphere.dss.appconn.loader.load

import com.webank.wedatasphere.dss.appconn.core.AppConn

public interface AppConnLoader {

AppConn getAppConn(String appConnName, String spi,

}

package com.webank.wedatasphere.dss.appconn.loader.load

import com.webank.wedatasphere.dss.appconn.loader.conf.

import org.apache.commons.lang.ClassUtils;

import org.apache.commons.lang.StringUtils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppConnLoaderFactory {

private static final Logger logger = LoggerFactory.

private static Class? extends AppConnLoader clazz =

private static AppConnLoader appConnLoader = null;

@SuppressWarnings("unchecked")

public static AppConnLoader getAppConnLoader(){

if (appConnLoader == null){

synchronized (AppConnLoaderFactory.class){

if (appConnLoader == null){

String className = AppConnLoaderCon

if (StringUtils.isNotBlank(classNam

try{

clazz = ClassUtils.getClass

}catch(ClassNotFoundException e

logger.warn(String.format("

}

}

try {

appConnLoader = clazz.newInstan

} catch (Exception e) {

logger.error(String.format("Can

}

logger.info("Use {} to load all App

}

}

}

return appConnLoader;

}

}

package com.webank.wedatasphere.dss.appconn.loader.load

import com.webank.wedatasphere.dss.appconn.core.AppConn

import com.webank.wedatasphere.dss.appconn.loader.clazz

import com.webank.wedatasphere.dss.appconn.loader.excep

import com.webank.wedatasphere.dss.appconn.loader.utils

import com.webank.wedatasphere.dss.common.utils.DSSExce

import com.webank.wedatasphere.dss.standard.common.util

import org.apache.commons.io.FileUtils;

import org.apache.commons.lang.StringUtils;

import org.apache.linkis.common.conf.BDPConfiguration;

import org.apache.linkis.common.exception.ErrorExceptio

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.io.File;

import java.net.URL;

import java.nio.file.Paths;

import java.util.List;

import java.util.Properties;

public class CommonAppConnLoader implements AppConnLoad

private static final String LIB\_NAME = "lib";

private static final String APP\_CONN\_PROPERTIES\_NAM

private static final Logger LOGGER = LoggerFactory.

@Override

public AppConn getAppConn(String appConnName, Strin

ClassLoader currentClassLoader = Thread.current

String libPathUrl;

if (StringUtils.isNotEmpty(homePath)){

libPathUrl = new File(homePath, LIB\_NAME).g

} else {

libPathUrl = Paths.get(AppConnUtils.getAppC

}

LOGGER.info("The libPath url of AppConn {} is {

ListURL jars = AppConnUtils.getJarsUrlsOfPath(l

ClassLoader classLoader = AppStandardClassUtils

Thread.currentThread().setContextClassLoader(cl

String fullClassName;

if (StringUtils.isEmpty(spi)) {

try {

fullClassName = AppConnUtils.getAppConn

} catch (NoSuchAppConnException e) {

Thread.currentThread().setContextClassL

throw e;

}

} else {

fullClassName = spi;

}

Class? clazz = null;

try {

clazz = classLoader.loadClass(fullClassName

} catch (ClassNotFoundException e) {

Thread.currentThread().setContextClassLoade

DSSExceptionUtils.dealErrorException(70062,

}

AppConn retAppConn = (AppConn) clazz.newInstanc

File appConnPropertiesPathUrl;

if (StringUtils.isNotEmpty(homePath)){

appConnPropertiesPathUrl = new File(homePat

} else {

appConnPropertiesPathUrl = Paths.get(AppCon

}

if(appConnPropertiesPathUrl.exists() appConnPr

Properties properties = new Properties();

properties.load(FileUtils.openInputStream(a

if(!properties.isEmpty()) {

LOGGER.info("AppConn {} try to load {},

properties.forEach((key, value) - {

if(key != null StringUtils.isNotBl

value != null StringUtils.

BDPConfiguration.setIfNotExists

}

});

}

}

Thread.currentThread().setContextClassLoader(cu

LOGGER.info("Loaded appConn {} with class {}.",

return retAppConn;

}

}

package com.webank.wedatasphere.dss.appconn.loader.util

import com.webank.wedatasphere.dss.appconn.core.AppConn

import com.webank.wedatasphere.dss.appconn.loader.excep

import com.webank.wedatasphere.dss.common.utils.DSSComm

import org.apache.linkis.common.conf.CommonVars;

import java.io.File;

import java.io.IOException;

import java.lang.reflect.Modifier;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.ArrayList;

import java.util.Enumeration;

import java.util.List;

import java.util.jar.JarEntry;

import java.util.jar.JarFile;

import org.apache.commons.lang.StringUtils;

public class AppConnUtils {

public static final String JAR\_SUF\_NAME = ".jar";

public static final String APPCONN\_DIR\_NAME = "dss-

public static final CommonVarsString APPCONN\_HOME\_P

new File(DSSCommonUtils.DSS\_HOME.getValue(), AP

public static String getAppConnHomePath() {

return APPCONN\_HOME\_PATH.acquireNew();

}

public static String getAppConnClassName(String app

ClassLoade

ListString jars = getJarsOfPath(libPath);

for (String jar : jars) {

for (String clazzName : getClassNameFrom(ja

if (isChildClass(clazzName, AppConn.cla

return clazzName;

}

}

}

throw new NoSuchAppConnException("Cannot find a

}

public static ListString getJarsOfPath(String path)

File file = new File(path);

ListString jars = new ArrayList();

if (file.listFiles() != null) {

for (File f : file.listFiles()) {

if (!f.isDirectory() f.getName().endsW

jars.add(f.getPath());

}

}

}

return jars;

}

public static ListURL getJarsUrlsOfPath(String path

File file = new File(path);

ListURL jars = new ArrayList();

if (file.listFiles() != null) {

for (File f : file.listFiles()) {

if (!f.isDirectory() f.getName().endsW

jars.add(f.toURI().toURL());

}

}

}

return jars;

}

private static ListString getClassNameFrom(String j

ListString fileList = new ArrayList();

JarFile jarFile = new JarFile(new File(jarName)

EnumerationJarEntry en = jarFile.entries();

while (en.hasMoreElements()) {

String name1 = en.nextElement().getName();

if (!name1.endsWith(".class")) {

continue;

}

String name2 = name1.substring(0, name1.las

fileList.add(name3);

}

return fileList;

}

private static boolean isChildClass(String classNam

if (StringUtils.isEmpty(className)) {

return false;

}

Class clazz = null;

try {

clazz = classLoader.loadClass(className);

if (Modifier.isAbstract(clazz.getModifiers(

return false;

}

if (Modifier.isInterface(clazz.getModifiers

return false;

}

} catch (Throwable t) {

return false;

}

return parentClazz.isAssignableFrom(clazz);

}

}

package com.webank.wedatasphere.dss.appconn.loader.util

import org.apache.linkis.common.exception.ErrorExceptio

public class ExceptionHelper {

public static void dealErrorException(int errorCode

ErrorException errorException = new ErrorExcept

errorException.initCause(t);

throw errorException;

}

}

package com.webank.wedatasphere.dss.appconn.manager.con

import org.apache.linkis.common.conf.CommonVars;

public class AppConnManagerClientConfiguration {

public final static CommonVarsString DSS\_APPCONN\_CL

public final static CommonVarsString LINKIS\_ADMIN\_U

}

package com.webank.wedatasphere.dss.appconn.manager.ent

import com.webank.wedatasphere.dss.common.entity.Resour

public class AppConnInfoImpl implements AppConnInfo {

private String appConnName;

private String className;

private Resource appConnResource;

private String reference;

public void setAppConnName(String appConnName) {

this.appConnName = appConnName;

}

public void setClassName(String className) {

this.className = className;

}

public void setAppConnResource(Resource appConnReso

this.appConnResource = appConnResource;

}

@Override

public String getAppConnName() {

return appConnName;

}

@Override

public String getClassName() {

return className;

}

@Override

public Resource getAppConnResource() {

return appConnResource;

}

@Override

public String getReference() {

return reference;

}

public void setReference(String reference) {

this.reference = reference;

}

@Override

public String toString() {

return "AppConnInfoImpl{" +

"appConnName='" + appConnName + '\'' +

", className='" + className + '\'' +

", appConnResource=" + appConnResource

", reference='" + reference + '\'' +

'}';

}

}

package com.webank.wedatasphere.dss.appconn.manager.ent

public class AppInstanceInfoImpl implements AppInstance

private Long id;

private String url;

private String homepageUri;

private String labels;

private String enhanceJson;

@Override

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

@Override

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

@Override

public String getHomepageUri() {

return homepageUri;

}

public void setHomepageUri(String homepageUri) {

this.homepageUri = homepageUri;

}

@Override

public String getLabels() {

return labels;

}

public void setLabels(String labels) {

this.labels = labels;

}

@Override

public String getEnhanceJson() {

return enhanceJson;

}

public void setEnhanceJson(String enhanceJson) {

this.enhanceJson = enhanceJson;

}

@Override

public String toString() {

return "AppInstanceInfoImpl{" +

"id=" + id +

", url='" + url + '\'' +

", homepageUrl='" + homepageUri + '\''

", labels='" + labels + '\'' +

", enhanceJson='" + enhanceJson + '\''

'}';

}

}

package com.webank.wedatasphere.dss.appconn.manager.exc

import com.webank.wedatasphere.dss.common.exception.DSS

public class AppConnHomeNotExistsWarnException extends

public AppConnHomeNotExistsWarnException(int errCod

super(errCode, desc);

}

public AppConnHomeNotExistsWarnException(int errCod

super(errCode, desc);

initCause(t);

}

}

package com.webank.wedatasphere.dss.appconn.manager.ser

import com.webank.wedatasphere.dss.appconn.manager.acti

import com.webank.wedatasphere.dss.appconn.manager.acti

import com.webank.wedatasphere.dss.appconn.manager.acti

import com.webank.wedatasphere.dss.appconn.manager.conf

import com.webank.wedatasphere.dss.appconn.manager.enti

import com.webank.wedatasphere.dss.appconn.manager.enti

import com.webank.wedatasphere.dss.appconn.manager.resu

import com.webank.wedatasphere.dss.appconn.manager.resu

import com.webank.wedatasphere.dss.appconn.manager.resu

import org.apache.linkis.common.conf.Configuration;

import org.apache.linkis.httpclient.config.ClientConfig

import org.apache.linkis.httpclient.dws.DWSHttpClient;

import org.apache.linkis.httpclient.dws.authentication.

import org.apache.linkis.httpclient.dws.config.DWSClien

import org.apache.linkis.httpclient.dws.config.DWSClien

import java.util.List;

import static com.webank.wedatasphere.dss.appconn.manag

public class AppConnInfoServiceImpl implements AppConnI

private ClientConfig clientConfig = DWSClientConfig

.connectionTimeout(300000).discoveryEnabled(fal

.setAuthTokenValue(AppConnManagerClientConfigur

private DWSHttpClient client = new DWSHttpClient((D

@Override

public List? extends AppConnInfo getAppConnInfos()

GetAppConnInfosAction getAppConnInfosAction =ne

getAppConnInfosAction.setUser(LINKIS\_ADMIN\_USER

GetAppConnInfosResult result = (GetAppConnInfos

return result.getAppConnInfos();

}

@Override

public AppConnInfo getAppConnInfo(String appConnNam

GetAppConnInfoAction action = new GetAppConnInf

action.setAppConnName(appConnName);

action.setUser(LINKIS\_ADMIN\_USER.getValue());

GetAppConnInfoResult result = (GetAppConnInfoRe

return result.getAppConnInfo();

}

@Override

public List? extends AppInstanceInfo getAppInstance

return getAppInstancesByAppConnName(appConnInfo

}

@Override

public List? extends AppInstanceInfo getAppInstance

GetAppInstancesAction action = new GetAppInstan

action.setAppConnName(appConnName);

action.setUser(LINKIS\_ADMIN\_USER.getValue());

GetAppInstancesResult result = (GetAppInstances

return result.getAppInstanceInfos();

}

}

package com.webank.wedatasphere.dss.appconn.manager.ser

import com.webank.wedatasphere.dss.appconn.loader.utils

import com.webank.wedatasphere.dss.appconn.manager.enti

import com.webank.wedatasphere.dss.appconn.manager.exce

import com.webank.wedatasphere.dss.appconn.manager.util

import com.webank.wedatasphere.dss.common.entity.Resour

import com.webank.wedatasphere.dss.common.exception.DSS

import com.webank.wedatasphere.dss.common.utils.ZipHelp

import org.apache.commons.lang.StringUtils;

import org.apache.linkis.bml.client.BmlClient;

import org.apache.linkis.bml.client.BmlClientFactory;

import org.apache.linkis.common.utils.Utils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.io.File;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.Arrays;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.IntStream;

public class AppConnResourceServiceImpl implements AppC

private static final Logger LOGGER = LoggerFactory.

private BmlClient bmlClient = BmlClientFactory.crea

@Override

public String getAppConnHome(AppConnInfo appConnInf

File appConnHomePath = new File(AppConnUtils.ge

String appConnName = appConnInfo.getAppConnName

if(!appConnHomePath.exists()) {

throw new AppConnHomeNotExistsWarnException

} else if (!appConnHomePath.isDirectory()) {

throw new AppConnHomeNotExistsWarnException

}

File appConnPath = new File(appConnHomePath, ap

Resource resource = appConnInfo.getAppConnResou

if(!appConnPath.exists() !appConnPath.mkdir())

throw new AppConnHomeNotExistsWarnException

}

if(AppConnIndexFileUtils.isLatestIndex(appConnP

return appConnPath.getPath();

}

if (StringUtils.isNotBlank(appConnInfo.getRefer

return appConnPath.getPath();

}

LOGGER.info("Try to download latest resource {}

resource.getVersion(), appConnName);

String zipFilePath = new File(appConnHomePath,

bmlClient.downloadResource(Utils.getJvmUser(),

if(!deleteAppConnDir(appConnPath)) {

throw new AppConnHomeNotExistsWarnException

}

try {

ZipHelper.unzip(zipFilePath);

} catch (DSSErrorException e) {

throw new AppConnHomeNotExistsWarnException

}

File oldIndexFile = AppConnIndexFileUtils.getIn

if (oldIndexFile != null !oldIndexFile.delete(

throw new AppConnHomeNotExistsWarnException

}

File[] historyZipFiles = appConnHomePath.listFi

if(historyZipFiles.length 2) {

ListFile files = Arrays.stream(historyZipFi

IntStream.range(0, files.size() - 2).forEac

}

Path indexFile = Paths.get(appConnPath.getPath(

try {

Files.createFile(indexFile);

} catch (IOException e) {

throw new AppConnHomeNotExistsWarnException

+ appConnName, e);

}

return appConnPath.getPath();

}

public boolean deleteAppConnDir(File f){

if(f.isDirectory()){

File[] files = f.listFiles();

for (File key : files) {

if(key.isFile()){

key.delete();

}else{

deleteAppConnDir(key);

}

}

}

return f.delete();

}

}

package com.webank.wedatasphere.dss.appconn.manager;

import com.webank.wedatasphere.dss.appconn.core.AppConn

import com.webank.wedatasphere.dss.appconn.core.excepti

import com.webank.wedatasphere.dss.appconn.manager.impl

import java.util.List;

import java.util.stream.Collectors;

public interface AppConnManager {

void init();

ListAppConn listAppConns();

default T extends AppConn ListT listAppConns(ClassT

return listAppConns().stream().filter(appConnCl

}

AppConn getAppConn(String appConnName);

default T extends AppConn T getAppConn(ClassT appCo

ListAppConn appConns = listAppConns().stream().

if(appConns.isEmpty()) {

throw new AppConnWarnException(25344, "Cann

} else if(appConns.size() 1) {

throw new AppConnWarnException(25344, "More

}

return (T) appConns.get(0);

}

void reloadAppConn(String appConnName);

String getAppConnHomePath(String appConnName);

static void setLazyLoad() {

AbstractAppConnManager.setLazyLoad();

}

static AppConnManager getAppConnManager() {

return AbstractAppConnManager.getAppConnManager

}

}

package com.webank.wedatasphere.dss.appconn.manager.ent

import com.webank.wedatasphere.dss.common.entity.Resour

public interface AppConnInfo {

String getAppConnName();

String getClassName();

Resource getAppConnResource();

String getReference();

}

package com.webank.wedatasphere.dss.appconn.manager.ent

public interface AppInstanceInfo {

Long getId();

String getUrl();

String getHomepageUri();

String getLabels();

String getEnhanceJson();

}

package com.webank.wedatasphere.dss.appconn.manager.exc

import com.webank.wedatasphere.dss.common.exception.DSS

public class AppConnIndexFileWarnException extends DSSR

public AppConnIndexFileWarnException(int errCode, S

super(errCode, desc);

}

}

package com.webank.wedatasphere.dss.appconn.manager.imp

import com.webank.wedatasphere.dss.appconn.core.AppConn

import com.webank.wedatasphere.dss.appconn.loader.loade

import com.webank.wedatasphere.dss.appconn.loader.loade

import com.webank.wedatasphere.dss.appconn.manager.AppC

import com.webank.wedatasphere.dss.appconn.manager.enti

import com.webank.wedatasphere.dss.appconn.manager.enti

import com.webank.wedatasphere.dss.appconn.manager.serv

import com.webank.wedatasphere.dss.appconn.manager.serv

import com.webank.wedatasphere.dss.appconn.manager.util

import com.webank.wedatasphere.dss.common.exception.DSS

import com.webank.wedatasphere.dss.common.label.DSSLabe

import com.webank.wedatasphere.dss.common.label.EnvDSSL

import com.webank.wedatasphere.dss.common.utils.ClassUt

import com.webank.wedatasphere.dss.common.utils.DSSComm

import com.webank.wedatasphere.dss.common.utils.DSSExce

import com.webank.wedatasphere.dss.standard.common.desc

import com.webank.wedatasphere.dss.standard.common.desc

import org.apache.commons.lang.StringUtils;

import org.apache.linkis.common.utils.Utils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.concurrent.TimeUnit;

import java.util.function.Consumer;

import java.util.stream.Collectors;

public abstract class AbstractAppConnManager implements

private static final Logger LOGGER = LoggerFactory.

private final AppConnLoader appConnLoader = AppConn

private final MapString, AppConn appConns = new Has

private boolean isLoaded = false;

private ListAppConn appConnList = null;

AppConnInfoService appConnInfoService;

private AppConnResourceService appConnResourceServi

private AppConnRefreshThread appConnRefreshThread;

private static AppConnManager appConnManager;

private static boolean lazyLoad = false;

public static void setLazyLoad() {

lazyLoad = true;

}

public static AppConnManager getAppConnManager() {

if (appConnManager != null) {

return appConnManager;

}

synchronized (AbstractAppConnManager.class) {

if (appConnManager == null) {

appConnManager = ClassUtils.getInstance

LOGGER.info("The instance of AppConnMan

appConnManager.init();

}

}

return appConnManager;

}

@Override

public void init() {

appConnInfoService = createAppConnInfoService()

LOGGER.info("The instance of AppConnInfoService

appConnResourceService = createAppConnResourceS

LOGGER.info("The instance of AppConnResourceSer

if (!lazyLoad !isLoaded) {

loadAppConns();

isLoaded = true;

}

}

protected abstract AppConnInfoService createAppConn

protected abstract AppConnResourceService createApp

protected void loadAppConns() {

LOGGER.info("Begin to init all AppConns.");

List? extends AppConnInfo appConnInfos = appCon

if (appConnInfos == null || appConnInfos.isEmpt

if (appConns.isEmpty()) {

throw new DSSRuntimeException("No AppCo

}

LOGGER.warn("No AppConnInfos returned, igno

return;

}

long refreshInterval = AppInstanceConstants.APP

appConnRefreshThread = new AppConnRefreshThread

Utils.defaultScheduler().scheduleAtFixedRate(ap

MapString, AppConn appConns = new HashMap();

ConsumerAppConnInfo loadAndAdd = DSSExceptionUt

AppConn appConn = loadAppConn(appConnInfo);

if (appConn != null) {

appConns.put(appConnInfo.getAppConnName

}

});

appConnInfos.forEach(DSSExceptionUtils.handling

if (appConns.containsKey(appConnInfo.getApp

return;

}

if (StringUtils.isNotBlank(appConnInfo.getR

if (!appConns.containsKey(appConnInfo.g

AppConnInfo referenceAppConnInfo =

.findAny().orElseThrow(() -

" for appConn " + a

loadAndAdd.accept(referenceAppConnI

}

AppConn appConn = loadAppConn(appConnIn

appConns.put(appConnInfo.getAppConnName

} else {

loadAndAdd.accept(appConnInfo);

}

}));

synchronized (this.appConns) {

this.appConns.clear();

this.appConns.putAll(appConns);

appConnList = Collections.unmodifiableList(

}

LOGGER.info("Inited all AppConns, the AppConn l

}

protected AppConn loadAppConn(AppConnInfo appConnIn

LOGGER.info("Ready to load AppConn {}, the appC

String appConnHome = appConnResourceService.get

LOGGER.info("Try to load AppConn {} with home p

AppConn appConn = appConnLoader.getAppConn(appC

appConnInfo.getClassName(), appConnHome

AppDescImpl appDesc = loadAppDesc(appConnInfo);

appConn.setAppDesc(appDesc);

appConn.init();

LOGGER.info("AppConn {} is loaded successfully.

return appConn;

}

protected AppConn loadAppConn(AppConnInfo appConnIn

LOGGER.info("Ready to load AppConn {} with refe

referenceAppConn.getClass().getSimpleNa

if (appConnInfo.getAppConnResource() != null) {

String appConnHome = appConnResourceService

LOGGER.warn("Because AppConn {} is a refere

appConnInfo.getAppConnName(), appCo

}

AppDescImpl appDesc = loadAppDesc(appConnInfo);

AppConn appConn = referenceAppConn.getClass().n

appConn.setAppDesc(appDesc);

appConn.init();

LOGGER.info("AppConn {} is loaded successfully.

return appConn;

}

protected AppDescImpl loadAppDesc(AppConnInfo appCo

List? extends AppInstanceInfo instanceInfos = a

LOGGER.info("The instanceInfos of AppConn {} ar

AppDescImpl appDesc = new AppDescImpl();

for (AppInstanceInfo instanceBean : instanceInf

AppInstanceImpl appInstance = new AppInstan

copyProperties(appConnInfo.getAppConnName()

appDesc.addAppInstance(appInstance);

}

if (appDesc.getAppInstances().isEmpty()) {

LOGGER.warn("The AppConn {} has no appInsta

}

appDesc.setAppName(appConnInfo.getAppConnName()

return appDesc;

}

private void copyProperties(String appConnName, App

appInstance.setBaseUrl(appInstanceInfo.getUrl()

MapString, Object config = new HashMap();

if (StringUtils.isNotEmpty(appInstanceInfo.getE

try {

config = DSSCommonUtils.COMMON\_GSON.fro

} catch (Exception e) {

LOGGER.error("The json of AppConn {} is

throw new DSSRuntimeException("The json

}

}

appInstance.setConfig(config);

if (StringUtils.isNotBlank(appInstanceInfo.getH

appInstance.setHomepageUri(appInstanceInfo.

}

ListDSSLabel labels = Arrays.stream(appInstance

.collect(Collectors.toList());

appInstance.setLabels(labels);

appInstance.setId(appInstanceInfo.getId());

}

private void lazyLoadAppConns() {

if (lazyLoad !isLoaded) {

synchronized (this.appConns) {

if (lazyLoad !isLoaded) {

loadAppConns();

}

}

}

}

@Override

public ListAppConn listAppConns() {

lazyLoadAppConns();

return appConnList;

}

@Override

public AppConn getAppConn(String appConnName) {

lazyLoadAppConns();

return appConns.get(appConnName.toLowerCase());

}

@Override

public void reloadAppConn(String appConnName) {

AppConnInfo appConnInfo = appConnInfoService.ge

if (appConnInfo == null) {

throw new DSSRuntimeException("Cannot get a

}

reloadAppConn(appConnInfo);

}

@Override

public String getAppConnHomePath(String appConnName

AppConnInfo appConnInfo = appConnRefreshThread.

.findAny().orElseThrow(() - new DSSRunt

return appConnResourceService.getAppConnHome(ap

}

public void reloadAppConn(AppConnInfo appConnInfo)

lazyLoadAppConns();

AppConn appConn;

try {

if (StringUtils.isNotBlank(appConnInfo.getR

AppConn referenceAppConn = getAppConn(a

if (referenceAppConn == null) {

throw new DSSRuntimeException("Load

" failed! Caused by: The re

}

appConn = loadAppConn(appConnInfo, refe

} else {

appConn = loadAppConn(appConnInfo);

}

} catch (DSSRuntimeException e) {

throw e;

} catch (Exception e) {

LOGGER.error("Reload AppConn failed.", e);

DSSRuntimeException exception = new DSSRunt

exception.initCause(e);

throw exception;

}

if (appConn == null) {

return;

}

synchronized (this.appConns) {

this.appConns.put(appConnInfo.getAppConnNam

appConnList = Collections.unmodifiableList(

}

LOGGER.info("Reloaded AppConn {}.", appConnInfo

}

}

package com.webank.wedatasphere.dss.appconn.manager.imp

import com.webank.wedatasphere.dss.appconn.manager.serv

import com.webank.wedatasphere.dss.appconn.manager.serv

import com.webank.wedatasphere.dss.common.exception.DSS

import com.webank.wedatasphere.dss.common.exception.DSS

import com.webank.wedatasphere.dss.common.utils.ClassUt

public class AppConnManagerImpl extends AbstractAppConn

@Override

protected AppConnInfoService createAppConnInfoServi

try {

return ClassUtils.getInstance(AppConnInfoSe

} catch (DSSErrorException e) {

throw new DSSRuntimeException(25000, "Canno

}

}

@Override

protected AppConnResourceService createAppConnResou

try {

return ClassUtils.getInstance(AppConnResour

} catch (DSSErrorException e) {

throw new DSSRuntimeException(25000, "Canno

}

}

}

package com.webank.wedatasphere.dss.appconn.manager.imp

import com.webank.wedatasphere.dss.appconn.manager.enti

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.List;

import java.util.Optional;

public class AppConnRefreshThread implements Runnable {

private static final Logger LOGGER = LoggerFactory.

private AbstractAppConnManager appConnManager;

private volatile List? extends AppConnInfo appConnI

public AppConnRefreshThread(AbstractAppConnManager

List? extends AppConnIn

this.appConnManager = appConnManager;

this.appConnInfos = appConnInfos;

}

public List? extends AppConnInfo getAppConnInfos()

return appConnInfos;

}

@Override

public void run() {

LOGGER.info("try to refresh all AppConns.");

List? extends AppConnInfo appConnInfos;

try {

appConnInfos = appConnManager.appConnInfoSe

} catch (Exception e) {

LOGGER.warn("Fetch appConn infos failed, ig

return;

}

if(appConnInfos == null || appConnInfos.isEmpty

LOGGER.warn("no appConnInfos fetched, ignor

return;

}

LOGGER.info("Fetched appConn infos list = {}.",

appConnInfos.forEach(appConnInfo - {

Optional? extends AppConnInfo oldOne = this

if(!oldOne.isPresent() || isChanged(oldOne.

LOGGER.warn("AppConn info {} has update

try {

appConnManager.reloadAppConn(appCon

} catch (Exception e) {

LOGGER.warn("Reload AppConn {} fail

}

}

});

this.appConnInfos = appConnInfos;

LOGGER.info("all AppConns have refreshed.");

}

private boolean isChanged(AppConnInfo one, AppConnI

if(one.getAppConnResource() == null other.getA

return false;

} else if(one.getAppConnResource() == null || o

return true;

} else {

return !one.getAppConnResource().equals(oth

}

}

}

package com.webank.wedatasphere.dss.appconn.manager.ser

import com.webank.wedatasphere.dss.appconn.manager.enti

import com.webank.wedatasphere.dss.appconn.manager.enti

import java.io.Serializable;

import java.util.List;

public interface AppConnInfoService{

List? extends AppConnInfo getAppConnInfos();

AppConnInfo getAppConnInfo(String appConnName);

List? extends AppInstanceInfo getAppInstancesByAppC

List? extends AppInstanceInfo getAppInstancesByAppC

}

package com.webank.wedatasphere.dss.appconn.manager.ser

import com.webank.wedatasphere.dss.appconn.manager.enti

import java.io.Serializable;

public interface AppConnResourceService{

String getAppConnHome(AppConnInfo appConnInfo);

}

package com.webank.wedatasphere.dss.appconn.manager.uti

import com.webank.wedatasphere.dss.appconn.manager.exce

import com.webank.wedatasphere.dss.common.entity.Resour

import java.io.File;

import java.util.Arrays;

import static com.webank.wedatasphere.dss.appconn.manag

public class AppConnIndexFileUtils {

private static boolean isIndexFile(String fileName)

return fileName.startsWith(INDEX\_FILE\_PREFIX)

}

public static File getIndexFile(File parent) {

File[] indexFiles = parent.listFiles((p, fileNa

if(indexFiles == null) {

return null;

}else if(indexFiles.length 1) {

throw new AppConnIndexFileWarnException(205

} else if(indexFiles.length == 0) {

return null;

} else {

return indexFiles[0];

}

}

public static boolean isLatestIndex(File parent, Re

File indexFile = getIndexFile(parent);

if(indexFile == null) {

return false;

}

String version = indexFile.getName().replaceAll

return resource.getVersion().equals(version);

}

public static String getIndexFileName(Resource reso

return INDEX\_FILE\_PREFIX + resource.getVersion(

}

}

package com.webank.wedatasphere.dss.appconn.manager.uti

import com.webank.wedatasphere.dss.appconn.manager.AppC

import org.apache.linkis.common.conf.CommonVars;

import org.apache.linkis.common.conf.TimeType;

import org.apache.linkis.common.utils.Utils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.concurrent.TimeUnit;

public class AppConnManagerUtils {

private static final Logger LOGGER = LoggerFactory.

public static void autoLoadAppConnManager() {

TimeType delay = CommonVars.apply("wds.dss.appc

LOGGER.info("We will try to auto-load all AppCo

Utils.defaultScheduler().schedule(() - {

LOGGER.info("Try to auto-load all AppConns

try {

AppConnManager.getAppConnManager();

} catch (Exception e) {

LOGGER.info("Load AppConns failed.", e)

return;

}

LOGGER.info("All AppConns have loaded succe

}, delay.toLong(), TimeUnit.MILLISECONDS);

}

}

package com.webank.wedatasphere.dss.appconn.manager.uti

import com.webank.wedatasphere.dss.standard.common.desc

import org.apache.commons.lang.StringUtils;

import org.apache.linkis.common.conf.CommonVars;

import org.apache.linkis.common.conf.TimeType;

public class AppInstanceConstants {

static final String INDEX\_FILE\_PREFIX = "index\_";

static final String INDEX\_FILE\_SUFFIX = ".index";

public static final CommonVarsTimeType APP\_CONN\_REF

public static String getHomepageUrl(AppInstance app

Long workspaceI

return getHomepageUrl(appInstance.getBaseUrl(),

workspaceId, workspaceName);

}

public static String getHomepageUrl(String baseUrl,

Long workspaceI

if(StringUtils.isBlank(homepageUri)) {

return baseUrl;

}

if(workspaceId != null homepageUri.contains("$

homepageUri = homepageUri.replace("${worksp

}

if(StringUtils.isNotBlank(workspaceName) homep

homepageUri = homepageUri.replace("${worksp

}

return baseUrl + homepageUri;

} else {

}

}

}

package com.webank.wedatasphere.dss.appconn.scheduler;

import com.webank.wedatasphere.dss.appconn.core.impl.Ab

import com.webank.wedatasphere.dss.orchestrator.convert

import com.webank.wedatasphere.dss.orchestrator.convert

import com.webank.wedatasphere.dss.workflow.conversion.

public abstract class AbstractSchedulerAppConn extends

private ConversionIntegrationStandard conversionInt

protected ConversionIntegrationStandard createConve

return new ProjectConversionIntegrationStandard

}

@Override

protected void initialize() {

conversionIntegrationStandard = createConversio

if(conversionIntegrationStandard instanceof Abs

((AbstractConversionIntegrationStandard) co

}

}

@Override

public ConversionIntegrationStandard getOrCreateCon

return conversionIntegrationStandard;

}

}

package com.webank.wedatasphere.dss.appconn.scheduler;

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.desc

public abstract class AbstractSchedulerStructureIntegra

extends AbstractStructureIntegrationStandard

implements SchedulerStructureIntegrationStandar

protected OrchestrationService createOrchestrationS

return null;

}

@Override

public final OrchestrationService getOrchestrationS

return getOrCreate(appInstance, this::createOrc

}

}

package com.webank.wedatasphere.dss.appconn.scheduler;

import com.webank.wedatasphere.dss.appconn.core.ext.Onl

import com.webank.wedatasphere.dss.appconn.core.ext.Onl

import com.webank.wedatasphere.dss.orchestrator.convert

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.workflow.conversion.

public interface SchedulerAppConn extends OnlySSOAppCon

ConversionIntegrationStandard getOrCreateConversion

@Override

SchedulerStructureIntegrationStandard getOrCreateSt

}

package com.webank.wedatasphere.dss.appconn.scheduler;

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.desc

public interface SchedulerStructureIntegrationStandard

OrchestrationService getOrchestrationService(AppIns

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.exce

public interface OrchestrationCreationOperationR extend

extends StructureOperationR, OrchestrationRespo

OrchestrationResponseRef createOrchestration(R orch

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.enti

import com.webank.wedatasphere.dss.standard.common.exce

public interface OrchestrationDeletionOperationR extend

extends StructureOperationR, ResponseRef {

ResponseRef deleteOrchestration(R requestRef);

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.exce

public interface OrchestrationSearchOperationR extends

extends StructureOperationR, OrchestrationRespo

OrchestrationResponseRef searchOrchestration(R requ

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.standard.app.structu

public abstract class OrchestrationService extends Abst

public final OrchestrationCreationOperation getOrch

return getOrCreate(this::createOrchestrationCre

}

protected abstract OrchestrationCreationOperation c

public final OrchestrationUpdateOperation getOrches

return getOrCreate(this::createOrchestrationUpd

}

protected abstract OrchestrationUpdateOperation cre

public final OrchestrationDeletionOperation getOrch

return getOrCreate(this::createOrchestrationDel

}

protected abstract OrchestrationDeletionOperation c

public final OrchestrationSearchOperation getOrches

return getOrCreate(this::createOrchestrationSea

}

protected abstract OrchestrationSearchOperation cre

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.enti

import com.webank.wedatasphere.dss.standard.common.exce

public interface OrchestrationUpdateOperationR extends

extends StructureOperationR, ResponseRef {

ResponseRef updateOrchestration(R orchestrationRef)

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.orchestrator.common.

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.app.structu

public interface DSSOrchestrationContentRequestRefR ext

extends RefProjectContentRequestRefR, Structure

default DSSOrchestration getDSSOrchestration() {

return (DSSOrchestration) getParameter("dssOrch

}

default R setDSSOrchestration(DSSOrchestration dssO

setParameter("dssOrchestration", dssOrchestrati

return (R) this;

}

class DSSOrchestrationContentRequestRefImpl extends

implements DSSOrchestrationContentRequestRe

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.standard.common.enti

import com.webank.wedatasphere.dss.standard.common.enti

import com.webank.wedatasphere.dss.standard.common.enti

public interface OrchestrationResponseRef extends Respo

Long getRefOrchestrationId();

static ExternalBuilder newExternalBuilder() {

return new ExternalBuilder();

}

class ExternalBuilder extends ResponseRefBuilder.Ex

private Long refOrchestrationId;

public ExternalBuilder setRefOrchestrationId(Lo

this.refOrchestrationId = refOrchestrationI

return this;

}

@Override

protected OrchestrationResponseRef createRespon

return new OrchestrationResponseRefImpl();

}

class OrchestrationResponseRefImpl extends Resp

public OrchestrationResponseRefImpl() {

super(ExternalBuilder.this.responseBody

ExternalBuilder.this.errorMsg,

}

@Override

public Long getRefOrchestrationId() {

return refOrchestrationId;

}

}

}

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.orchestrator.common.

import com.webank.wedatasphere.dss.standard.app.structu

public interface OrchestrationUpdateRequestRefR extends

extends DSSOrchestrationContentRequestRefR, Ref

@Override

default String getOrchestrationName() {

return getDSSOrchestration().getName();

}

@Override

default R setOrchestrationName(String orchestration

if(getDSSOrchestration() instanceof DSSOrchestr

((DSSOrchestratorInfo) getDSSOrchestration(

}

return (R) this;

}

class OrchestrationUpdateRequestRefImpl extends Str

implements OrchestrationUpdateRequestRefOrc

}

package com.webank.wedatasphere.dss.appconn.scheduler.s

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.app.structu

public interface RefOrchestrationContentRequestRefR ext

extends RefProjectContentRequestRefR, Structure

default Long getRefOrchestrationId() {

return (Long) getParameter("refOrchestrationId"

}

default R setRefOrchestrationId(Long refOrchestrati

setParameter("refOrchestrationId", refOrchestra

return (R) this;

}

default String getOrchestrationName() {

return (String) getParameter("orchestrationName

}

default R setOrchestrationName(String orchestration

setParameter("orchestrationName", orchestration

return (R) this;

}

class RefOrchestrationContentRequestRefImpl extends

implements RefOrchestrationContentRequestRe

}

package com.webank.wedatasphere.dss.appconn.scheduler.u

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.appconn.scheduler.st

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.app.structu

import com.webank.wedatasphere.dss.standard.common.enti

import java.util.function.BiFunction;

import java.util.function.Consumer;

import java.util.function.Function;

import java.util.function.Supplier;

import static com.webank.wedatasphere.dss.standard.app.

public class OrchestrationOperationUtils {

public static K extends StructureRequestRef, V exte

return tryStructureOperation(getOrchestrationSe

if(dssOrchestrationContentRequestRefConsume

dssOrchestrationContentRequestRefConsum

}

if(refOrchestrationContentRequestRefConsume

refOrchestrationContentRequestRefConsum

}

}, responseRefConsumer, errorMsg);

}

}

package com.webank.wedatasphere.dss.appconn.scheduler.u

import org.apache.linkis.common.conf.CommonVars;

public class SchedulerConf {

public static final CommonVarsString JOB\_LABEL = Co

}

package org.apache.linkis.manager.engineplugin.appconn.

import com.webank.wedatasphere.dss.common.label.DSSLabe

import com.webank.wedatasphere.dss.standard.app.develop

import com.webank.wedatasphere.dss.standard.app.develop

import com.webank.wedatasphere.dss.standard.app.develop

import com.webank.wedatasphere.dss.standard.app.develop

import com.webank.wedatasphere.dss.standard.app.sso.Wor

import com.webank.wedatasphere.dss.standard.common.enti

import java.util.List;

import java.util.Map;

public class AppConnExecutionUtils {

public static ResponseRef tryToOperation(RefExecuti

ExecutionRequestR

String name, Stri

MapString, Object

return DevelopmentOperationUtils.tryRefJobConte

developmentService - refExecutionServic

refJobContentRequestRef - refJobContent

dssContextRequestRef - dssContextReques

projectRefRequestRef - projectRefReques

(developmentOperation, developmentReque

RefExecutionRequestRef requestRef =

requestRef.setExecutionRequestRefCo

.setVariables(variables)

.setDSSLabels(dssLabels)

.setName(name)

.setType(type)

.setUserName(userName)

.setWorkspace(workspace);

return ((RefExecutionOperation) dev

}, "execute node " + name + " with type " + typ

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

public class ApiCommentUpdateRequest {

private Long id;

private String comment;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getComment() {

return comment;

}

public void setComment(String comment) {

this.comment = comment;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

import org.apache.linkis.ujes.client.response.JobExecut

public class ApiServiceJob {

private String submitUser;

private String proxyUser;

private JobExecuteResult jobExecuteResult;

public String getSubmitUser() {

return submitUser;

}

public void setSubmitUser(String submitUser) {

this.submitUser = submitUser;

}

public String getProxyUser() {

return proxyUser;

}

public void setProxyUser(String proxyUser) {

this.proxyUser = proxyUser;

}

public JobExecuteResult getJobExecuteResult() {

return jobExecuteResult;

}

public void setJobExecuteResult(JobExecuteResult jo

this.jobExecuteResult = jobExecuteResult;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

public class ApiServiceQuery {

private String name;

private String tag;

private Integer status;

private String creator;

private int currentPage;

private int pageSize;

public int getWorkspaceId() {

return workspaceId;

}

public void setWorkspaceId(int workspaceId) {

this.workspaceId = workspaceId;

}

private int workspaceId;

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

private String userName;

public ApiServiceQuery(String userName,String name,

this.userName= userName;

this.name = name;

this.tag = tag;

this.status = status;

this.creator = creator;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getTag() {

return tag;

}

public void setTag(String tag) {

this.tag = tag;

}

public Integer getStatus() {

return status;

}

public void setStatus(Integer status) {

this.status = status;

}

public String getCreator() {

return creator;

}

public void setCreator(String creator) {

this.creator = creator;

}

public int getCurrentPage() {

return currentPage;

}

public void setCurrentPage(int currentPage) {

this.currentPage = currentPage;

}

public int getPageSize() {

return pageSize;

}

public void setPageSize(int pageSize) {

this.pageSize = pageSize;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

import java.util.Date;

public class ApiServiceToken {

private String applyUser;

private Date applyTime;

private Long apiServiceId;

private String publisher;

public String getPublisher() {

return publisher;

}

public void setPublisher(String publisher) {

this.publisher = publisher;

}

public Long getApiServiceId() {

return apiServiceId;

}

public void setApiServiceId(Long apiServiceId) {

this.apiServiceId = apiServiceId;

}

public Date getApplyTime() {

return applyTime;

}

public void setApplyTime(Date applyTime) {

this.applyTime = applyTime;

}

public String getApplyUser() {

return applyUser;

}

public void setApplyUser(String applyUser) {

this.applyUser = applyUser;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

import java.util.HashMap;

import java.util.List;

public class DirFileTree {

private String name;

private String path;

private HashMapString,String properties;

private ListDirFileTree children;

private Boolean isLeaf = false;

private String parentPath;

public Boolean getIsLeaf() {

return isLeaf;

}

public void setIsLeaf(Boolean isLeaf) {

this.isLeaf = isLeaf;

}

public String getParentPath() {

return parentPath;

}

public void setParentPath(String parentPath) {

this.parentPath = parentPath;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPath() {

return path;

}

public void setPath(String path) {

this.path = path;

}

public HashMapString, String getProperties() {

return properties;

}

public void setProperties(HashMapString, String pro

this.properties = properties;

}

public ListDirFileTree getChildren() {

return children;

}

public void setChildren(ListDirFileTree children) {

this.children = children;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

public class LinkisExecuteResult {

public LinkisExecuteResult(String taskId,String exe

this.taskId = taskId;

this.execId = execId;

}

public String getTaskId() {

return taskId;

}

public void setTaskId(String taskId) {

this.taskId = taskId;

}

private String taskId;

public String getExecId() {

return execId;

}

public void setExecId(String execId) {

this.execId = execId;

}

private String execId;

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

import java.util.Map;

public class QueryRequest {

private String moduleName;

private MapString, Object params;

public MapString, Object getParams() {

return params;

}

public void setParams(MapString, Object params) {

this.params = params;

}

public String getModuleName() {

return moduleName;

}

public void setModuleName(String moduleName) {

this.moduleName = moduleName;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

import java.util.Date;

public class TokenQuery {

private Long apiId;

private String user;

private Integer status;

private String startDate;

private String endDate;

private int currentPage;

private int pageSize;

public String getCreator() {

return creator;

}

public void setCreator(String creator) {

this.creator = creator;

}

private String creator;

public TokenQuery(Long apiId, String user, Integer

this.apiId = apiId;

this.user = user;

this.status = status;

}

public TokenQuery(Long apiId, String user, Integer

this.apiId = apiId;

this.user = user;

this.status = status;

this.startDate = startDate;

this.endDate = endDate;

}

public TokenQuery() {

}

public String getStartDate() {

return startDate;

}

public void setStartDate(String startDate) {

this.startDate = startDate;

}

public String getEndDate() {

return endDate;

}

public void setEndDate(String endDate) {

this.endDate = endDate;

}

public int getCurrentPage() {

return currentPage;

}

public void setCurrentPage(int currentPage) {

this.currentPage = currentPage;

}

public int getPageSize() {

return pageSize;

}

public void setPageSize(int pageSize) {

this.pageSize = pageSize;

}

public Long getApiId() {

return apiId;

}

public void setApiId(Long apiId) {

this.apiId = apiId;

}

public String getUser() {

return user;

}

public void setUser(String user) {

this.user = user;

}

public Integer getStatus() {

return status;

}

public void setStatus(Integer status) {

this.status = status;

}

}

package com.webank.wedatasphere.dss.apiservice.core.bo;

public class VariableString {

private String path;

public VariableString(String unparsedPath) {

this.path = unparsedPath;

}

public String getPath() {

return path;

}

public void setPath(String path) {

this.path = path;

}

}

package com.webank.wedatasphere.dss.apiservice.core.con

import org.apache.linkis.common.conf.CommonVars;

public class ApiServiceConfiguration {

public final static CommonVarsString LINKIS\_AUTHOR\_

public final static CommonVarsString LINKIS\_ADMIN\_U

public final static CommonVarsInteger LINKIS\_CONN

public final static CommonVarsString LINKIS\_API\_VER

public final static CommonVarsString API\_SERVICE\_TO

public final static CommonVarsString DSS\_API\_TOKEN\_

public final static CommonVarsInteger LINKIS\_JOB\_

public final static CommonVarsInteger LOG\_ARRAY\_LEN

public final static CommonVarsInteger RESULT\_PRINT\_

}

package com.webank.wedatasphere.dss.apiservice.core.con

public class ApiCommonConstant {

public static final String DEFAULT\_APPROVAL\_NO="000

public static final Integer API\_DELETE\_STATUS=2;

}

package com.webank.wedatasphere.dss.apiservice.core.con

public enum ApprovalStatus {

INIT("init", 0),

APPLYING("apply", 1),

SUCCESS("success", 2),

FAIELD("failed", 3);

String key;

int value;

ApprovalStatus(String key, int value) {

this.key = key;

this.value = value;

}

public String getKey() {

return key;

}

public void setKey(String key) {

this.key = key;

}

public int getValue() {

return value;

}

public void setValue(int value) {

this.value = value;

}

}

package com.webank.wedatasphere.dss.apiservice.core.con

public class ParamType {

public static final String number = "2";

public static final String array = "4";

}

package com.webank.wedatasphere.dss.apiservice.core.con

public enum ParamTypeEnum {

STRING(1, "String"),

NUMBER(2, "Number"),

DATE(3, "Date"),

ARRAY(4, "Array");

private Integer index;

private String name;

ParamTypeEnum(Integer index, String name) {

this.index = index;

this.name = name;

}

public static ParamTypeEnum getEnum(Integer index)

if (index == null) {

return STRING;

}

for (ParamTypeEnum statusEnum : values()) {

if (statusEnum.getIndex().equals(index)) {

return statusEnum;

}

}

return STRING;

}

public Integer getIndex() {

return this.index;

}

public String getName() {

return this.name;

}

}

package com.webank.wedatasphere.dss.apiservice.core.con

public enum RequireEnum {

YES(1, "是"),

NO(0, "否");

private Integer index;

private String name;

RequireEnum(Integer index, String name) {

this.index = index;

this.name = name;

}

public static RequireEnum getEnum(Integer index) {

if (index == null) {

return NO;

}

for (RequireEnum statusEnum : values()) {

if (statusEnum.getIndex().equals(index)) {

return statusEnum;

}

}

return NO;

}

public Integer getIndex() {

return this.index;

}

public String getName() {

return this.name;

}

}

package com.webank.wedatasphere.dss.apiservice.core.con

public enum SaveTokenEnum {

SUCCESS(1, "success"), FAILED(2, "failed"), DUPLICA

int index;

String value;

SaveTokenEnum(int index, String value) {

this.index = index;

this.value = value;

}

public int getIndex() {

return index;

}

public void setIndex(int index) {

this.index = index;

}

public String getValue() {

return value;

}

public void setValue(String value) {

this.value = value;

}

}

package com.webank.wedatasphere.dss.apiservice.core.con

public enum SQLMetadataInfoCheckStatus {

LEGAL(0, "该SQL查询合法"), ILLEGALSQL(1,"不支持无库表信息的SQL查询"

int index;

String desc;

SQLMetadataInfoCheckStatus(int index, String desc)

this.index = index;

this.desc = desc;

}

public String getDescByIndex(int index) {

for (SQLMetadataInfoCheckStatus item : values()

if (item.getIndex() == index) {

return item.getDesc();

}

}

return null;

}

public int getIndex() {

return index;

}

public String getDesc() {

return desc;

}

}

package com.webank.wedatasphere.dss.apiservice.core.con

public enum StatusEnum {

DISABLE(0, "停止"),

ENABLE(1, "运行中");

private Integer index;

private String name;

StatusEnum(Integer index, String name) {

this.index = index;

this.name = name;

}

public static StatusEnum getEnum(Integer index) {

if (index == null) {

return null;

}

for (StatusEnum statusEnum : values()) {

if (statusEnum.getIndex().equals(index)) {

return statusEnum;

}

}

return null;

}

public Integer getIndex() {

return this.index;

}

public String getName() {

return this.name;

}

}

package com.webank.wedatasphere.dss.apiservice.core.dao

import com.webank.wedatasphere.dss.apiservice.core.vo.A

public interface ApiServiceAccessDao {

void addAccessRecord(ApiAccessVo apiAccessVo);

ApiAccessVo queryByVersionId(Long versionId);

ApiAccessVo queryByApiId(Long apiServiceId);

}

package com.webank.wedatasphere.dss.apiservice.core.dao

import com.webank.wedatasphere.dss.apiservice.core.vo.A

import org.apache.ibatis.annotations.Param;

public interface ApiServiceApprovalDao {

void insert(ApprovalVo approvalVo);

ApprovalVo queryByApprovalNo(@Param("approvalNo") S

void setApprovalStatusSuccess(@Param("approvalNo")

void setApprovalStatusFailed(@Param("approvalNo") S

void setApprovalStatusInit(@Param("approvalNo") Str

void setApprovalStatusApplying(@Param("approvalNo")

void updateApprovalStatus(@Param("approvalNo") Stri

void deleteApproval(@Param("approvalNo") String app

ApprovalVo queryByVersionId(@Param("apiVersionId")

}

package com.webank.wedatasphere.dss.apiservice.core.dao

import com.webank.wedatasphere.dss.apiservice.core.bo.A

import com.webank.wedatasphere.dss.apiservice.core.vo.A

import org.apache.ibatis.annotations.Param;

import java.util.List;

public interface ApiServiceDao {

void insert(ApiServiceVo apiServiceVo);

void update(ApiServiceVo apiServiceVo);

void updateToTarget(ApiServiceVo apiServiceVo);

ListApiServiceVo query(ApiServiceQuery query);

ListApiServiceVo queryByScriptPath(@Param("scriptPa

ApiServiceVo queryByPath(String path);

Integer queryCountByPath(@Param("scriptPath") Strin

Integer queryCountByName(String name);

Integer enableApi(@Param("id") Long id);

Integer disableApi(@Param("id") Long id);

Integer deleteApi(@Param("id") Long id);

List String queryAllTags(@Param("userName")String u

ApiServiceVo queryById(@Param("id") Long id);

ListApiServiceVo queryByWorkspaceId(@Param("workspa

Integer updateApiServiceComment(@Param("id") Long i

}

package com.webank.wedatasphere.dss.apiservice.core.dao

import com.webank.wedatasphere.dss.apiservice.core.vo.P

import org.apache.ibatis.annotations.Param;

import java.util.List;

public interface ApiServiceParamDao {

int insert(ParamVo param);

void update(ParamVo param);

ListParamVo queryByVersionId(@Param("apiVersionId")

int queryById(@Param("paramId") Long paramId);

}

package com.webank.wedatasphere.dss.apiservice.core.dao

import com.webank.wedatasphere.dss.apiservice.core.vo.T

import com.webank.wedatasphere.dss.apiservice.core.bo.T

import org.apache.ibatis.annotations.Param;

import java.util.List;

public interface ApiServiceTokenManagerDao {

void insert(TokenManagerVo tokenManagerVo);

void insertList(ListTokenManagerVo tokenManagerVos)

ListTokenManagerVo queryByApplyUser(String userName

ListTokenManagerVo queryByApplyUserAndServiceId(@Pa

ListTokenManagerVo queryByApplyUserAndVersionId(@Pa

ListTokenManagerVo query(TokenQuery tokenQuery);

TokenManagerVo queryTokenById(@Param("id") Long tok

ListTokenManagerVo queryTokenByApiServiceId(@Param(

int queryApprovalNo(@Param("approvalNo") String app

ListTokenManagerVo queryTokenByStatus(@Param("statu

void disableTokenStatus(@Param("id") Long id);

void disableTokenStatusByVersionId(@Param("apiVersi

void disableTokenStatusByApiId(@Param("apiId") Long

void enableTokenStatus(@Param("id") Long id);

void enableTokenStatusByVersionId(@Param("apiVersio

void enableTokenStatusByApiId(@Param("apiId") Long

void deleteTokenById(@Param("id") Long tokenId);

}

package com.webank.wedatasphere.dss.apiservice.core.dao

import com.webank.wedatasphere.dss.apiservice.core.vo.A

import org.apache.ibatis.annotations.Param;

import java.util.List;

public interface ApiServiceVersionDao {

void insert(ApiVersionVo apiVersionVo);

ApiVersionVo queryApiVersionByVersionId(@Param("id"

ListApiVersionVo queryApiVersionByApiServiceId(@Par

void updateApiVersionStatusById(@Param("id") Long a

void updateApiVersionStatus(ApiVersionVo apiVersion

void updateAllApiVersionStatusByApiServiceId(@Param

void disableApiVersionStatusByApiId(@Param("api\_id"

void deleteApiVersionById(@Param("id") Long apiVers

}

package com.webank.wedatasphere.dss.apiservice.core.exc

import org.apache.linkis.common.exception.ErrorExceptio

public class ApiExecuteException extends ErrorException

public ApiExecuteException(int errCode, String

super(errCode, desc);

}

public ApiExecuteException(int errCode, String

super(errCode, desc, ip, port, serviceKind)

}

}

package com.webank.wedatasphere.dss.apiservice.core.exc

import org.apache.linkis.common.exception.ErrorExceptio

public class ApiServiceQueryException extends ErrorExce

public ApiServiceQueryException(int errCode, String

super(errCode, desc);

}

public ApiServiceQueryException(int errCode, String

super(errCode, desc, ip, port, serviceKind);

}

}

package com.webank.wedatasphere.dss.apiservice.core.exc

public class ApiServiceRuntimeException extends Runtime

public ApiServiceRuntimeException(String message) {

super(message);

}

public ApiServiceRuntimeException(String message, T

super(message, cause);

}

}

package com.webank.wedatasphere.dss.apiservice.core.exc

import org.apache.linkis.common.exception.ErrorExceptio

public class ApiServiceTokenException extends ErrorExce

public ApiServiceTokenException(int errCode, St

super(errCode, desc);

}

public ApiServiceTokenException(int errCode, St

super(errCode, desc, ip, port, serviceKind)

}

}

package com.webank.wedatasphere.dss.apiservice.core.exc

public class AssertException extends ApiServiceRuntimeE

public AssertException(String message) {

super(message);

}

}

package com.webank.wedatasphere.dss.apiservice.core.exc

import com.webank.wedatasphere.dss.common.utils.Message

import org.apache.linkis.server.Message;

import javax.validation.ConstraintViolationException;

import javax.validation.ValidationException;

import javax.ws.rs.core.Response;

import javax.ws.rs.ext.ExceptionMapper;

import javax.ws.rs.ext.Provider;

@Provider

public class BeanValidationExceptionMapper implements E

@Override

public Response toResponse(ValidationException exce

StringBuilder stringBuilder = new StringBuilder

((ConstraintViolationException)exception)

.getConstraintViolations().forEach(cons

Message message = Message.error("Bean validatio

return MessageUtils.messageToResponse(message);

}

}

package com.webank.wedatasphere.dss.apiservice.core.exe

import org.apache.linkis.ujes.client.response.JobExecut

import java.util.Map;

public interface ApiServiceExecuteJob {

String getCode();

void setCode(String code);

String getEngineType();

void setEngineType(String engineType);

String getRunType();

void setRunType(String runType);

String getUser();

void setUser(String user);

String getJobName();

MapString, Object getParams();

void setParams(MapString, Object params);

MapString,Object getRuntimeParams();

void setRuntimeParams(MapString, Object runtime

JobExecuteResult getJobExecuteResult();

void setJobExecuteResult(JobExecuteResult jobEx

MapString, String getJobProps();

void setJobProps(MapString, String jobProps);

String getScriptPath();

void setScriptePath(String path);

}

package com.webank.wedatasphere.dss.apiservice.core.exe

import org.apache.linkis.ujes.client.request.JobExecute

import org.apache.linkis.ujes.client.response.JobExecut

import java.util.Map;

public class DefaultApiServiceJob implements ApiService

private MapString, String jobProps;

private String user;

private MapString, String source;

private MapString, Object variables;

private MapString, Object configuration;

private String code;

private String engineType;

private String runType;

private MapString, Object params;

private MapString, Object runtimeParams;

private JobExecuteResult jobExecuteResult;

private String scriptPath;

@Override

public String getCode() {

return this.code;

}

@Override

public void setCode(String code) {

this.code = code;

}

@Override

public String getEngineType() {

return engineType;

}

@Override

public void setEngineType(String engineType) {

this.engineType = engineType;

}

@Override

public String getRunType() {

return runType;

}

@Override

public void setRunType(String runType) {

this.runType = runType;

}

@Override

public String getUser() {

return user;

}

@Override

public void setUser(String user){

this.user =user;

}

@Override

public String getJobName() {

return null;

}

@Override

public MapString, Object getParams() {

return params;

}

@Override

public void setParams(MapString, Object params) {

this.params = params;

}

@Override

public MapString, Object getRuntimeParams() {

return runtimeParams;

}

@Override

public void setRuntimeParams(MapString, Object runt

this.runtimeParams = runtimeParams;

}

@Override

public JobExecuteResult getJobExecuteResult() {

return this.jobExecuteResult;

}

@Override

public void setJobExecuteResult(JobExecuteResult jo

this.jobExecuteResult = jobExecuteResult;

}

@Override

public MapString, String getJobProps() {

return jobProps;

}

@Override

public void setJobProps(MapString, String jobProps)

this.jobProps = jobProps;

}

@Override

public String getScriptPath() {

return scriptPath;

}

@Override

public void setScriptePath(String path) {

this.scriptPath = path;

}

}

package com.webank.wedatasphere.dss.apiservice.core.exe

import com.webank.wedatasphere.dss.apiservice.core.acti

import com.webank.wedatasphere.dss.apiservice.core.acti

import com.webank.wedatasphere.dss.apiservice.core.acti

import com.webank.wedatasphere.dss.apiservice.core.conf

import com.webank.wedatasphere.dss.apiservice.core.exce

import org.apache.commons.lang.StringUtils;

import org.apache.linkis.common.utils.Utils;

import org.apache.linkis.ujes.client.UJESClient;

import org.apache.linkis.ujes.client.request.ResultSetA

import org.apache.linkis.ujes.client.request.ResultSetL

import org.apache.linkis.ujes.client.response.JobExecut

import org.apache.linkis.ujes.client.response.JobInfoRe

import org.apache.linkis.ujes.client.response.JobLogRes

import org.apache.linkis.ujes.client.response.ResultSet

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

public class ExecuteCodeHelper {

private static final Logger LOGGER = LoggerFactory.

private static final String RELEASE\_SCALA\_TEMPLATE

"val sql = %s\n" +

"val df = sqlContext.sql(sql)\n" +

"show(df)\n";

private static final String SCALA\_MARK = "\"\"\"";

private static final String EXECUTE\_SCALA\_TEMPLATE

"val executeCode = %s\n" +

"val df = sqlContext.sql(executeCode)\n" +

"show(df)\n";

public static String packageCodeToRelease(String ex

String retStr = String.format(RELEASE\_SCALA\_TEM

LOGGER.info("release scala code is {}", retStr)

return retStr;

}

public static String packageCodeToExecute(String ex

String retStr = String.format(EXECUTE\_SCALA\_TEM

LOGGER.info("execute scala code is {}", retStr)

return retStr;

}

public static MapString,Object getMetaDataInfoByEx

String exe

MapString,

String scr

MapString,Object resultMap = new HashMap();

UJESClient client = LinkisJobSubmit.getClient()

ApiServiceExecuteJob job = new DefaultApiServic

job.setCode(ExecuteCodeHelper.packageCodeToRele

job.setEngineType("spark");

job.setRunType("scala");

job.setUser(user);

job.setParams(null);

job.setRuntimeParams((MapString,Object) params.

job.setScriptePath(scriptPath);

JobExecuteResult jobExecuteResult = LinkisJobSu

job.setJobExecuteResult(jobExecuteResult);

try {

waitForComplete(job, client);

} catch (Exception e) {

LOGGER.warn("Failed to execute job", e);

String reason = getLog(job, client);

LOGGER.error("Reason for failure: " + reaso

throw new ApiExecuteException(800024,"获取库表信

}

int resultSize = getResultSize(job, client);

for(int i =0; i resultSize; i++){

String result = getResult(job, i, ApiServic

LOGGER.info("The content of the " + (i + 1)

+ result);

resultMap.put(Integer.toString(i),result);

}

LOGGER.info("Finished to execute job");

return resultMap;

}

public static void waitForComplete(ApiServiceExecu

JobInfoResult jobInfo = client.getJobInfo(job.g

while (!jobInfo.isCompleted()) {

LOGGER.info("Update Progress info:" + getPr

LOGGER.info("----linkis log ----");

Utils.sleepQuietly(ApiServiceConfiguration.

jobInfo = client.getJobInfo(job.getJobExecu

}

if (!jobInfo.isSucceed()) {

throw new ApiExecuteException(90101, "Faile

}

}

public static void cancel(ApiServiceExecuteJob job,

client.kill(job.getJobExecuteResult());

}

public static double getProgress(ApiServiceExecute

return client.progress(job.getJobExecuteResult(

}

public static Boolean isCompleted(ApiServiceExecut

return client.getJobInfo(job.getJobExecuteResul

}

public static String getResult(ApiServiceExecuteJo

String resultContent = null;

JobInfoResult jobInfo = client.getJobInfo(job.g

String[] resultSetList = jobInfo.getResultSetLi

if (resultSetList != null resultSetList.length

Object fileContent = client.resultSet(Resul

.setPath(resultSetList[index])

.setUser(job.getJobExecuteResult().

.setPageSize(maxSize).build()).getF

if (fileContent instanceof ArrayList) {

ArrayListArrayListString resultSetRow =

resultContent = StringUtils.join(result

} else {

resultContent = fileContent.toString();

}

}

return resultContent;

}

public static int getResultSize(ApiServiceExecuteJ

JobInfoResult jobInfo = client.getJobInfo(job.g

if (jobInfo.isSucceed()) {

String[] resultSetList = jobInfo.getResultS

if (resultSetList != null resultSetList.le

return resultSetList.length;

}

}

return 0;

}

public static String getLog(ApiServiceExecuteJob jo

JobLogResult jobLogResult = client

.log(job.getJobExecuteResult(),

0,

50);

ArrayListString logArray = jobLogResult.getLog(

if (logArray != null logArray.size()

= ApiServiceConfiguration.LOG\_ARRAY\_LEN

StringUtils.isNotEmpty(logArray.get(3)

return logArray.get(3);

}

return null;

}

public static String getResultList(JobExecuteResul

ResultSetListResult resultList = (ResultSetList

.setUser(executeResult.getUser()).setPa

return resultList.getResponseBody();

}

public static String getResultContent(String user,

return client.resultSet(ResultSetAction.builder

.setPath(path)

.setUser(user)

.setPageSize(maxSize).build()).getR

}

public static InputStream downloadResultSet(String

String

String

String

String

String

String

String

UJESCli

ResultSetDownloadAction resultSetDownloadAction

resultSetDownloadAction.setUser(user);

resultSetDownloadAction.setParameter("path",pat

resultSetDownloadAction.setParameter("charset",

resultSetDownloadAction.setParameter("outputFil

resultSetDownloadAction.setParameter("csvSepera

resultSetDownloadAction.setParameter("outputFil

resultSetDownloadAction.setParameter("sheetName

resultSetDownloadAction.setParameter("nullValue

client.executeUJESJob(resultSetDownloadAction);

return resultSetDownloadAction.getInputStream()

}

public static MapString, Object getTaskInfoById(Jo

return (MapString, Object) client.getJobInfo(jo

}

public static String getUserWorkspaceIds(String use

ApiServiceGetAction apiServiceGetAction = new A

apiServiceGetAction.setUser(userName);

apiServiceGetAction.setParameter("userName",use

ResultWorkspaceIds userWorkspaceIds = (ResultWo

return userWorkspaceIds.getUserWorkspaceIds();

}

}

package com.webank.wedatasphere.dss.apiservice.core.exe

import com.webank.wedatasphere.dss.apiservice.core.conf

import org.apache.linkis.common.conf.Configuration;

import org.apache.linkis.httpclient.dws.authentication.

import org.apache.linkis.httpclient.dws.config.DWSClien

import org.apache.linkis.httpclient.dws.config.DWSClien

import org.apache.linkis.ujes.client.UJESClient;

import org.apache.linkis.ujes.client.UJESClientImpl;

import org.apache.linkis.ujes.client.request.JobExecute

import org.apache.linkis.ujes.client.response.JobExecut

import java.util.HashMap;

import java.util.Map;

import java.util.concurrent.TimeUnit;

public class LinkisJobSubmit {

private static final MapString, UJESClient ujesClie

public static UJESClient getClient() {

return getClient(new HashMap(0));

}

public static UJESClient getClient(MapString, Strin

return getUJESClient(

Configuration.GATEWAY\_URL().getValue(pa

ApiServiceConfiguration.LINKIS\_ADMIN\_US

ApiServiceConfiguration.LINKIS\_AUTHOR\_U

params);

}

public static UJESClient getUJESClient(String url,

String key = url + user + token;

if(ujesClientMap.containsKey(key)) {

return ujesClientMap.get(key);

}

synchronized (LinkisJobSubmit.class) {

if(!ujesClientMap.containsKey(key)) {

ujesClientMap.put(key, new UJESClientIm

}

}

return ujesClientMap.get(key);

}

public static DWSClientConfig getClientConfig(Strin

return ((DWSClientConfigBuilder) (DWSClientConf

.addServerUrl(url)

.connectionTimeout(ApiServiceConfigurat

.discoveryEnabled(false).discoveryFrequ

.loadbalancerEnabled(true)

.maxConnectionSize(5)

.retryEnabled(false).readTimeout(ApiSer

.setAuthenticationStrategy(new TokenAut

.setAuthTokenKey(user).setAuthTokenValu

.setDWSVersion(ApiServiceConfiguration.

}

public static JobExecuteResult execute(ApiServiceEx

MapString, Object source = new HashMap();

source.put("DSS-ApiService",job.getScriptPath()

JobExecuteAction.Builder builder = JobExecuteAc

.addExecuteCode(job.getCode())

.setEngineTypeStr(job.getEngineType())

.setRunTypeStr(job.getRunType())

.setUser(job.getUser())

.setParams(job.getParams())

.setVariableMap(job.getRuntimeParams())

.setSource(source);

JobExecuteAction jobAction = builder.build();

JobExecuteResult res = client.execute(jobAction

return res;

}

public static JobExecuteResult execute(ApiServiceEx

MapString, Object source = new HashMap();

source.put("DSS-ApiService",job.getScriptPath()

JobExecuteAction.Builder builder = JobExecuteAc

.addExecuteCode(job.getCode())

.setEngineTypeStr(job.getEngineType())

.setRunTypeStr(job.getRunType())

.setUser(job.getUser())

.setParams(job.getParams())

.setVariableMap(job.getRuntimeParams())

.setSource(source);

JobExecuteAction jobAction = builder.build();

JobExecuteResult res = client.execute(jobAction

return res;

}

}

package com.webank.wedatasphere.dss.apiservice.core.jdb

import com.zaxxer.hikari.HikariConfig;

import com.zaxxer.hikari.HikariDataSource;

import org.springframework.context.annotation.Configura

import org.springframework.jdbc.core.namedparam.NamedPa

import javax.sql.DataSource;

import java.util.HashMap;

import java.util.Map;

@Configuration

public class DatasourceService {

private static final MapString, NamedParameterJdbcT

private DataSource getDatasource(String url, String

HikariConfig hikariConfig = new HikariConfig();

hikariConfig.setJdbcUrl(url);

hikariConfig.setUsername(username);

hikariConfig.setPassword(password);

hikariConfig.setMaximumPoolSize(100);

hikariConfig.setMinimumIdle(10);

return new HikariDataSource(hikariConfig);

}

public NamedParameterJdbcTemplate getNamedParameter

String key = genKey(url, username);

NamedParameterJdbcTemplate jdbcTemplate = CLIEN

if (jdbcTemplate == null) {

synchronized (this) {

jdbcTemplate = CLIENT\_MAP.get(key);

if (jdbcTemplate == null) {

jdbcTemplate = new NamedParameterJd

CLIENT\_MAP.put(key, jdbcTemplate);

}

}

}

return jdbcTemplate;

}

private String genKey(String jdbcUrl, String userna

return String.join("-", jdbcUrl, username);

}

}

package com.webank.wedatasphere.dss.apiservice.core.res

import com.webank.wedatasphere.dss.apiservice.core.bo.A

import com.webank.wedatasphere.dss.apiservice.core.bo.A

import com.webank.wedatasphere.dss.apiservice.core.serv

import com.webank.wedatasphere.dss.apiservice.core.serv

import com.webank.wedatasphere.dss.apiservice.core.util

import com.webank.wedatasphere.dss.apiservice.core.util

import com.webank.wedatasphere.dss.apiservice.core.vo.A

import com.webank.wedatasphere.dss.apiservice.core.vo.A

import com.webank.wedatasphere.dss.apiservice.core.vo.A

import com.webank.wedatasphere.dss.apiservice.core.vo.Q

import org.apache.commons.collections.MapUtils;

import org.apache.commons.lang.StringUtils;

import org.apache.linkis.server.Message;

import org.apache.linkis.server.security.SecurityFilter

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Aut

import javax.servlet.http.HttpServletRequest;

import javax.validation.ConstraintViolation;

import javax.validation.ConstraintViolationException;

import javax.validation.Validator;

import javax.validation.groups.Default;

import java.util.Calendar;

import java.util.List;

import java.util.Map;

import java.util.Set;

import java.util.regex.Pattern;

import java.util.stream.Collectors;

@RestController

public class ApiServiceCoreRestfulApi {

private static final Logger LOG = LoggerFactory.get

@Autowired

private ApiService apiService;

@Autowired

private ApiServiceQueryService apiServiceQueryServi

@Autowired

private Validator beanValidator;

private static final Pattern WRITABLE\_PATTERN = Pat

public Message insert(@RequestBody ApiServiceVo api

return ApiUtils.doAndResponse(() - {

if (apiService.getWorkspaceId() == null){

apiService.setWorkspaceId(180L);

}

if (StringUtils.isBlank(apiService.getAlias

return Message.error("'api service alia

}

if (StringUtils.isBlank(apiService.getScrip

return Message.error("'api service scri

}

if (StringUtils.isBlank(apiService.getConte

return Message.error("'api service

}

if (null == apiService.getWorkspaceId()) {

return Message.error("'api service work

}

if (apiService.getContent().contains(";"))

if(!apiService.getContent().toLowerCase

return Message.error("'api service

}

}

if (WRITABLE\_PATTERN.matcher(apiService.get

return Message.error("'api service scri

}

MapString, Object metadata = apiService.get

if (apiService.getScriptPath().endsWith(".j

if (MapUtils.isEmpty(metadata)) {

return Message.error("'api service

}

MapString, Object configuration = (MapS

if (MapUtils.isEmpty(configuration)) {

return Message.error("'api service

}

MapString, Object datasource = (MapStri

if (MapUtils.isEmpty(datasource)) {

return Message.error("'api service

}

}

String userName = SecurityFilter.getLoginUs

SetConstraintViolationApiServiceVo result =

if (result.size() 0) {

throw new ConstraintViolationException(

}

ApprovalVo approvalVo = apiService.getAppro

if (StringUtils.isBlank(approvalVo.getApply

return Message.error("'applyUser' is mi

}

apiService.setCreator(userName);

apiService.setModifier(userName);

this.apiService.save(apiService);

return Message.ok().data("insert\_id", apiSe

}

public Message create(@RequestBody ApiServiceVo api

return ApiUtils.doAndResponse(() - {

if (apiService.getWorkspaceId() == null){

apiService.setWorkspaceId(180L);

}

if (StringUtils.isBlank(apiService.getAlias

return Message.error("'api service alia

}

if (StringUtils.isBlank(apiService.getScrip

return Message.error("'api service scri

}

if (StringUtils.isBlank(apiService.getConte

return Message.error("'api service scri

}

if (null == apiService.getWorkspaceId()) {

return Message.error("'api service work

}

if (apiService.getContent().contains(";"))

if(!apiService.getContent().toLowerCase

return Message.error("'api service

}

}

if (WRITABLE\_PATTERN.matcher(apiService.get

return Message.error("'api service scri

}

MapString, Object metadata = apiService.get

if (apiService.getScriptPath().endsWith(".j

if (MapUtils.isEmpty(metadata)) {

return Message.error("'api service

}

MapString, Object configuration = (MapS

if (MapUtils.isEmpty(configuration)) {

return Message.error("'api service

}

MapString, Object datasource = (MapStri

if (MapUtils.isEmpty(datasource)) {

return Message.error("'api service

}

}

String userName = SecurityFilter.getLoginUs

SetConstraintViolationApiServiceVo result =

if (result.size() 0) {

throw new ConstraintViolationException(

}

ApprovalVo approvalVo = apiService.getAppro

if (StringUtils.isBlank(approvalVo.getAppro

return Message.error("'approvalName' is

}

if (StringUtils.isBlank(approvalVo.getApply

return Message.error("'applyUser' is mi

}

apiService.setCreator(userName);

apiService.setModifier(userName);

this.apiService.saveByApp(apiService);

return Message.ok().data("insert\_id", apiSe

}

public Message update(@RequestBody ApiServiceVo api

@PathVariable("api\_service\_v

HttpServletRequest req) {

return ApiUtils.doAndResponse(() - {

if (StringUtils.isBlank(apiService.getScrip

return Message.error("'api service scri

}

if(apiServiceVersionId !=0) {

if (StringUtils.isBlank(apiService.getP

return Message.error("'api service

}

}

if (StringUtils.isBlank(apiService.getConte

return Message.error("'api service scri

}

if (null == apiService.getWorkspaceId()) {

return Message.error("'api service work

}

if (null == apiService.getTargetServiceId()

return Message.error("'api service upda

}

if (apiService.getContent().contains(";"))

return Message.error("'api service scri

}

ApprovalVo approvalVo = apiService.getAppro

if (StringUtils.isBlank(approvalVo.getApply

return Message.error("'applyUser' is mi

}

if (WRITABLE\_PATTERN.matcher(apiService.get

return Message.error("'api service scri

}

MapString, Object metadata = apiService.get

if (apiService.getScriptPath().endsWith(".j

if (MapUtils.isEmpty(metadata)) {

return Message.error("'api service

}

MapString, Object configuration = (MapS

if (MapUtils.isEmpty(configuration)) {

return Message.error("'api service

}

MapString, Object datasource = (MapStri

if (MapUtils.isEmpty(datasource)) {

return Message.error("'api service

}

}

String userName = SecurityFilter.getLoginUs

SetConstraintViolationApiServiceVo result =

if (result.size() 0) {

throw new ConstraintViolationException(

}

apiService.setLatestVersionId(apiServiceVer

apiService.setModifier(userName);

apiService.setModifyTime(Calendar.getInstan

this.apiService.update(apiService);

return Message.ok().data("update\_id", apiSe

}

public Message query(@RequestParam(required = false

@RequestParam(requi

@RequestParam(requi

@RequestParam(requi

@RequestParam(requi

HttpServletRequest

String userName = SecurityFilter.getLoginUserna

return ApiUtils.doAndResponse(() - {

if (null == workspaceId) {

return Message.error("'api service sear

}

ApiServiceQuery query = new ApiServiceQuery

query.setWorkspaceId(workspaceId);

if(!this.apiService.checkUserWorkspace(user

return Message.error("'api service sear

}

ListApiServiceVo queryList = apiService.que

return Message.ok().data("query\_list", quer

}

public Message getUserServices(@RequestParam(requir

HttpServletRequest req){

String userName = SecurityFilter.getLoginUserna

return ApiUtils.doAndResponse(() - {

if(!this.apiService.checkUserWorkspace(user

return Message.error("'api service getU

}

ListApiServiceVo apiServiceList = apiService.qu

return Message.ok().data("query\_list", apiServi

}

public Message query( HttpServletRequest req,@Reque

String userName = SecurityFilter.getLoginUserna

return ApiUtils.doAndResponse(() - {

ListString tags= apiService.queryAllTags(us

return Message.ok().data("tags", tags);

}

public Message queryByScriptPath(@RequestParam(requ

HttpServletReques

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (StringUtils.isBlank(scriptPath)) {

return Message.error("'api service scri

}

ApiServiceVo apiServiceVo = apiService.quer

if(null != apiServiceVo) {

if (!this.apiService.checkUserWorkspace

return Message.error("'api service

}

if (apiServiceVo.getCreator().equals(us

return Message.ok().data("result",

} else {

return Message.error("'api service

}

}else {

return Message.ok().data("result", apiS

}

}

public Message queryById(@RequestParam(required = f

HttpServletRequest req) {

String userName = SecurityFilter.getLoginUserna

return ApiUtils.doAndResponse(() - {

if (id==null) {

return Message.error("'api service id'

}

ApiServiceVo apiServiceVo = apiService.quer

AssertUtil.notNull(apiServiceVo,"未找到数据服务，有可

if(!this.apiService.checkUserWorkspace(user

return Message.error("'api service quer

}

return Message.ok().data("result", apiServi

}

public Message checkPath(@RequestParam(required = f

return ApiUtils.doAndResponse(() - {

if (StringUtils.isBlank(scriptPath)) {

return Message.error("'api service scri

}

if (StringUtils.isBlank(path)) {

return Message.error("'api service path

}

Integer apiCount = apiService.queryCountByP

return Message.ok().data("result", 0 Integ

}

public Message checkName(@RequestParam(required = f

return ApiUtils.doAndResponse(() - {

if (StringUtils.isBlank(name)) {

return Message.error("'api service name

}

Integer count = apiService.queryCountByName

return Message.ok().data("result", count 0

}

public Message apiDisable(@RequestParam(required =

HttpServletRequest req)

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (null == id) {

return Message.error("'api service api

}

boolean resultFlag = apiService.disableApi(

return Message.ok().data("result", resultFl

}

public Message apiEnable(@RequestParam(required = f

HttpServletRequest req) {

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (null == id) {

return Message.error("'api service api

}

boolean resultFlag = apiService.enableApi(i

return Message.ok().data("result", resultFl

}

public Message apiDelete(@RequestParam(required = f

HttpServletRequest req)

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (null == id) {

return Message.error("'api service api

}

boolean resultFlag = apiService.deleteApi(i

return Message.ok().data("result", resultFl

}

public Message apiCommentUpdate(HttpServletRequest

@RequestBody ApiCom

Long id = apiCommentUpdateRequest.getId();

String comment = apiCommentUpdateRequest.getCom

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (null == id) {

return Message.error("'api service api

}

boolean resultFlag = apiService.updateComme

return Message.ok().data("result", resultFl

}

public Message apiParamQuery(@RequestParam(required

@RequestParam(require

HttpServletRequest re

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (StringUtils.isEmpty(scriptPath)) {

return Message.error("'api service api

}

if (null == versionId) {

return Message.error("'api service api

}

ListQueryParamVo queryParamVoList = apiServ

return Message.ok().data("result", queryPar

}

public Message apiVersionQuery(@RequestParam(requir

HttpServletRequest

return ApiUtils.doAndResponse(() - {

String userName = SecurityFilter.getLoginUs

if (null == serviceId) {

return Message.error("'api service api

}

ListApiVersionVo apiVersionVoList = apiServ

.stre

.coll

return Message.ok().data("result", apiVersi

}

public Message apiContentQuery(@RequestParam(requir

HttpServletRequest

String userName = SecurityFilter.getLoginUserna

return ApiUtils.doAndResponse(() - {

if (null== versionId) {

return Message.error("'api service api

}

ApiServiceVo apiServiceVo = apiServiceQuery

if(!this.apiService.checkUserWorkspace(user

return Message.error("'api service apiC

}

return Message.ok().data("result", apiServi

}

}

package com.webank.wedatasphere.dss.apiservice.core.res

import com.fasterxml.jackson.core.JsonProcessingExcepti

import com.fasterxml.jackson.databind.JavaType;

import com.webank.wedatasphere.dss.apiservice.core.conf

import com.webank.wedatasphere.dss.apiservice.core.exce

import com.webank.wedatasphere.dss.apiservice.core.exec

import com.webank.wedatasphere.dss.apiservice.core.exec

import com.webank.wedatasphere.dss.apiservice.core.serv

import com.webank.wedatasphere.dss.apiservice.core.toke

import com.webank.wedatasphere.dss.apiservice.core.util

import com.webank.wedatasphere.dss.apiservice.core.util

import com.webank.wedatasphere.dss.apiservice.core.vo.M

import com.webank.wedatasphere.dss.common.utils.DSSComm

import org.apache.commons.collections4.MapUtils;

import org.apache.commons.io.IOUtils;

import org.apache.commons.lang3.exception.ExceptionUtil

import org.apache.linkis.server.BDPJettyServerHelper;

import org.apache.linkis.server.Message;

import org.apache.linkis.server.security.SecurityFilter

import org.apache.linkis.ujes.client.UJESClient;

import org.apache.linkis.ujes.client.response.JobExecut

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Aut

import org.springframework.util.CollectionUtils;

import org.springframework.util.StringUtils;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.DefaultValue;

import javax.ws.rs.HttpMethod;

import javax.ws.rs.core.Response;

import java.io.IOException;

import java.io.InputStream;

import java.util.Enumeration;

import java.util.HashMap;

import java.util.Map;

import java.util.regex.Pattern;

@RestController

public class ApiServiceExecuteRestfulApi {

public static final String XLSX\_RESPONSE\_CONTENT\_TY

private static final Logger logger = LoggerFactory.

@Autowired

private ApiServiceQueryService queryService;

private static final String SYS\_COLUMN\_PREFIX = "\_"

private static final String requestBodyDemo = "{\"m

public Message post(@PathVariable("path") VariableS

HttpServletRequest req) {

String userName = SecurityFilter.getLoginUserna

return getResponse(userName,path.getPath(), que

}

public Message get(@PathVariable("path") VariableSt

HttpServletRequest req) throws

String userName = SecurityFilter.getLoginUserna

QueryRequest queryRequest = new QueryRequest();

if (MapUtils.isEmpty(queryRequest.getParams()))

String paramSuffix = "params.";

EnumerationString parameterNames = req.getP

MapString, Object params = new HashMap();

while (parameterNames.hasMoreElements()) {

String name = parameterNames.nextElemen

if (name.startsWith(paramSuffix)) {

params.put(name.substring(paramSuff

}

}

queryRequest.setParams(params);

}

String paramsJsonStr = req.getParameter("params

JavaType javaType = BDPJettyServerHelper.jackso

MapString,Object parmas = BDPJettyServerHelper.

queryRequest.setParams(parmas);

if (StringUtils.isEmpty(queryRequest.getModuleN

queryRequest.setModuleName(req.getParameter

}

return getResponse(userName,path.getPath(), que

}

public Message put(@PathVariable("path") VariableSt

HttpServletRequest req) {

String userName = SecurityFilter.getLoginUserna

return getResponse(userName,path.getPath(), que

}

public Message delete(@PathVariable("path") Variabl

HttpServletRequest req) {

String userName = SecurityFilter.getLoginUserna

return getResponse(userName,path.getPath(), que

}

private void validParam(QueryRequest queryRequest)

AssertUtil.notNull(queryRequest, "请求体不能为空，正确的格式

AssertUtil.notEmpty(queryRequest.getModuleName(

AssertUtil.notNull(queryRequest.getParams().get

}

public Message getDirFileTrees(HttpServletRequest r

@RequestParam(required

@RequestParam(required

String userName = SecurityFilter.getLoginUserna

if (!isNumber(taskId)) {

return Message.error("请求参数 taskId 非法.");

} else if(StringUtils.isEmpty(taskId)){

return Message.error("taskId 为空.");

}

if (StringUtils.isEmpty(path)) {

return Message.error("path 为空.");

}

ApiServiceJob apiServiceJob = queryService.getJ

if(apiServiceJob == null) {

return Message.error("当前用户不存在运行的TaskId: " +

} else if(userName.equals(apiServiceJob.getSubm

JobExecuteResult jobExecuteResult = new Job

jobExecuteResult.setTaskID(taskId);

jobExecuteResult.setUser(apiServiceJob.getP

UJESClient client = LinkisJobSubmit.getClie

String dirFileTree = ExecuteCodeHelper.getR

return DSSCommonUtils.COMMON\_GSON.fromJson(

} else {

return Message.error("You are not the submi

}

}

private void writeMessage(HttpServletResponse resp,

if(message == null) {

return;

} else {

resp.setStatus(Message.messageToHttpStatus(

}

String str = DSSCommonUtils.COMMON\_GSON.toJson(

resp.getWriter().println(str);

resp.getWriter().flush();

}

public Message openFile(HttpServletRequest req,

@RequestParam(required = f

@RequestParam(required = f

@RequestParam(required = f

@RequestParam(required = f

@RequestParam(required = f

String userName = SecurityFilter.getLoginUserna

logger.info("User {} wants to open resultSet fi

if (!isNumber(taskId)) {

return Message.error("请求参数 taskId 非法.");

} else if(StringUtils.isEmpty(taskId)){

return Message.error("taskId 为空.");

}

if (StringUtils.isEmpty(path)) {

return Message.error("path 为空.");

}

ApiServiceJob apiServiceJob = queryService.getJ

if(apiServiceJob == null) {

return Message.error("您不存在运行的TaskId: "+task

} else if (userName.equals(apiServiceJob.getSub

UJESClient client = LinkisJobSubmit.getClie

try {

String fileContent = ExecuteCodeHelper.

return DSSCommonUtils.COMMON\_GSON.fromJ

} catch (Exception e) {

logger.error("User {} fetch resultSet {

return Message.error("Get resultSet fai

}

} else {

return Message.error("You are not the submi

}

}

@RequestMapping(value = "resultsetToExcel",method =

public void resultsetToExcel(

HttpServletRequest req,

HttpServletResponse resp,

@RequestParam(required = false, name = "pat

@RequestParam(required = false, name = "tas

@RequestParam(required = false, name = "cha

@RequestParam(required = false, name = "out

@RequestParam(required = false, name = "csv

@RequestParam(required = false, name = "out

@RequestParam(required = false, name = "she

@RequestParam(required = false, name = "nul

String userName = SecurityFilter.getLoginUserna

logger.info("User {} wants to download resultSe

if (!isNumber(taskId)) {

writeMessage(resp, Message.error("请求参数 task

return;

} else if(StringUtils.isEmpty(taskId)){

writeMessage(resp, Message.error("taskId 为空

return;

}

if (StringUtils.isEmpty(path)) {

writeMessage(resp, Message.error("path 为空."

return;

}

resp.addHeader("Content-Disposition", "attachme

+ new String(outputFileName.getBytes("U

resp.setCharacterEncoding(charset);

switch (outputFileType) {

case "csv":

break;

case "xlsx":

resp.addHeader("Content-Type", XLSX\_RES

break;

default:

writeMessage(resp, Message.error("不支持的下

return;

}

InputStream inputStream = null;

ApiServiceJob apiServiceJob = queryService.getJ

if(null == apiServiceJob) {

writeMessage(resp, Message.error("您不存在运行的Ta

return;

}