**Prérequis**

-jre version 32bits car la 64bits ne fonctionne pas avec com4j, a télécharger sur le lien ci-après (<https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>), pour l’installer il suffit de faire un double clic sur l’exécutable et suivre les instructions.

-jdk (<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> )et netbeans ou eclipse pour le dev.

**Comment se connecter au Quality Center depuis un programme java ?**

La solution la plus rapide et la plus simple est l’utilisation de [com4j](https://com4j.java.net/). Il s’agit d’une API supportée par le très actif *Kohsuke Kawaguchi* qui permet l’interopérabilité avec COM (Microsoft Component Object Model).

Pour commencer, il faut récupérer le projet com4j (https://github.com/kohsuke/com4j/downloads), le dé-zipper puis exécuter la commande suivante :

**java -jar tlbimp.jar -o client -p com.wordpress.doktapepa.qc.client "C:\Program Files\Fichiers** **communs\Mercury Interactive\Quality Center\OTAClient.dll**

Il faut au préalable enregistrer les fichiers dll (OTAClient + COM4J)

Vous pourrez ensuite importer les classes générée dans un projet Java classique afin d’utiliser l’API Quality Center.

**Code java pour se connecter au Quality Center**

// Initialize Quality Center connection

ITDConnection connection = ClassFactory.createTDConnection();

connection.initConnectionEx("<http://localhost:8081/qcbin>");

connection.connectProjectEx(domainName, projectName, userName, userPassword);

**Code java pour manipuler les bugs**

// Retrieve the bug factory

IBugFactory bugFactory = connection.bugFactory().queryInterface(IBugFactory.class);

// Retrieve all bugs

IList allBugs = bugFactory.newList();

// Retrieve filtered bugs

ITDFilter filter = bugFactory.filter().queryInterface(ITDFilter.class);

filter.filter("BG\_MY\_CUSTOM\_FIELD", "TOTO");

IList filteredBugs = filter.newList();

// Iterate on bugs

Iterator iterator = allBugs.iterator();

while(iterator.hasNext()){

  IBug bug = iterator.next().queryInterface(IBug.class);

  // Few accessible properties

  bug.id ()

  bug.summary()

  bug.status()

  bug.priority()

  bug.field("BG\_MY\_CUSTOM\_FIELD")//recuperer un champs spécifique

  bug.detectedBy()

  bug.assignedTo()

}

**Code java pour ajouter un enregistrement (defect)**

Variant v = new Variant(Variant.Type.VT\_NULL);//créer un variant

IBug bug =(IBug) bugFactory.addItem(v).queryInterface(IBug.class); //créer un bug vide

bug.field("BG\_USER\_10",ref); //Donner la valeur ref au champ BG\_USER\_10

bug.post();//Pour enregistrer

**Code java pour modifier un enregistrement (defect)**

IBug bug = it.next().queryInterface(IBug.class);

bug.field("champs que vous voulez modifier",valeur);

bug.post();//Pour enregistrer

**creation Attchements( piéces jointes)**

-connexion au QC

IBugFactory bugFactoryS = itdS.bugFactory().queryInterface(IBugFactory.class); //recuperation des bugs

ITDFilter filterS = bugFactoryS.filter().queryInterface(ITDFilter.class);

filterS.filter(referenceIP, ref.toString());//filtrer les bugs par reference

IList filteredBugsS = filterS.newList();//mettre les bugs filtrés dans une liste

java.util.Iterator<Com4jObject> itFILTERS=filteredBugsS.iterator();

while (itFILTERS.hasNext()){

try{

IBug bugS = itFILTERS.next().queryInterface(IBug.class);

IAttachmentFactory attS=(IAttachmentFactory) bugS.attachments().queryInterface(IAttachmentFactory.class); //recuperation **des** attachements de ce bug

IList listS=attS.newList("");//mettre les attachments dans une liste

java.util.Iterator<Com4jObject> iter2=listS.iterator();

File file=new File("C:/Users/sdidi/Documents/NetBeansProjects/javafx\_1/src/attachmentSYNCH/"+list\_attachmentIP.get(j).nom\_attachment); //il faut obligatoirement créer un FILE( ne pas mettre directement un STRING)

String folderName = file.getParent();

while (iter2.hasNext()){

IAttachment attach = iter2.next().queryInterface(IAttachment.class); //recuperer la piece jointe

String fileName = attach.name(1);

IExtendedStorage extAttach = attach.attachmentStorage().queryInterface(IExtendedStorage.class);

extAttach.clientPath(folderName); // chemin ou se télecharger le fichier

extAttach.load(attach.name(1), true); // attach.name(1)==le nom du fichier a telecharger

Object ob=attach.data();

}

}//fin while

}

catch(Exception e) {System.out.println("QC Exceptione : "+e.getMessage());}

}

**Uploader un atatchment dans un bug QC**

-connexion au QC

IBugFactory bugFactoryS = itdS.bugFactory().queryInterface(IBugFactory.class); //recuperation des bugs

ITDFilter filterS = bugFactoryS.filter().queryInterface(ITDFilter.class);

filterS.filter(referenceIP, ref.toString());//filtrer les bugs par reference

IList filteredBugsS = filterS.newList();//mettre les bugs filtrés dans une liste

java.util.Iterator<Com4jObject> itFILTERS=filteredBugsS.iterator();

while (itFILTERS.hasNext()){

try{

IBug bugS = itFILTERS.next().queryInterface(IBug.class);

IAttachmentFactory attS=(IAttachmentFactory) bugS.attachments().queryInterface(IAttachmentFactory.class); //recuperation **des** attachements de ce bug

IAttachment attach = att.addItem(fileName).queryInterface(IAttachment.class);

IExtendedStorage extAttach = attach.attachmentStorage().queryInterface(IExtendedStorage.class);

extAttach.clientPath(folderName); //le dossier qui contient le fichier a uploader

extAttach.save(fileName, true);

attach.post();

attach.refresh();

}