**FANALISI CARTELLA VIRTUALE**

**Premesse**

Premessa A Le classi menzionate in questo doc sono presenti su:

06-Requirements->Wave3->Cartella\_Virtuale->ANALISI BSB\_BO\_BSBATCH->Classi

Premessa B Le tabelle con relativi collegamenti sono presenti su:

06-Requirements->Wave3->Cartella\_Virtuale->ANALISI BSB\_BO\_BSBATCH->DB

Premessa C La cartella BsbConfigKeys contiene i valori di default, i valori reali vanno ricercati sul db tramite query

Esempio

BsbUtl.*getProperty*(BsbConfigKeys.***ECS\_NAMESPACE\_ADMIN\_USERNAME***)

Sulla classe BsbUtl

***ECS\_NAMESPACE\_ADMIN\_USERNAME("ecs\_namespace\_admin\_username","QUERY\_DEFAULT\_NOT\_DEFINED"),***

Ma il vero valore si ricava tramite questa query su ORA\_BSB\_USR

select config\_valore from bsb\_config

where config\_chiave = 'ecs\_namespace\_admin\_username'

Premessa D Le configurazioni dei servizi esterni sono presenti sul db ORA\_BSB\_USR effettuando la seguente query:

select

g.interf\_param\_id, b.interf\_nome, c.srv\_nome, g.interf\_param\_name, g.interf\_param\_value

from BSB\_ASS\_INTERF\_SRV a, BSB\_INTERFACCIA b, BSB\_SERVIZIO c, BSB\_POLICY e, BSB\_INTERFACCIA\_PARAMS g

where

a.INTERF\_SRV\_INTERF\_ID = b.INTERF\_ID

and a.INTERF\_SRV\_SRV\_ID = c.SRV\_ID

and b.interf\_ID = g.INTERF\_PARAM\_INTERF\_ID

and c.srv\_nome in ('yyyyy')

and b.interf\_nome in ('xxxxx')

group by g.interf\_param\_id,b.interf\_nome, c.srv\_nome, g.interf\_param\_value, g.interf\_param\_name

Dove yyyyy è il nome del service, xxxxx è il nome dell’interfaccia

Premessa E In tutti gli Input FE verso BSB è sempre obbligatorio che siano presenti i seguenti campi:

**private** String provenienza = "";

**private** String tid = "";

**private** String username = "";

quindi per evitare ridondanze, in questo documento non saranno ripetuti i campi sopra elencati

**ELENCO SERVIZI BSB**

**Servizio BSB virtualfolder/create**

**Input**

com.nttdata.bsb.services.virtualfolder.create.CreateVirtualFolderInput

**Validazioni**

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.create.steps.LoginInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

p.s. (se la query restituisce risultati, stop processo con codice -1)

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_LOGIN” e interfaccia = “dellEmcLogin”

com.nttdata.bsb.services.virtualfolder.create.steps.LoginMappingStep

com.nttdata.bsb.services.virtualfolder.create.steps.ListUsersInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_LIST\_USERS” e interfaccia = “dellEmcListUsers”

com.nttdata.bsb.services.virtualfolder.create.steps.ListUsersMappingStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByObjectUser(userid)

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.objectUser like concat(:objusr,'%')

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepByUser

***CATENA CREATE\_VIRTUAL\_FOLDER\_USER\_PRESENT***

com.nttdata.bsb.services.virtualfolder.create.steps.UserInfoInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_USER\_SECRETKEY\_GETBUCKET” e interfaccia = “dellEmcCreateObjectUser”

com.nttdata.bsb.services.virtualfolder.create.steps.GetUserSecretKeyMappingStep

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepLogout

Com.nttdata.bsb.services.virtualfolder.create.steps.GetBucketsMappingStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_USER\_SECRETKEY\_GETBUCKET” e interfaccia = “dellEmcGetBuckets”

com.nttdata.bsb.services.virtualfolder.create.steps.AddUserSecretBucketMappigStep

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepLogout

com.nttdata.bsb.services.virtualfolder.create.steps.CreateBucketInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_CREATE\_BUCKET” e interfaccia = “dellEmcCreateBucket”

com.nttdata.bsb.services.virtualfolder.create.steps.CreateBucketMappingStep

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepLogout

***CATENA CREATE\_VIRTUAL\_FOLDER\_USER\_NOT\_PRESENT***

com.nttdata.bsb.services.virtualfolder.create.steps.CreateObjectUserInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_CREATE\_OBJECT\_USER” e interfaccia = “dellEmcCreateObjectUser”

com.nttdata.bsb.services.virtualfolder.create.steps.CreateObjectUserMappingStep

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepLogout

com.nttdata.bsb.services.virtualfolder.create.steps.CreateSecretKeyInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_CREATE\_SECRET\_KEY” e interfaccia = “dellEmcCreateSecretKey”

com.nttdata.bsb.services.virtualfolder.create.steps.CreateSecretKeyMappingStep

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepLogout

com.nttdata.bsb.services.virtualfolder.create.steps.CreateBucketInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_CREATE\_BUCKET” e interfaccia = “dellEmcCreateBucket”

com.nttdata.bsb.services.virtualfolder.create.steps.CreateBucketMappingStep

com.nttdata.bsb.services.virtualfolder.create.steps.ChainDispatcherStepLogout

***CATENA CREATE\_VIRTUAL\_FOLDER\_LOGOUT***

com.nttdata.bsb.services.virtualfolder.create.steps.LogoutInputDataTransformationStep

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_LOGOUT” e interfaccia = “dellEmcLogout”

com.nttdata.bsb.services.virtualfolder.create.steps.LogoutMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.create.CreateVirtualFolderResponse

**Servizio BSB virtualfolder/listObjects**

**Input**

com.nttdata.bsb.services.virtualfolder.listobjects.ListObjectsInput

**Validazioni**

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.listobjects.steps.ListObjectsInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_LIST\_OBJECTS” e interfaccia = “dellEmcListObjects”

com.nttdata.bsb.services.virtualfolder.listobjects.steps.ListObjectsMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.listobjects.ListObjectsResponse

**Servizio BSB virtualfolder/getFile**

**Input**

com.nttdata.bsb.services.virtualfolder.getfile.GetFileInput

**Validazioni**

absolutePath obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.getfile.steps.GetFileInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_GET\_FILE” e interfaccia = “dellEmcGetFile”

com.nttdata.bsb.services.virtualfolder.getfile.steps.GetFileMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.getfile.GetFileResponse

**Servizio BSB virtualfolder/isVirtualFolderActive**

**Input**

com.nttdata.bsb.services.virtualfolder.isvirtualfolderactive.VirtualFolderActiveInput

**Validazioni**

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.isvirtualfolderactive.steps.VirtualFolderActiveMappingStep

DAOFactory.*getProfileDAO*().hasRoleVirtualFolder(input.getCodiceFiscaleAzienda(),input.getUsername())

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT count(1) FROM BsbVirtualFolderConfDetail b WHERE b.id.customerFiscalCode = :cfa AND b.id.portalUsername = :username

**Output**

com.nttdata.bsb.services.virtualfolder.isvirtualfolderactive.VirtualFolderActiveResponse

**Servizio BSB virtualfolder/enableDisableVirtualFolder**

**Input**

com.nttdata.bsb.services.virtualfolder.enabledisablevirtualfolder.EnableDisableVirtualFolderInput

**Validazioni**

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.enabledisablevirtualfolder.steps.EnableDisableVirtualFolderMappingStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda())

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

DAOFactory.getProfileDAO().hasRoleVirtualFolder(input.getCodiceFiscaleAzienda(),input.getUsername());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT count(1) FROM BsbVirtualFolderConfDetail b WHERE b.id.customerFiscalCode = :cfa AND b.id.portalUsername = :username

**Output**

com.nttdata.bsb.services.virtualfolder.enabledisablevirtualfolder.EnableDisableVirtualFolderResponse

**Servizio BSB virtualfolder/createDirectory**

**Input**

com.nttdata.bsb.services.virtualfolder.createdirectory.CreateDirectoryInput

**Validazioni**

name obbligatorio

absolutePath obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.createdirectory.steps.CreateDirectoryInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_CREATE\_DIRECTORY” e interfaccia = “dellEmcCreateDirectory”

com.nttdata.bsb.services.virtualfolder.createdirectory.steps.CreateDirectoryMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.createdirectory.CreateDirectoryResponse

**Servizio BSB virtualfolder/deleteDirectory**

**Input**

com.nttdata.bsb.services.virtualfolder.deletedirectory.DeleteDirectoryInput

**Validazioni**

name obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.deletedirectory.steps.DeleteDirectoryInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_DELETE\_DIRECTORY” e interfaccia = “dellEmcDeleteDirectory”

com.nttdata.bsb.services.virtualfolder.deletedirectory.steps.DeleteDirectoryMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.deletedirectory.DeleteDirectoryResponse

**Servizio BSB virtualfolder/addFile**

**Input**

com.nttdata.bsb.services.virtualfolder.addfile.AddFileInput

**Validazioni**

name obbligatorio

absolutePath obbligatorio

content obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.addfile.steps.AddFileInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_ADD\_FILE” e interfaccia = “dellEmcAddFile”

com.nttdata.bsb.services.virtualfolder.addfile.steps.AddFileMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.addfile.AddFileResponse

**Servizio BSB virtualfolder/deleteFile**

**Input**

com.nttdata.bsb.services.virtualfolder.deletefile.DeleteFileInput

**Validazioni**

absolutePath obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.deletefile.steps.DeleteFileInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_DELETE\_FILE” e interfaccia = “dellEmcDeleteFile”

com.nttdata.bsb.services.virtualfolder.deletefile.steps.DeleteFileMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.deletefile.DeleteFileResponse

**~~Servizio BSB virtualfolder/copyFile~~**

**~~Input~~**

~~com.nttdata.bsb.services.virtualfolder.copyfile.CopyFileInput~~

**~~Validazioni~~**

~~name obbligatorio~~

~~oldAbsolutePath obbligatorio~~

~~newAbsolutePath obbligatorio~~

~~codiceFiscaleAzienda obbligatorio, mask\*~~

~~\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$~~

**~~Classi (in sequenza)~~**

~~com.nttdata.bsb.services.virtualfolder.copyfile.steps.CopyFileInputDataTransformationStep~~

~~DAOFactory.~~*~~getProfileDAO~~*~~().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());~~

~~Viene effettuata la query di seguito su ORA\_PROFILE\_USR:~~

~~SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa~~

~~Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_COPY\_FILE” e interfaccia = “dellEmcCopyFile”~~

~~com.nttdata.bsb.services.virtualfolder.copyfile.steps.CopyFileMappingStep~~

**~~Output~~**

~~com.nttdata.bsb.services.virtualfolder.copyfile.CopyFileResponse~~

**Servizio BSB virtualfolder/verifyCreateDirectory**

**Input**

com.nttdata.bsb.services.virtualfolder.checkcreatedirectory.VerifyCreateDirectoryInput

**Validazioni**

name obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.checkcreatedirectory.steps.VerifyCreateDirectoryMappingStep

DAOFactory.getProfileDAO().esitoCreazioneVirtualFolderOffline(input.getCodiceFiscaleAzienda(), input.getName());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderOffline b WHERE b.codiceFiscaleCliente = :cfa AND b.objeckKey = :name AND b.stato != :stato

**Output**

com.nttdata.bsb.services.virtualfolder.checkcreatedirectory.VerifyCreateDirectoryResponse

**Servizio BSB virtualfolder/verifySubDirectory**

**Input**

com.nttdata.bsb.services.virtualfolder.checksubdirectory.VerifySubDirectoryInput

**Validazioni**

name obbligatorio

codiceFiscaleAzienda obbligatorio, mask\*

\*mask = ^(([A-Z]{6}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}[0-9LMNPQRSTUV]{3}[A-Z]{1})|(0{5}[\d]{11})|([\d]{11}))$

**Classi (in sequenza)**

com.nttdata.bsb.services.virtualfolder.checksubdirectory.steps.VerifySubDirectoryInputDataTransformationStep

DAOFactory.*getProfileDAO*().getVirtualFolderConfByCFA(input.getCodiceFiscaleAzienda().toUpperCase());

Viene effettuata la query di seguito su ORA\_PROFILE\_USR:

SELECT b FROM BsbVirtualFolderConf b WHERE b.customerFiscalCode = :cfa

Viene richiamato il servizio esterno (Premessa D) con service = “VIRTUAL\_FOLDER\_LIST\_OBJECTS” e interfaccia = “dellEmcListObjects”

com.nttdata.bsb.services.virtualfolder.checksubdirectory.steps.VerifySubDirectoryMappingStep

**Output**

com.nttdata.bsb.services.virtualfolder.checksubdirectory.VerifySubDirectoryResponse

**Chiamate presenti su BO\_OLD verso BSB**

**virtualfolder/create**

**virtualfolder/listObjects**

**virtualfolder/getFile**

**virtualfolder/ isVirtualFolderActive**

**virtualfolder/enableDisableVirtualFolder**

**virtualfolder/createDirectory**

**virtualfolder/deleteDirectory**

**virtualfolder/addFile**

**virtualfolder/deleteFile**

**virtualfolder/verifyCreateDirectory**

**virtualfolder/verifySubDirectory**

**INFO BO OLD**

Per sapere se un utente che accede al BO è o meno abilitato alle operazioni di virtualFolder viene effettuata la chiamata al servizio BSB **virtualfolder/isVirtualFolderActive**

Solo se il servizio BSB restuisce true, l’utente è abilitato

Altrimenti comparirà il messaggio a video:

CARTELLA VIRTUALE NON PRESENTE PER IL CODICE FISCALE INDICATO: PER CREARNE UNA NUOVA PREMERE IL PULSANTE SEGUENTE:

**Chiamate presenti su BSBATCH verso BSB**

**virtualfolder/listObjects**

**virtualfolder/deleteDirectory**

**virtualfolder/deleteFile**

**ELENCO BATCH BSBATCH**

**BATCH virtualFolderDeleteDirectoryJob**

Il job permette di rimuovere tutte le cartelle virtuali che ne hanno fatto richiesta.

Viene recuperato la lista dei codici fiscali dal DB BSCHED e memorizzata in una lista

*SELECT CODICE\_FISCALE\_CLIENTE FROM BSB\_VIRTUAL\_FOLDER\_OFFLINE WHERE STATO = 1 ORDER BY ID\_VIRTUAL\_FOLDER ASC'*

Per ogni codice fiscale presente nella lista codici fiscali:

Vengono recuperati i dati utente dal DB ORA\_PROFILE\_USR e memorizzata in un oggetto virtualFolderMapperDTO

*SELECT \* FROM BSB\_VIRTUAL\_FOLDER\_OFFLINE WHERE CODICE\_FISCALE\_CLIENTE = ? AND STATO = 1*

Dove al posto di ‘?’ verrà inserito il codice fiscale di riferimento.

Viene eseguito il seguente update per modificare lo stato dell’utente da 1 a 2

*UPDATE BSB\_VIRTUAL\_FOLDER\_OFFLINE SET STATO = 2 WHERE CODICE\_FISCALE\_CLIENTE = ? AND STATO = 1 AND OBJECT\_KEY = ?*

Dove al posto di ‘?’ verranno inseriti rispettivamente il codice fiscale e il keyobject presenti entrambi sull’oggetto virtualFolderMapperDTO.

Viene effettuata la chiamata al servizio BSB /bsb/rest/virtualfolder/listObjects, inviando come request il cf.

Il servizio BSB restituirà come response una lista di oggetti VirtualFolderDeleteDirectoryDTO, tale lista verrà memotizzata nella variabile virtualFolderDeleteDirectoryList.

Elementi dell’oggetto VirtualFolderDeleteDirectoryDTO:

|  |
| --- |
| id |
| name |
| type |
| lastModified |
| absolutePath |
| size |
| parent |

Se la lista virtualFolderDeleteDirectoryList è piena, per ogni elemento della lista virtualFolderDeleteDirectoryList (virtualFolderDeleteDirectoryitem):

se è di tipo file ( type = ‘f’ ), viene effettuata la chiamata al servizio BSB /bsb/rest/virtualfolder/deleteFile

se è di tipo directory ( type = ‘d’ ), viene effettuata la chiamata al servizio BSB /bsb/rest/virtualfolder/deleteDirectory

Viene RI-effettuata la chiamata al servizio BSB /bsb/rest/virtualfolder/listObjects, inviando come request il cf.

Il servizio BSB restituirà come response una lista di oggetti VirtualFolderDeleteDirectoryDTO, tale lista verrà memotizzata nella variabile virtualFolderDeleteDirectoryList.

Se la lista virtualFolderDeleteDirectoryList è vuota:

Viene eseguito il seguente update per modificare lo stato dell’utente da 2 a 4

*UPDATE BSB\_VIRTUAL\_FOLDER\_OFFLINE SET STATO = 4 WHERE CODICE\_FISCALE\_CLIENTE = 2 AND STATO = 1 AND OBJECT\_KEY = ?*

Dove al posto di ‘?’ verranno inseriti rispettivamente il codice fiscale e il keyobject presenti entrambi sull’oggetto virtualFolderMapperDTO.

Se la lista virtualFolderDeleteDirectoryList è presente almeno un elemento:

Viene eseguito il seguente update per modificare lo stato dell’utente da 2 a 3

*UPDATE BSB\_VIRTUAL\_FOLDER\_OFFLINE SET STATO = 3 WHERE CODICE\_FISCALE\_CLIENTE = 2 AND STATO = 1 AND OBJECT\_KEY = ?*

Dove al posto di ‘?’ verranno inseriti rispettivamente il codice fiscale e il keyobject presenti entrambi sull’oggetto virtualFolderMapperDTO.