# Mobility2 Server REST APIs

**Help**: <http://localhost/Mobility2/MobilityService.svc/help>

**Latest Available Calls**: <http://localhost/Mobility2/MobilityService.svc/help>

**Example Request and Response**: <http://localhost/Mobility2/MobilityService.svc/help>

**Base URL:** <http://localhost/Mobility2/MobilityService.svc>

## Update History:

|  |  |
| --- | --- |
| 30 Aug 2013 | Added “ImageRenderType” JSON object for GetEmailDetail call .  Added GetVersionChenges call details. |
| … | … |
| 21 July 2014 | Security Update Jobs |

## Please Note:

1. Please always cross-check with <http://localhost/Mobility2/MobilityService.svc/help> for the latest details regarding an API
2. All Requests are in JSON format, except /UploadDocument which is multipart.

Hence these should have the Request header: **Content-Type: application/json**

1. All Results are in JSON format except /GetEmailAttachment and /GetCopy which returns Stream. Hence these will have Response Header: **Content-Type: application/json**
2. All Response will contain Response Header: **statusCode: 0,** if the value of this status code is non-zero, an error has occurred. In this case please look into the **StatusCode** and **StatusDetail** in the Response JSON for details.

Even though an error has occurred and the **statusCode** is non-zero, the HTTP code will be 200 OK.

1. Wherever date related values (DateTime) are requested or given as response, specified as a Int64 value, this denotes the number of seconds elapsed from 1970-1-1. It is the Epoch Time

public static long EpochTime(this DateTime curtime) {

var epoch = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);

return Convert.ToInt64((curtime - epoch).TotalSeconds);

}

public static DateTime FromEpochTime(long EpochTime) {

var epoch = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc).ToUniversalTime();

return epoch.AddSeconds(EpochTime);

}

1. Wherever object ***Dictionary <T, T1> XYZ*** is mentioned, the JSON will be of format,

"XYZ":

[

{

"Key":T,

"Value":T1

},

{

"Key":T,

"Value":T1

}

]

1. Wherever object ***KeyValuePair <T, T1>*** is mentioned, the JSON will be of format,

{

"Key":T,

"Value":T1

}

1. The Example Request JSONs provided are for illustrative purpose only, please refer the original definitions. For Example Response JSONs please use the Test UI tool.

## Basic Supporting Definitions:

### AuthenticationInfo

The AuthenticationInfo object is used to carry the credentials of the user on a per-call basis.

object {

string userID;

string password;

string domain;

string appUuid;

}\* AuthenticationInfo;

* **userID** - is the user’s logon name.
* **password** - is the user’s password.
* **domain** - is the user’s domain when doing trusted (domain-based) authentication. Must be empty/absent when using non-trusted authentication.
* **appUuid** - is a unique GUID created by the app at initialization. May be empty/unset.

### The following name/value pairs are required for all the Requests.

Int32 Version;

Double TimeZoneOffset;

* **Version** – the version of the Server, use 2.
* **TimeZoneOffset** – if DMS server is in a non-UTC format, this value could be used to set the TimeZone. This is a double value of the number of seconds offset from UTC for the device's timestamp (this is positive for Chicago, negative for Bangalore).

### DatabaseDetails:

object {

String DatabaseName;

Dictionary <Int32, Boolean> RoleValues;

}\* DatabaseDetails;

* **DatabaseName** – the name of the Database.
* **RoleValues**  – This is a dictionary of ***imRoleAttributeID*** enum and their Boolean Values.

### AdditionalProperty:

object {

String Name;

String Value;

}\* AdditionalProperty;

### AttributeDefinition:

object {

String Caption;

Int32 Attribute;

}\* AttributeDefinition;

### Database:

object {

array [AdditionalProperty, {0,}] ConfigurationData;

String Name;

String Description;

array [AttributeDefinition, (0,}] AttributeDefinitions;

Int32 EmailFieldCC;

Int32 EmailFieldSendTime;

Int32 EmailFieldReceiveTime;

Int32 EmailFieldBcc;

Int32 EmailFieldTo;

Int32 EmailFieldFrom;

}\* Database;

### Folder:

object {

String Owner;

Int32 ObjectType;

array [AdditionalProperty, {0,}] AddlProps;

Int64 EditTime;

Int32 Inherited;

String DBName;

String Name;

String Description;

Int64 FolderID;

Int64 ParentFolderID;

array [Folder, {0,} ] Subfolders;

array [Content, {0,} ] Contents;

String TargetDatabaseName;

Int64 TargetFolderID;

Int32 TargetFolderKind;

Int32 FolderKind;

Profile Profile;

Folder Target;

Dictionary <Int32, Boolean> ContentsOp;

Int32 DefaultSecurity;

}\* Folder;

* **AddlProps** – the set of additional properties for the folder. Only set when actively requested.
* **EditTime** – last EditTime of the Folder. Seconds from EpochTime
* **Inherited** – Security is inherited or not, (0 or 1)
* **DBName** – Database name in which the Folder is present.
* **Name** – Name of the Folder.
* **Description** – Description of the Folder.
* **FolderID** – ID of the Folder. This is the value that needs to be sent in further requests relating to the Folder.
* **ParentFolderID** – FolderID of the parent.
* **Subfolders –** Array of ***Folder***s. The children Folders.
* **Contents** – Array of ***Content***s. This represents the document contents of the Folder
* **TargetDatabaseName** – Target Database name if the Folder is a Shortcut.
* **TargetFolderID –** ID of the Target Folder if the Folder is a shortcut.
* **TargetFolderKind** – Returns the relating ***CustomFolderKind*** enum value.
* **FolderKind** – Returns the relating ***CustomFolderKind*** enum value.
* **Profile** – ***Profile*** values of the Folder, if it’s a workspace.
* **Target –** A ***Folder*** object of the resolved shortcut of this Folder
* **ContentsOp** – Dictionary of ***imContentsOp*** enumand their Boolean values.
* **DefaultSecurity** – If the Security of the Folder is inherited, a value of ‘73’ is returned, else the Default Visibility of the Folder is returned. This value might be anyone of the ***imSecurityType*** enum.

### SearchFolder:

object {

String Owner;

Int32 ObjectType;

array [AdditionalProperty, {0,}] AddlProps;

Int64 EditTime;

Int32 Inherited;

String DBName;

String Name;

String Description;

Int64 FolderID;

Int64 ParentFolderID;

array [Folder, {0,} ] Subfolders;

array [Content, {0,} ] Contents;

String TargetDatabaseName;

Int64 TargetFolderID;

Int32 TargetFolderKind;

Int32 FolderKind;

Profile Profile;

Folder Target;

Dictionary <String, String> ProfileSearchCriteria;

Dictionary <String, String> FolderSearchCriteria;

String Query

[SearchEmail](#_SearchEmail) SearchEmail

Dictionary <Int32, Boolean> ContentsOp;

Int32 DefaultSecurity;

}\* SearchFolder;

* ProfileSearchCriteria – Dictionary of ProfilesAttributes and their values.
* FolderSearchCriteria – Dictionary of Folder Attributes and their values.
* Query – Query String Used

### Workspace:

object {

Int64 EditTime;

String DatabaseName;

String Description;

Int32 FolderID;

Int32 ParentFolderID;

Int32 DefaultSecurity;

}\* Workspace;

### Content:

object {

String DatabaseName;

Profile Profile;

ContentShortcut Shortcut;

}\* Content;

* **DatabaseName** – Name of the Database the Document is present.
* **Profile** – ***Profile*** value of the document.
* **Shortcut** – ***ContentShortcut*** value if it’s a shortcut.

### Profile:

object {

String DatabaseName;

Dictionary < Int32, Object> Fields;

bool IsEmail;

}\* Profile;

* **DatabaseName** – Name of the Database the Document is present.
* **Fields** – Only thefollowing ***imProfilAttributeID*** are returned.

imProfileAttributeID.imProfileDatabase,

imProfileAttributeID.imProfileDocNum,

imProfileAttributeID.imProfileVersion,

imProfileAttributeID.imProfileDescription,

imProfileAttributeID.imProfileName,

imProfileAttributeID.imProfileAuthor,

imProfileAttributeID.imProfileOperator,

imProfileAttributeID.imProfileType,

imProfileAttributeID.imProfileClass,

imProfileAttributeID.imProfileSubClass,

imProfileAttributeID.imProfileEditDate,

imProfileAttributeID.imProfileCreateDate,

imProfileAttributeID.imProfileSize,

imProfileAttributeID.imProfileLocation,

imProfileAttributeID.imProfileDefaultSecurity,

imProfileAttributeID.imProfileInUseBy,

imProfileAttributeID.imProfileInUse,

imProfileAttributeID.imProfileCheckedOut,

imProfileAttributeID.imProfileArchived,

imProfileAttributeID.imProfileComment,

imProfileAttributeID.imProfileCustom1,

imProfileAttributeID.imProfileCustom2,

imProfileAttributeID.imProfileHasAttachment,

imProfileAttributeID.imProfileFrozen,

imProfileAttributeID.imProfileSystemModifyDate,

* **IsEmail** – whether the document profile is an Email or not.

### ContentShortcut:

object {

String Description;

String TargetDocType;

String TargetDatabase;

Int32 TargetDocNum;

Int32 TargetDocVer;

}\* ContentShortcut;

* **Description** – Description of the shortcut.
* **TargetDocType** – Target Document Type Name.
* **TargetDatabase** – Target Database.
* **TargetDocNum** – Target Document Number.
* **TargetDocVer** – Target Document Version.

### User:

object {

String UserID;

String FullName;

String Domain;

String Email;

Boolean IsExternal;

String Email2;

String Email3;

String Email4;

String Email5;

}\* User;

### AttachmentInfo:

object {

string Mime;

string Name;

Int64 Size;

Boolean Hidden;

String ID;

}\* **AttachmentInfo**;

### VersionInfo:

object {

string DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

Int64 EditDate;

}\* **VersionInfo**;

### UploadRequest:

object {

Int32 CheckinKind;

String DatabaseName;

Int32 FolderID;

Int32 DocNum;

Int32 DocVer;

Int32 Refile;

Int32 HistoryEvent;

String AppName;

String Comment;

String Location;

Int64 SystemEditDate;

Boolean SuppressHistory;

Int32 DefaultSecurity;

String Checksum;

Boolean SupervisedUpload;

}\* UploadRequest;

* **CheckinKind** – an ***imCheckinDisposition*** value. ***imCheckinReplaceDocumentOnly*** is not supported.
* **DatabaseName** – destination database name.
* **FolderID** – Upload the document to the mentioned Folder with FolderID. May be 0 for flatspace filing.
* **DocNum** – Document Number – ignored for NewDocument
* **DocVer** –Document Version – ignored for NewDocument
* **Refile** –
  + if ‘0’ document DefaultVisibilty is made “Private” and all ACLs (both User and Group) are cleared.
  + if ‘1’ document DefaultVisibility will be parent Folders DefaultVisibility and Folder ACLs (both User and Group) will be copied to the Document.
  + Default is ‘1’
* **DefaultSecurity –** this could be of any [imSecurityType](#_imSecurityType) value.
  + **Please Note:** This takes priority over Refile parameter.
  + Ie. When **Refile = 0,** and **DefaultSecurity** = **imView(86),**

Then ACLs will be cleared and then document DefaultVisibility will be set to View.

* + When **Refile = 1,** and **DefaultSecurity = imView(86),**

Then Folder ACLs will be copied to document, and then DefaultVisibility will be set to View.

* + When **Refile = 1,** and **DefaultSecurity** is not provided,

Then Folder ACLs will be copied to document, and Folder DefaultVisibilty will be copied to the document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Refile*** | ***DefaultSecurity*** | **Folder ACLs** | **Folder DefaultSecurity** | **Final Document DefaultSecurity** |
| 0 | imPublic | Not Copied | Not Copied | imPublic |
| 0 | imView | Not Copied | Not Copied | imView |
| 0 | imPrivate | Not Copied | Not Copied | imPrivate |
| 0 | Not defined (-1) | Not Copied | Not Copied | imPrivate |
| 1 | imPublic | Copied | Not Copied | imPublic |
| 1 | imView | Copied | Not Copied | imView |
| 1 | imPrivate | Copied | Not Copied | imPrivate |
| 1 | Not defined (-1) | Copied | Copied | Folders DefaultSecurity |

* **AppName** – Client Application Name.
* **Comment** – Any Comment related to this transaction.
* **Location** – May be empty.
* **SystemEditDate** – not used.
* **SuppressHistory** – Not used. Not functional.
* **SupervisedUpload –** Upload document, even when it is checked-out by another user. Only an Admin can use this provision.
* **Checksum –** Provide the MD5 checksum value for the file to upload,

By default this will be Empty String,

If this value is populated, then a MD5 Checksum is calculated in the Server for the uploaded file and compared with the one provided here,   
if the values are not equal an Exception is thrown.

Eg: "328B96DEC75297BEB5B0CCDC1FE71EE1"

### Group:

object {

String Db;

Int32 GroupNum;

String GroupName;

String GroupFullName;

Boolean GroupEnabled;

Boolean GroupIsExternal;

String GroupNOS;

}\* Group;

* **Db** – Name of the Database the in which the group resides.
* **GroupNum** – The unique numeric identifier for the group
* **GroupName** – Name of the group.
* **GroupDescription** – Description of the group.

### UpdateJobDetails:

object {

String JobDescription;

String JobID;

String JobStatus;

};

* **JobDescription**  – Description of the Job.
* **JobID** – ID of the Job

### HistoryEntry:

object {

String Db;

String DocumentNumber;

String User;

String DocumentVersion;

Int64 Date;

String Application;

String Operation;

};

### Document:

object {

Int32 Number;

Int32 Version;

};

## Supporting Enum Definitions:

### imRoleAttributeID

enum imRoleAttributeID

{

imRoleUseAdmin = 0,

imRoleUseMonitorApp = 1,

imRoleAdminUseImportApp = 2,

imRoleExternal = 3,

imRoleDocumentCheckout = 4,

imRoleDocumentCreate = 5,

imRoleDocumentDelete = 6,

imRoleDocumentFullTextSearch = 8,

imRoleDocumentReadOnly = 9,

imRoleDocumentUnlockDocument = 10,

imRoleDocumentViewPublicDocument = 11,

imRoleFolderCreatePublicFolder = 12,

imRoleFolderCreatePublicSearchFolder = 13,

imWebRoleSearchWorkspaces = 19,

imWebRoleAuthorWorkspaces = 20,

imWebRoleAuthorPublicWorkspaces = 21,

imWebRoleCreateSystemWorkspaces = 24,

imRoleWorkspaceDeleteWorkspace = 28,

imRoleAdminViewDocuments = 29,

}

### CustomFolderKind

enum CustomFolderKind

{

WSFolderKindMatters = 0,

WSFolderKindMatterShortcut = 11,

WSFolderKindFavorites = 1,

WSFolderKindWorkspace = 2,

WSFolderKindWorkspaceShortcut = 12,

WSFolderKindOrdinary = 3,

WSFolderKindShortcut = 4,

WSFolderKindWorkSiteExplorer = 5,

WSFolderKindDatabase = 6,

WSFolderKindMyFolders = 7,

WSFolderKindMyWorkSpaces = 8,

WSFolderKindOtherFolders = 9,

WSFolderKindOtherWorkSpaces = 10,

WSFolderKindSearch = 13,

WSFolderKindSearchShortcut = 15,

WSFolderKindTab = 14,

}

### imContentsOp

enum imContentsOp

{

imAddContentsOp = 0,

imRemoveContentsOp = 1,

imCheckInContentsOp = 2,

imUpdateContentShortcutOp = 3,

}

### imSecurityType

enum imSecurityType

{

imPublic = 80,

imView = 86,

imPrivate = 88,

}

### imCheckinDisposition

enum imCheckinDisposition

{

imCheckinNewVersion = 0,

imCheckinNewDocument = 1,

imCheckinReplaceOriginal = 2,

imCheckinReplaceDocumentOnly = 3,

imCheckinReplaceProfileOnly = 4,

}

### RenderType

enum RenderType

{

IgnoreImages = 0,

EmbedBase64Images = 1,

EmbedBase64ImagesZipped = 2,

ZippedFolder = 3,

}

* **IgnoreImages** - Just HTML. Do not include the images,
* **EmbedBase64Images** - HTML with Embedded Images,
* **EmbedBase64ImagesZipped** - HTML with Embedded Images – return this single HTML file as a ZIP file,
* **ZippedFolder** - HTML and resources inside a folder - return folder and its contents as a ZIP file

### imSearchEmail

enum imSearchEmail

{

imSearchEmailOrDocuments = 0,

imSearchEmailOnly = 1,

imSearchDocumentsOnly = 2,

}

### imNOSType:

enum imNOSType {

externalNOS = 1,

internalNOS = 2,

allNOS = 3,

}

### imSearchAttributeType:

enum imSearchAttributeType {

searchID = 1,

searchFullName = 2,

searchBoth = 3,

searchExact = 4,

}

### OperationKind

enum OperationKind

{

FolderContentsOps = 0,

DocumentOps = 1,

FoldersOps = 2,

FolderOps = 3,

}

### imFoldersOp

enum imFoldersOp

{

imAddFoldersOp = 0,

imRemoveFoldersOp = 1,

}

### imFolderOp

enum imFolderOp

{

imModifyFolderOp = 0,

imSetSecurityFolderOp = 1,

imRemoveThisFolderOp = 2,

imChangeNameFolderOp = 3,

imModifyLayoutOp = 4,

}

### imContentsOp

enum imContentsOp

{

imAddContentsOp = 0,

imRemoveContentsOp = 1,

imCheckInContentsOp = 2,

imUpdateContentShortcutOp = 3,

}

### imDocumentOperation

enum imDocumentOperation

{

imCheckOutDocumentOp = 0,

imCheckInDocumentOp = 1,

imCopyDocumentOp = 2,

imUpdateDocumentOp = 3,

imLockDocumentOp = 4,

imUnlockDocumentOp = 5,

imArchiveDocumentOp = 6,

imRestoreDocumentOp = 7,

imSetSecurityDocumentOp = 8,

imDeleteDocumentOp = 9,

imRelateDocumentOp = 10,

imSyncDocumentOp = 11,

imFreezeDocumentOp = 12,

imCheckinReplaceDocumentOp = 13,

imUnfreezeDocumentOp = 14,

imMoveDocumentOp = 15,

}

### imHistEvent

enum imHistEvent

{

imHistoryLaunch = 0,

imHistoryView = 1,

imHistoryCheckout = 2,

imHistoryCheckin = 3,

imHistoryProfileEdit = 4,

imHistoryClose = 5,

imHistoryNew = 6,

imHistoryVersionNew = 7,

imHistorySecurityChange = 8,

imHistoryCopy = 9,

imHistoryPrint = 10,

imHistoryMail = 11,

imHistoryEchoSync = 12,

imHistoryDelete = 13,

imHistoryPurge = 14,

imHistoryArchive = 15,

imHistoryRestore = 16,

imHistoryRelease = 17,

imHistoryExport = 18,

imHistoryModify = 19,

imHistoryEditTime = 20,

imHistoryNotLogged = 21,

imHistoryFrozen = 22,

imHistoryMigrated = 23,

imHistoryUndeclared = 24,

imHistoryReconciled = 25,

imHistoryRemoveFromFolder = 26,

imHistoryDeleteFolder = 27,

imHistoryWorkflowEvent = 28,

imHistoryWorkflowAttach = 29,

imHistoryWorkflowDetach = 30,

imHistoryShred = 1025,

}

### imFolderAttributeID

enum imFolderAttributeID

{

imFolderID = 0,

imFolderParentID = 1,

imFolderDefaultSecurity = 2,

imFolderName = 3,

imFolderOwner = 4,

imFolderDescription = 5,

imFolderLocation = 6,

imFolderType = 7,

imFolderSubtype = 8,

imFolderInherits = 9,

imFolderProfileID = 10,

imFolderVersion = 11,

imFolderCustom1 = 12,

imFolderCustom2 = 13,

imFolderCustom3 = 14,

imFolderRootFolderID = 15,

imFolderRootFolderType = 16,

imFolderIsContentSearch = 17,

imFolderIsFolderSearch = 18,

imFolderEmail = 19,

imFolderHiddenOnDesktop = 20,

imFolderEditDate = 21,

imFolderReferenceType = 22,

}

### imProfileAttributeID

enum imProfileAttributeID

{

imProfileDatabase = 0,

imProfileDocNum = 1,

imProfileVersion = 2,

imProfileDescription = 3,

imProfileName = 4,

imProfileAuthor = 5,

imProfileOperator = 6,

imProfileType = 7,

imProfileClass = 8,

imProfileSubClass = 9,

imProfileEditDate = 10,

imProfileCreateDate = 11,

imProfileRetainDays = 12,

imProfileSize = 13,

imProfileIndexable = 14,

imProfileIsRelated = 15,

imProfileLocation = 16,

imProfileDefaultSecurity = 17,

imProfileLastUser = 18,

imProfileInUseBy = 19,

imProfileNetNode = 20,

imProfileInUse = 21,

imProfileCheckedOut = 22,

imProfileArchived = 23,

imProfileComment = 24,

imProfileCustom1 = 25,

imProfileCustom2 = 26,

imProfileCustom3 = 27,

imProfileCustom4 = 28,

imProfileCustom5 = 29,

imProfileCustom6 = 30,

imProfileCustom7 = 31,

imProfileCustom8 = 32,

imProfileCustom9 = 33,

imProfileCustom10 = 34,

imProfileCustom11 = 35,

imProfileCustom12 = 36,

imProfileCustom13 = 37,

imProfileCustom14 = 38,

imProfileCustom15 = 39,

imProfileCustom16 = 40,

imProfileCustom17 = 41,

imProfileCustom18 = 42,

imProfileCustom19 = 43,

imProfileCustom20 = 44,

imProfileCustom21 = 45,

imProfileCustom22 = 46,

imProfileCustom23 = 47,

imProfileCustom24 = 48,

imProfileCustom25 = 49,

imProfileCustom26 = 50,

imProfileCustom27 = 51,

imProfileCustom28 = 52,

imProfileCustom29 = 53,

imProfileCustom30 = 54,

imProfileCustom1Description = 60,

imProfileCustom2Description = 61,

imProfileCustom3Description = 62,

imProfileCustom4Description = 63,

imProfileCustom5Description = 64,

imProfileCustom6Description = 65,

imProfileCustom7Description = 66,

imProfileCustom8Description = 67,

imProfileCustom9Description = 68,

imProfileCustom10Description = 69,

imProfileCustom11Description = 70,

imProfileCustom12Description = 71,

imProfileCustom29Description = 72,

imProfileCustom30Description = 73,

imProfileAuthorDescription = 74,

imProfileOperatorDescription = 75,

imProfileTypeDescription = 76,

imProfileClassDescription = 77,

imProfileSubClassDescription = 78,

imProfileLastUserDescription = 79,

imProfileInUseByDescription = 80,

imProfileEditTime = 81,

imProfileExtension = 82,

imProfileFullText = 83,

imProfileSubType = 84,

imProfileEditProfileTime = 85,

imProfileContainerID = 89,

imProfileCustom31 = 96,

imProfileMarkedForArchive = 97,

imProfileEchoEnabled = 98,

imProfileAccessTime = 99,

imProfileMessageUniqueID = 100,

imProfileIsExternal = 101,

imProfileHasAttachment = 102,

imProfileExternalAsNormal = 103,

imProfileProjectItemSID = 104,

imProfileProjectItemParentSID = 105,

imProfileCheckoutPath = 106,

imProfileCheckoutDueDate = 107,

imProfileCheckoutComments = 108,

imProfileCheckoutWhen = 109,

imProfileQuickRetrieve = 110,

imProfileFrozen = 111,

imProfileDeclareDate = 112,

imProfileFileCreateDate = 113,

imProfileFileModifyDate = 114,

imProfileSystemCreateDate = 115,

imProfileSystemModifyDate = 116,

imProfileRestorable = 117,

imProfileHipaaCompliant = 118,

imProfileFtLanguageId = 119,

}

### ContentTypes

{

document,

email,

contentShortcut

}

### UserFieldEnumerators

{

username,

userfullname,

userlocation,

userphone,

userextension,

userloginenabled,

userfax,

usercomments,

useremail,

userpreferreddatabase,

userdomainname,

usernos,

useremail2,

userpager,

usermobile,

userother,

usercustom1,

usercustom2,

usercustom3,

useremail3,

useremail4,

useremail5

}

### AccessRight

{

RightNone,

RightRead,

RightReadWrite,

RightAll

}

### DefaultSecurity

{

NoUpdate,

Public,

Private,

View

}

### JobTarget

{

documents,

workspaces,

folders

}

### JobStatus

{

StatusWaiting,

StatusRunning,

StatusCompleted,

StatusFailed,

StatusCancelled,

StatusAll

}

### JobFailureReason

{

JOB\_NO\_FAILURE,

JOB\_ID\_NOT\_FOUND,

JOB\_CANNOT\_BE\_CANCELLED\_IN\_THIS\_STATE

}

### JobType

{

add,

delete,

ensure

}

### NOS

{

NOSNovell3x,

NOSVirtual,

NOSWindowsNT,

NOSNovellNDS,

NOSExternal,

NOSNtADS,

NOSNetScapeDS

}

### FolderKind

{

folder,

workspace,

folderShortcut,

workspaceShortcut,

subscriptionShortcut

}

### SearchEmail

{

SearchEmailOrDocuments,

SearchEmailOnly,

SearchDocumentsOnly

}

### ProfileFieldEnumerators(Paging)

{

db,

num,

ver,

desc,

name,

author,

operator,

type,

class,

subClass,

editDate,

createDate,

retainDays,

size,

isRelated,

location,

defaultSecurity,

lastUser,

inUseBy,

netNode,

inUse,

checkedOut,

archived,

comment,

c1,

c2,

c3,

c4,

c5,

c6,

c7,

c8,

c9,

c10,

c11,

c12,

c13,

c14,

c15,

c16,

c17,

c18,

c19,

c20,

c21,

c22,

c23,

c24,

c25,

c26,

c27,

c28,

c29,

c30,

c31,

c1Desc,

c2Desc,

c3Desc,

c4Desc,

c5Desc,

c6Desc,

c7Desc,

c8Desc,

c9Desc,

c10Desc,

c11Desc,

c12Desc,

c29Desc,

c30Desc,

c31Desc,

authorDesc,

operatorDesc,

typeDesc,

classDesc,

subClassDesc,

lastUserDesc,

inUseByDesc,

editTime,

extension,

fullText,

subType,

editProfileTime,

containerID,

markedForarchive,

echoEnabled,

accessTime,

messageUniqueID,

isExternal,

hasAttachment,

externalAsNormal,

projectItemSID,

projectItemParentSID,

checkoutPath,

checkoutDueDate,

checkoutComments,

checkoutWhen,

quickRetrieve,

declared,

declareDate,

fileCreateDate,

fileModifyDate,

systemCreateDate,

systemModifyDate,

restorable,

hippaCompliant

}

### HistoryFieldEnumerators

Same as ProfileFieldEnumerators(Paging)

### FolderFieldEnumerators

{

FolderID,

FolderParentID,

FolderDefaultSecurity,

FolderName,

FolderOwner,

FolderDescription,

FolderLocation,

FolderType,

FolderSubtype,

FolderInherits,

FolderProfileID,

FolderVersion,

FolderCustom1,

FolderCustom2,

FolderCustom3,

FolderRootFolderID,

FolderRootFolderType,

FolderIsContentSearch,

FolderIsFolderSearch,

FolderEmail,

FolderHiddenOnDesktop,

FolderEditDate,

FolderReferenceType

}

## StatusCode and StatusDetail:

0 Success

1 General Failure

2 Invalid Server Configuration

200 Failed to create

2001 Cannot access WorkSite user.

2002 Cannot create WorkSite document.

2003 WorkSite Application Type not found.

2004 WorkSite destination folder not found.

2005 WorkSite ACLS error.

2006 WorkSite profile error.

2007 WorkSite configuration error.

2008 Cannot create WorkSite session.

2009 WorkSite admin session logon error.

2010 Cannot access WorkSite database.

2011 Cannot file to the WorkSite destination.

2012 Destination is a WorkSite shortcut that could not be resolved.

2013 WorkSite folder is read-only.

2014 Duplicate document found in WorkSite.

2015 Access denied to WorkSite location.

2016 Document already Checked-out

2017 Document Editdate is greater

2018 Failed to parse the multipart document

2019 Refile problem

2020 CheckinFailure

2021 Checkin disabled in iOS Server

2022 ReplaceOriginal disabled in iOS Server

2023 Document is not a latest version

2024 Not able to Create Folder/WorkSpace

2025 Document deletion failed

2026 Add/Remove Document from/to Folder failed

2027 Required role permission not set

2028 User is not a Supervisor

2029 Not able to Logon Supervisor

2030 Database not available

2031 Folder not found

2032 Root Folder would not be deleted

2033 WorkSpace Folder would not be deleted

2034 Contents for Folder are not available

2035 Document not found

2036 Create Public Folder in Database Permission Error

2037 SubFolders for given folder is Not Available

3 WorkSite Record Not found

3000 Error converting to HTML file

4 WorkSite Server Space Exhausted

5 UploadedFileChecksumFailure

5000 WorkSite Web Service error.

6001 DM Server

6002 Database

7000 User Disabled by WorkSite Policy

7001 Clients written for older servers are Not Allowed to connect

8000 Attachment not found. Please check the Attachment Name.

9000 Folder is not a Workspace or Document Folder

## Service Calls:

### DMSDetails - Get DMS Database Details:

The following API returns the details about the number of Databases in a Server and their RoleValues.

#### Method: POST /DMSDetails

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string DMSServer;

array [[DatabaseDetails](#_DatabaseDetails:), {0,} ] DataBase;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### Top - Get full Database, Server and Caption details:

The following API returns the details about the Databases in a Server, its Configuration Data, Favourites Folder and Matter Folder details.

#### Method: POST /Top

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

String UserID;

String SessionName;

String PreferredDatabaseName;

array [[AdditionalProperty](#_AdditionalProperty:), {0,}] ConfigurationData;

[Folder](#_Folder:) MatterFolder;

[Folder](#_Folder:) FavoritesFolder;

array [[Database](#_Database:), {0,}] Databases;

String DMSName;

Dictionary <Int32, bool> UserPreferences;

Dictionary <Int32, String> Captions;

Int32 ReplaceOriginal;

Dictionary <String, String> Policies;

Int32 FlatSpaceFiling;

[User](#_User:) UserInfo;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### MyMatters - Get MyMatters:

The following API returns the “MyMatters” Folder.

#### Method: POST /MyMatters

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

object {[Folder](#_Folder:)}\*;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### MyFavorites - Get MyFavorites:

The following API returns the “**MyFavorites**” Folder.

#### Method: POST /MyFavorites

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

object {[Folder](#_Folder:)}\*;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### MatterWorkList - Get MatterWorkList:

The following API returns the Folder array for “**MatterWorkList”**.

#### Method: POST /MatterWorkList

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Folder](#_Folder:), (0,}] Folders;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### WorkList - Get WorkList:

The following API returns the Contents of “**WorkList**” folder.

#### Method: POST /WorkList

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Content](#_Content:), (0,}] Contents;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### CheckoutList - Get CheckoutList:

The following API returns the details of the Checked out documents.

#### Method: POST /CheckoutList

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Content](#_Content:), (0,}] Contents;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2

}

### Folder - Get the Folder details:

The following API returns the details of a Folder.

#### Method: POST /Folder

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

Int64 FolderID;

String DatabaseName;

Boolean GetSubfolders;

Boolean GetContents;

Boolean GetAdditionalProperties;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

object {[Folder](#_Folder:)}\*;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"userID":"test1",

"password":"test1"

},

"DatabaseName":"DSNine",

"FolderID":3,

"GetSubfolders":true,

"GetContents":true,

"GetAdditionalProperties":true,

}

##### Discussion:

GetSubfolders and GetContents specify whether the Subfolders and Contents properties of the returned folder object are to be returned or not. If these are not set, the default value is TRUE.

The GetAdditionalProperties argument specifies whether the target folder’s AdditionalProperties collection is to be populated. Only the target folder’s AdditionalProperties is populated, not any Subfolders AdditionalProperties.

### SearchDocuments - Search for Documents:

The following API returns the document search result.

#### Method: POST /SearchDocuments

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int64 FolderID;

Int32 SearchType;

Dictionary<Int32, String> ProfileSearchCriteria;

}\*

* **SearchType –** An ***imSearchEmail*** value.
* **ProfileSearchCriteria – *imProfileAttributeID*** and their related value. Only add the imProfileAttributeID that has a value to search. Do not include null values for imProfileAttributeID.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Content](#_Content:), (0,}] Contents;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"DatabaseName":"DSNine",

"FolderID":3,

"SearchType":0,

"ProfileSearchCriteria":

[

{

"Key":3,

"Value":"del"

},

{

"Key":83,

"Value":"del"

}

]

}

### SearchFolders - Search for Folders:

The following API returns the document search result.

#### Method: POST /SearchFolders

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Dictionary<int, Object> FolderSearchCriteria;

}\*

* **FolderSearchCriteria – *imFolderAttributeID*** and their related value. Only add the ***imFolderAttributeID*** that has a value to search. Do not include null values for ***imFolderAttributeID***.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Folder](#_Folder:), (0,}] Folders;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"DatabaseName":"DSNine",

"FolderSearchCriteria":[

{

"Key":3,

"Value":"d"

}

]

}

### SearchWorkspaces - Search for Workspaces:

The following API returns the document search result.

#### Method: POST /SearchWorkspaces

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Dictionary<int, String> ProfileSearchCriteria;

Dictionary<int, Object> WorkspacesSearchCriteria;

}\*

* **ProfileSearchCriteria – *imProfileAttributeID*** and their related value. Only add the ***imProfileAttributeID*** that has a value to search. Do not include null values for ***imProfileAttributeID***.
* **WorkspacesSearchCriteria – *imFolderAttributeID*** and their related value. Only add the ***imFolderAttributeID*** that has a value to search. Do not include null values for ***imFolderAttributeID***.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Folder](#_Folder:), (0,}] Folders;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"DatabaseName":"DSNine",

"ProfileSearchCriteria":

[

],

"WorkspacesSearchCriteria":

[

{

"Key":3,

"Value":"w"

}

]

}

### SearchUsers - Search for Users:

#### Method: POST /SearchUsers

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DetailLevel;

String SearchCriteria;

Int32 SearchType;

Int32 SearchAtBeginning;

Int32 SearchUserType;

}\*

* **DatabaseName** - is empty, then the default database is searched (recommended).
* **SearchCriteria** - must be set. This is the search string. The response to wildcards is undefined.
* **DetailLevel –** should be **3** for now**.**
* **SearchAtBeginning –** Could be 0 or 1.
* **SearchUserType** - this may be left open for “all” or may be set for internal or external only. This is a [imNOSType](#_imNOSType:) value.
* **SearchType** - may be left blank for searchID or may be set. This is a [imSearchAttributeType](#_imSearchAttributeType:) value.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[User](#_User:), (0,}] Users;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"DatabaseName":"dsnine",

"DetailLevel":3,

"SearchCriteria":"test",

"SearchType":3,

"SearchAtBeginning":1,

"SearchUserType":2,

}

### GetCopy - Get a Copy of the Document:

The following API returns the document.

#### Method: POST /GetCopy

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

Boolean AsHTML;

Int64 SystemEditDate;

Int32 ImageRenderType;

Int32 ImageByteCount;

Boolean ChecksumRequired;

}\*

* **AsHTML –** if true will return a HTML representation of the File, else the original file will be returned.
* **ImageRenderType –** ignored if AsHTML is false. A ***RenderType*** value.
* **ImageByteCount –** ignored if AsHTML is false. The maximum byteCount allowed for the Embedded Image.
* **ChecksumRequired** – If this is set to true, the Response header will contain a “X-Checksum” header with the MD5 checksum for the File.

##### Response:

File as Stream

##### Response Header:

X-Content-Length: {size of the File}

X-Checksum: {MD5 checksum value of the File}

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"userID":"test1",

"password":"test1"

},

"DatabaseName":"DsNine",

"DocumentNumber":10,

"DocumentVersion":1,

"AsHTML":true,

"SystemEditDate":123456,

"ImageRenderType": 2,

"ImageByteCount": 32768,

“ChecksumRequired”:true

}

### GetEmailDetail - Get the Details of an Email document:

The following API returns the email document details.

#### Method: POST /GetEmailDetail

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocNum;

Int32 DocVer;

Int32 ImageRenderType;

}\*

* **ImageRenderType –** A ***RenderType*** value. But only **IgnoreImages** and **EmbedBase64Images** are supported.
  + **IgnoreImages –** Response Body HTML will not have the ContentIDImages, embedded to the HTML.
  + **EmbedBase64Images –** Response Body HTML will have the ContentIDAttachments embedded to the HTML.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string Subject;

string From;

array [string, (0,}] To;

array [string, (0,}] CC;

array [string, (0,}] BCC;

Int64 Sent;

Int64 Received;

array [[AttachmentInfo](#_AttachmentInfo:), {0,}] Attachments;

array [[AttachmentInfo](#_AttachmentInfo:), {0,}] ContentIDAttachments;

String ContentIDAttachmentsFolder;

String Body;

String BodySnippet;

Dictionary<String, String> ContentBodies;

Int64 EditDate;

}\*

* **Attachments –** These will contain the Attachment details of the Email. These attachments are the visible ones (generally displayed with a paper-clip). To get these attachments, a “/GetEmailAttachment” call to the Mobility Server is required.
* **ContentIDAttachments –** These will contain the Attachment details that are embedded inside the email. The attachments are converted to Base64 String and are provided in **ContentBodies.**

When **ImageRenderType** is **RenderType.EmbedBase64Images** this will be empty.

* **ContentIDAttachmentFolder –** The folder name inside which the Attachments could be copied, for viewing the HTML, do there would not be any broken links.

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"userID":"test1",

"password":"test1"

},

"DatabaseName":"DSNine",

"DocNum":11,

"DocVer":1,

“ImageRenderType”:1,

}

### GetEmailAttachment - Get an Attachment in Email:

The following API returns the attachment.

#### Method: POST /GetEmailAttachment

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocNum;

Int32 DocVer;

String AttachmentName;

String AttachmentID;

Boolean ChecksumRequired**;**

}\*

* **ChecksumRequired** – If this is set to true, the Response header will contain a “X-Checksum” header with the MD5 checksum for the File.
* **AttachmentName –** When attachment name is provided, the first attachment from email that has the same name as the “AttachmentName” will be returned.
* **AttachmentID –** When AttachmentID is provided along with “AttachmentName”, the attachment in the mail that has the same values for both of these parameters will be returned.

**If AttachmentID is provided, AttachmentName should also be provided.**

AttachmentID is mostly provided when a mail has more than one Attachment with the same name.

##### Response:

File as Stream

##### Response Header:

X-Content-Length: {size of the File}

X-Checksum: {MD5 checksum value of the File}

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"userID":"test1",

"password":"test1"

},

"DatabaseName":"DSNine",

"DocNum":173,

"DocVer":1,

"AttachmentName":"Urban Assessment.doc",

“ChecksumRequired”:true

}

### UploadDocument - Upload a single Document:

The following API returns the Profile of the Uploaded document.

#### Method: POST /UploadDocument

##### Request Multipart format for NewDocument, NewVersion, ReplaceOriginal:

Content-Type: multipart/form-data; boundary=AaB03x

--AaB03x

Content-Disposition: form-data; name="GeneralJson"

{

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

object {[UploadRequest](#_UploadRequest:)}\*

}

--AaB03x

Content-Disposition: form-data; name="ProfileJson"

{

//Profile values in the following format

"3":"PDF Upload",

"7":"ACROBAT",

"8":"DOC"

}

--AaB03x

Content-Disposition: form-data; name="APJson"

{

//Name Value Pairs. If providing iMan\_\_\_ date props please provide the Seconds in Value.

"UploadedBy":"XYZ",

"AnyName":"AnyValue"

"iMan\_\_\_45":"1392166857"

}

--AaB03x

Content-Disposition: form-data; name=" file0"; filename="*filename.pdf*"

Content-Type: *application/pdf*

…Binary Data…

--AaB03x--

##### Request Multipart format for ReplaceProfileOnly:

Content-Type: multipart/form-data; boundary=AaB03x

--AaB03x

Content-Disposition: form-data; name="GeneralJson"

{

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

object {[UploadRequest](#_UploadRequest:)}\*

}

--AaB03x

Content-Disposition: form-data; name="ProfileJson"

{

//Profile values in the following format

"3":"PDF Upload",

"7":"ACROBAT",

"8":"DOC"

}

--AaB03x--

##### Request Multipart format for ReplaceDocumentOnly:

Content-Type: multipart/form-data; boundary=AaB03x

--AaB03x

Content-Disposition: form-data; name="GeneralJson"

{

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

object {[UploadRequest](#_UploadRequest:)}\*

}

--AaB03x

Content-Disposition: form-data; name=" file0"; filename="*filename.pdf*"

Content-Type: *application/pdf*

…Binary Data…

--AaB03x--

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[AdditionalProperty](#_AdditionalProperty:), {0,}] AddlProps,

[Profile](#_Profile:) Profile;

Dictionary<String, String> ErrorDetails;

array [KeyValuePair<String, String>, {0,}] ErrorList;

}\*

* **Profile –** Profile of the Checked-in document.
* **ErrorDetails –** Any other error details.
* **ErrorList –** Profile Errors if any.

### SetFolderProperties - Set folder additional properties:

This sets properties for a folder.

#### Method: POST /SetFolderProperties

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 FolderID;

Array [AdditionalProperty, {0,}] AdditionalPropertiesToSet;

Array [AdditionalProperty, {0,}] AdditionalPropertiesToRemove;

Boolean Inherited;

Int32 DefaultSecurity;

String Description;

String Name;

Dictionary<int, String> ProfileValues;

}\*

##### Response:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

object {Folder}\*;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"userID":"test1",

"password":"test1"

},

"DatabaseName":"DSNine",

"FolderID":2557,

"Inherited":true,

"DefaultSecurity":88,

"Description":"Hello I am changing this",

"AdditionalPropertiesToSet":[ {"Name":"Name", "Value":"K Thompson"}, {"Name":"iMan\_\_\_8", "Value":"COMPARE"}],

"AdditionalPropertiesToRemove":[ ],

"ProfileValues":

[

{

"Key":4,

"Value":"Name Changed"

}

]

}

##### Discussion:

The request’s AdditionalPropertiesToSet will overwrite any additional properties included in the set without modifying or removing those not included. Thus it is a delta from the existing additional properties. Likewise, AdditionalPropertiesToRemove will remove any additional properties that exist the set of values without removing others.

**DefaultSecurity** – is a ***imSecurityType*** value.

**ProfileValues – *imProfileAttributeID*** and their related value. Only add the ***imProfileAttributeID*** that has a value to search. Do not include null values for ***imProfileAttributeID***.

The behavior of the security flags is as follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Inherited flag | | |
| Default Security |  | True | False | Not set |
| Set | Inherited flag wins | Default security flag wins | Default security flag wins |
| Not set | Inherited flag wins | Current effective security | No change |

Inherited flag is ignored for root level folders in the document.

The Description property, if set, replaces the current description.

The Profile property, if set, replaces the profile metadata. This applies only to a workspace and is ignored otherwise.

This call makes no change to a folder or workspace shortcut.

Proper rights to the folder are required or an error is returned.

The Folder in question is returned. Neither contents nor subfolders are set in that folder. The AdditionalProperties of the folder are set.

### SupervisedAddToHistory - Add History Event to a Document as a Supervisor:

The following API adds an History Event to a Document. It will succeed only if called by an Admin User.

The Admin Authentication info should be provided,

#### Method: POST /SupervisedAddToHistory

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

String UserID;

String Application;

Int64 DateTime;

String Activity;

String Location;

String Comments;

String Data1;

String Data2;

Int32 Num1;

Int32 Num2;

Int32 Num3;

Int32 Duration;

Int32 PagesPrintd;

Int32 Event;

}\*

**UserID –** The User under whose name the History Event should be logged.

**Application –** The name of the Application which calls this function.

**DateTime –** Time of Logging.

**Activity –** The activity as String, Best to have standard values like { Launch,View,Checkout,Checkin,ProfileEdit,Close,New,VersionNew,SecurityChange,Copy,Print,Mail,EchoSync,Delete,Purge,Archive,Restore,Release,Export,Modify,EditTime,NotLogged,Frozen,Migrated,Undeclared,Reconciled,RemoveFromFolder,DeleteFolder,WorkflowEvent,WorkflowAttach,WorkflowDetach,Shred}

**Location –** Server doing the work.

**Data** and **Num** fields are arbitrary.

**Event – *nrHistEvent*** enum value.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

}\*

##### Example Request JSON:

{

"Activity":"Activity",

"Application":"Fiddler",

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"Comments":"Comment what",

"Data1":"String content",

"Data2":"String content",

"DatabaseName":"DSNine",

"DateTime":9223375807,

"DocumentNumber":340,

"DocumentVersion":1,

"Duration":2,

"Event":1,

"Location":"",

"Num1":2147483647,

"Num2":2147483647,

"Num3":2147483647,

"PagesPrintd":2147483647,

"TimeZoneOffset":0,

"UserID":"test2",

"Version":2

}

### GetVersionChanges – Returns all latest versions of a document created after the given Document Profile values.

The server’s /GetVersionChanges call takes a set of basic document information (database, number, version and edit date) and determines a minimal set of changed profiles based on those.

Profiles may be changed for two reasons:

1. Because there exists a newer version. In this case, the profile of the latest version is returned if it is not included in the requested document set. Or,

2. because the given version (matching the given database, number and version) has a newer edit date. In this case, the profile of the requested version is returned.

In all other cases, no version information is returned.

The ordering of returned profiles is indeterminate.

#### Method: POST / GetVersionChanges

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

array [ integer, {0,} ] [VersionInfo](#_VersionInfo:);

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [ [Profile](#_Profile:), {0,} ] Profiles;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"VersionInfo":[{

"DatabaseName":"dsnine",

"DocumentNumber":8479,

"DocumentVersion":1,

"EditDate":1347628478

}]

}

### AllVersions – Returns all the versions of a document.

#### Method: POST / AllVersions

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

string DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [ [Content](#_Content:), {0,} ] Contents;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DocumentNumber":98,

"DocumentVersion":1,

"DatabaseName":"dsnine"

}

### CreateFolder – Creates a new folder.

#### Method: POST /CreateFolder

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

String FolderName;

Int32 ParentFolderID;

Boolean InheritSecurity;

String Description;

Int32 DefaultSecurity;

Array [AdditionalProperty, {0,}] AdditionalPropertiesToSet;

}\*

* **ParentFolderID** – Under which folder should the new folder be created
* **InheritSecurity –** ShouldInherity security from the parent?
* **DefaultSecurity –** If Inherit Security is false, then Default Security can be of [**imSecurityType**](#_imSecurityType)value.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

object {[Folder](#_Folder:)}\*;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"dsnine",

"FolderName":"NewFolder",

"ParentFolderID":3,

"InheritSecurity":false,

“DefaultSecurity”:88,

"Description":"Hello"

}

### CreateWorkspace – Creates a new Workspace.

#### Method: POST /CreateWorkspace

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

String WorkspaceName;

String WorkspaceSubtype;

String Description;

Int32 DefaultSecurity;

String Body;

Dictionary<int, String> Profile;

}\*

* **Profile** – Dictionary of **imProfileAttribute** values

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

[Workspace](#_Workspace:) workspace;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"dsnine",

"WorkspaceName":"Hello",

"DefaultSecurity":87,

"WorkspaceSubtype":"",

“Description”: "",

“Body”:"",

"Profile":

[

{

"Key":3,

"Value":"w"

}

]

}

### DeleteFolder – Deletes a folder.

#### Method: POST /DeleteFolder

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 FolderID;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"DSNine",

"FolderID":159

}

### DeleteDocument – Deletes a document.

#### Method: POST /DeleteDocument

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"dsnine",

"DocumentNumber":3,

"DocumentVersion":3

}

### AddDocumentToFolder – Adds a document to a specified folder.

#### Method: POST /AddDocumentToFolder

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

Int32 FolderID;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"dsnine",

"DocumentNumber":3,

"DocumentVersion":3,

"FolderID":3

}

### RemoveDocumentFromFolder – Remove a document from a specified folder.

#### Method: POST /RemoveDocumentFromFolder

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

Int32 FolderID;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"dsnine",

"DocumentNumber":3,

"DocumentVersion":3,

"FolderID":3

}

### IsOperationAllowed – Check whether an Operation is allowed for an user.

#### Method: POST /IsOperationAllowed

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 [OperationKind](#_OperationKind_1);

Int32 TargetID;

Int32 TargetID2;

array [ Int32, {0,} ] Operations;

}\*

* **OperationKind** - This determine whether it is looking at the folder contents operations (e.g., add document), the document operation (e.g., update), etc.
* **TargetID -**  if the target is a folder operation, this is the folder ID. If the target is a document, this is the document number.
* **TargetID2** - is a second ID if needed. For folder operations, this may be ignored or absent. For documents, this is the version number.
* **Operations** - parameter a collection of the appropriate enumerators as listed in the appendix. These are the operations the user is looking for. If not set or empty, then all operations will be evaluated.
* The **Operations** - return array is an array of key/value pairs where the key is the appropriate operation as defined in the appendix and the value is 0 or 1 depending on whether it is false or true.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

Dictionary<Int32, Boolean> Operations;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"dsnine",

"OperationKind":2,

"TargetID":3,

"TargetID2":1,

}

### UnlockDocument – Unlock a ckecked-out document.

#### Method: POST /UnlockDocument

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Int32 DocumentNumber;

Int32 DocumentVersion;

}\*

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

Boolean IsUnlocked;

}\*

* **IsUnlocked –** Whether the document got Unlocked or Not.

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"test1",

"userID":"test1"

},

"TimeZoneOffset":0,

"Version":2,

"DatabaseName":"DsNine",

"DocumentNumber":8518,

"DocumentVersion":4

}

### Security Update Jobs

#### Discussion:

The /updateJob call is used to create, modify, query and delete update jobs.

An update job is used to update the security of an object, including ACLs and default security. The job is a long-running job and while creation is an atomic operation, completion of the job is asynchronous. It is expected that jobs will be created with the POST form, queried or listed with the GET form, and deleted with the DELETE form (once it has completed). Additionally the PUT form can be used to cancel or restart a job via. its action=cancel, action=restart argument.

Headers:

The call must include the header Accept: application/json or a wider set of types.

**Base URL:** dms/v1

#### Method: GET

GET /updateJob

?dbName=value

&jobID=*value*

Parameters:

* dbName. Required. The job is created in the given database.
* jobID. Required. Along with the database information, the jobID determines the exact update job to target in the database. If jobID is not included in the GET form, all jobs are returned.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string JobID;

string JobDesc;

string JobStatus;

integer DocumentsUpdated;

integer DocumentsFailedToUpdate;

string JobFailureReason;

string ErrorQuery;

}\*

Where,

**JobStatus – value of** [**JobStatus**](#_JobStatus)

**JobFailureReason** – value of [JobFailureReason](#_JobFailureReason)

#### Method: POST

POST /updateJob

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

Dictionary < String, String > ProfileSearchCriteria;

Dictionary < String, String > FolderSearchCriteria;

Dictionary < String, String > AclUserValues;

Dictionary < String, String > AclGroupValues;

String JobType;

String JobTarget;

String JobDesc;

Boolean Recurse;

String DefaultSecurity;

Boolean SkipNonInheritedFolders;

Boolean UpdateInherited;

}\*

Where,

ProfileSearchCriteria – is a dictionary of Key as <[ProfileFieldEnumerators](#_ProfileFieldEnumerators(Paging))> and their Values

FolderSearchCriteria – is a dictionary of Key as <[FolderFieldEnumerators](#_FolderFieldEnumerators)> and their Values

AclUsersValues – is a dictionary of Key as Worksite User ID and Values as <[AccessRight](#_AccessRight)>

AclGroupValues – is a dictionary of Key Worksite Group Name and Values as <[AccessRight](#_AccessRight)>

JobType – is a value of [JobType](#_JobType)

JobTarget – is a value of [JobTarget](#_JobTarget)

DefaultSeurity – is a value of [DefaultSecurity](#_DefaultSecurity)

Recurse – true or false, Will recurse through the sub-folders only if the Security Update is specified for a Single FolderID.

SkipNonInheritedFolders – true or false, based on the value it would Skip the Non-Inherited Subfolders. This is effective only when Recurse os true.

UpdateInherited – true or false, based on the value it would Update the Folders that are Inherited. If any Inherited folder is selected for update job, then the folder will become a Non-Inherited one. This is effective only when Recurse is true. This is applicable only to Folder Jobs.

In all other search cases Recurse will always be false.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

integer JobID;

}\*

* JobID **–** The Bulk Security Update Job ID for further reference.

##### Example Request JSON:

{

"AclUserValues":[{

"Key":"server\_dev\_dan",

"Value":"RightRead"

},

{

"Key":"server\_dev\_ryan",

"Value":"RightReadWrite"

}],

"AclGroupValues":[{

"Key":"NRTAdmin",

"Value":"RightRead"

}],

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"Password2",

"userID":"server\_dev\_admin"

},

"DatabaseName":"DFTestDB1",

"JobTarget":"Documents",

"JobType":"Add",

"ProfileSearchCriteria":[{

"Key":"num",

"Value":"2844"

},

{

"Key":"ver",

"Value":"2"

}],

"DefaultSecurity":"View",

"TimeZoneOffset":0,

"Version":2

}

**Body:**

No body is passed in the GET version of the call.

No body is passed in the PUT call.

No body is passed in the DELETE call.

The body passed in the POST call.

Return:

Header Content-Type: application/json

#### Method: PUT

PUT /updateJob

?dbName=value

&jobID=*value*

*&action=value*

Parameters:

* dbName. Required. The job is created in the given database.
* jobID. Required. Along with the database information, the jobID determines the exact update job to target in the database. If jobID is not included in the GET form, all jobs are returned.
* action. Required. Specify whether the action to perform is cance or restart job.

Holds any one of the following vaues,

action

{

cancel,

restart

}

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string JobID;

string JobStatus;

string JobFailureReason;

}\*

Where,

**JobStatus –** is not used currently

**JobFailureReason** – value of [JobFailureReason](#_JobFailureReason)

### Security Update Jobs List

Discussion:

The /updateJobList returns the Job List, according to their status

**Base URL:** dms/v1

#### Method: GET

GET /updateJobList

?dbName=value

&jobID=*value*

&jobStatus=value

&pageLength=value

Parameters:

* dbName. Required. The job is created in the given database.
* jobID.Optional. Along with the database information, the jobID determines the exact update job to target in the database. If jobID is not included in the GET form, all jobs are returned.

JobID can contain more than one job separated with comma.

* jobStatus.Optional.

jobStatus is used to restrict the search according to status.

JobStatus can have one or more valuesof [**JobStatus**](#_JobType) enum values.

If jobStatus is not included, **StatusAll** is the the default values.

* pageLength.Optional.

Set the number of items to be returned in a page.

Default can be set in *“%installationDir%/Interwoven\WorkSite\WorkSitePro\Config\ DmsSettings.config”*

Add, *“<add key="DMS.PaginatedSearchPageLength" value="25" />”* inside <appSettings> tag.

If this is not mentioned the default will be 25.

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

String cursor;

Boolean more;

array [[**UpdateJobDetails**](#_UpdateJobDetails:), {0,} ] JobList;

}\*

### Documents

#### Method: POST

POST /contents

**Base URL:** dms/v1

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DBName;

Dictionary < String, String > ProfileSearchCriteria;

String Container;

Boolean Recurse;

String Filter;

String AddlFields;

Boolean AllVersions;

String FullText;

String FullTextLang;

String Cursor;

Int32 PageLength;

}\*

Eg:

http://localhost/Mobility2/MobilityService.svc/dms/v1/contents?

{

"Authentication":{

"userID":"test1",

"password":"test1",

"domain":"",

"appUuid":"e069fc1d-e1b8-4414-a3dc-d584f2b9b5fc"

},

"Version":2,

"TimeZoneOffset":0,

"DBName":"dsNine",

"Container":"3",

"Recurse":false,

"Filter":"email",

"AddlFields":"lastUser",

"AllVersions":false,

"FullText":"",

"FullTextLang":"",

"Cursor":"",

"PageLength":0,

"ProfileSearchCriteria":

[{

"Key":"desc",

"Value":"\*"

}],

}

#### Method: GET

**Base URL:** dms/v1

GET /contents

?dbName=*value*

&container=*value*

&recurse=*bool*

&filter=*ContentTypes*

&cursor=*value*

&*addlFields*=*profileFields*

&allVersions=*bool*

&fullText=*value*

&fullTextLang=*value*

&*profileCriteria*

&pageLength=value

#### Discussion

The /contents call is the general form for returning a set of documents and other contents from a folder or database. Examples are search, contents of a folder, or searching under a folder, as well as the worklist.

Not all requests are representable as GET calls due to URL length restrictions. A semantically-incorrect POST call will be provided with the arguments encoded as a JSON body, but the GET call is the preferred mechanism.

Calls including *profileCriteria* constitute searches. Documents are selected based on the field values passed in.

#### Variants

##### Search

A **search** exists when fullText or *profileCriteria* values are specified. If and only if a fullText value is provided a fullTextLang value may be. A search may specify a filter to limit the kinds of documents returned.

##### Container contents

If a container is specified but the call is not a search, then the call is for the contents of a container (worklist, folder, and so forth). The recurse, filter, fullText, fullTextLang and profileCriteria are not allowed.

##### Folder contents

If a container contents request includes a numeric folderID as the container, the call is for the contents of a folder. It behaves as a container contents call. All requirements of a Container contents call must be met but the dbName is required.

##### Folder contents search

If a container is specified and fullText or profileCriteria are included, the call is a folder contents search. The container must be a numeric folderID and the dbName must be specified. recurse is optional with a default value of true. Filter, cursor and allVersions may be are optional with default values of everything, no cursor and false, respectively.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbName. Optional. When included, the return contents are limited in scope to the named database. If not included, the whole DMS is searched. *DMS-wide searches are not currently available.*
* container. Optional. When included, the returned contents are limited in scope to the container specified. It is an error to include container when dbName is not set.
* recurse. Optional. If set true, contents from all subsidiary containers are included. If set false, only the current container is used. The recurse flag may only be set true when a container and *profileCriteria* are specified.
* filter. Optional. When included, the contents returned are limited in type to the listed ContentTypes. It is an error to include ContentShortcut in the list, and ContentShortcut objects are only returned in non-search circumstances. It is an error to set filter when *profileCriteria* are not specified (i.e., for non-searches).
* cursor. Optional. If set, begin the search a the passed cursor. See “Cursors, length, and pagination.” If not set, start from beginning. Only allowed when profileCriteria are specified.
* addlFields. Optional. A comma separated list of profile fields to return in addition to the default. See “Profile fields.” Note that accessTime may not be requested.
* allVersions. Optional. When true, returns all versions of the contents selected. When false, returns only the latest versions.
* fullText. Optional. A full text parameter to search. *profileCriteria* are considered to be set if fullText is specified (that is, it makes the request a search).
* fullTextLang. Optional. An identifier for the language used to index the document. Only allowed if fullText is specified.
* pageLength.Optional.

Set the number of items to be returned in a page.

Default can be set in *“%installationDir%/Interwoven\WorkSite\WorkSitePro\Config\ DmsSettings.config”*

Add, *“<add key="DMS.PaginatedSearchPageLength" value="25" />”* inside <appSettings> tag.

If this is not mentioned the default will be 25.

* *profileCriteria*. Optional. Not a specific parameter, but a set of parameters. The names of these parameters are taking from the values of **ProfileFieldEnumerators**, excepting db, accessTime and other non-searchable fields. The values are the strings to be searched. *profileCriteria* are considered to be set if fullText is set.

#### Example:

Doc in Container:

dbName=DFTestDB1&container=1518

dbName=DFTestDB1&container=1518&cursor=2844;1

Search:

Date Range Search:

Within two days including both: dbName=dsNineOne&editDate=1422748800,1423094400

On or before a date: dbName=dsNineOne&editDate=,1423094400

On or after a date: dbName=dsNineOne&editDate=1423094400

dbName=DFTestDB1&author=server\_dev\_admin,new\_user&desc=test&addlFields=lastUser,inUseBy

dbName=DFTestDB1&container=2585&filter=document&c1=CUSTOM1&c2=CUSTOM2

#### Body

No body is passed in the GET version of the call.

The POST version of the call’s body is TBD.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string cursor;

boolean more;

array [Content, (0,}] Contents;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

### Folders

GET /folders

?dbName=*value*

&container=*value*

&recurse=*bool*

&filter=*FolderTypes*

&cursor=*value*

&*profileCriteria*

*&folderCriteria*

&pageLength=value

**Base URL:** dms/v1

#### Discussion

The /folders call is the general form for returning a set of folders from a folder or database, either by a search or from a folder.

The call permits various scope restrictions and, depending on the arguments may simply return the subfolders of a folder or may perform a container-based or containerless search.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbName. Required. The returned folders are limited in scope to the named database
* filter. Required. When included, the folders returned are limited in type to the listed FolderTypes. Some folder types are mutually exclusive.
* container. Optional. When included, the returned folders are limited in scope to the container specified. This argument requires a dbName be set, container search is restricted to Workspace Shortcut search
* recurse. Optional. If set true, contents from all subsidiary containers are included. If set false, only the current container is used. This argument requires container to be set.

This is not used currently.

* cursor. Optional. If set, begin the search a the passed cursor. See “Cursors, length, and pagination.” If not set, start from beginning.
* profileCriteria. Optional. Not a specific parameter, but a set of parameters. The names of these parameters are taking from the values of **ProfileFieldEnumerators**, excepting db, accessTime and other non-searchable fields. The values are the strings to be searched. These are only allowed if the filter contains only a workspace value.
* folderCriteria. Optional. Not a specific parameter, but a set of parameters from the **FolderFieldEnumerators**. These are search criteria for a search.
* pageLength.Optional.

Set the number of items to be returned in a page.

Default can be set in *“%installationDir%/Interwoven\WorkSite\WorkSitePro\Config\ DmsSettings.config”*

Add, *“<add key="DMS.PaginatedSearchPageLength" value="25" />”* inside <appSettings> tag.

If this is not mentioned the default will be 25.

#### Example

dbName=DSNine&container=3

Search:

dbName=DSNine&filter=folder&foldername=df

dbName=DSNine&filter=workspace&FolderName=\*ws\*

dbName=DSNine&filter=workspace&FolderName=\*ws\*&cursor=2648

dbName=DFTestDB1&key=iMan\_\_\_5&value=server\_dev\_admin

dbName=DFTestDB1&filter=workspace&author=server\_dev\_admin

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string cursor;

boolean more;

array [Folder, (0,}] Folders;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

### SavedSearchFolderDetails

GET /savedSearchDetails

?dbName=*value*

&folderID=*value*

**Base URL:** dms/v1

#### Discussion

The /savedSearchDetails call is the general form for returning the folder details of the Saved Search Folders

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbName. Required. The returned folders are limited in scope to the named database
* folderID. Required. Folder ID of the Saved Search Folder

#### Example

dbName=dsNine&folderID=3678

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string cursor;

boolean more;

array [SearchFolder, (0,}] Folders;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

**Example Response Json:**

{

"ServerVersion":3,

"statusCode":0,

"statusDetail":"Success",

"AddlProps":null,

"Contents":[],

"ContentsOp":null,

"DBName":"DSNine",

"DefaultSecurity":73,

"Description":"",

"EditTime":1430939687,

"FolderID":2705,

"FolderKind":13,

"Inherited":1,

"Name":"Saved Search",

"ParentFolderID":2,

"Profile":null,

"Subfolders":[],

"Target":null,

"TargetDatabaseName":"",

"TargetFolderID":0,

"TargetFolderKind":3,

"FolderSearchCriteria":[

{"Key":"folderid","Value":"2"}

],

"ProfileSearchCriteria":[

{"Key":"desc","Value":"test\*"},

{"Key":"type","Value":"ANSI"},

{"Key":"systemcreatedate","Value":"1322697600,1430870400"},

{"Key":"containerid","Value":"2"},

{"Key":"subtype","Value":"D,5"}

],

"Query":""

}

### Groups

#### Method: GET

GET /groups

?dbName=*value*

&groupID=value

&groupFullname=*value*

&cursor=*value*

&userId=*value*

&pageLength=value

**Base URL:** dms/v1

Discussion

The /groups call is the general form for returning a set of groups. The groups may be based on a search of groups or may be enumerated as a user’s groups.

Headers

The call must include the header Accept: application/json or a wider set of types.

Parameters

* dbName. Required. The returned groups are limited in scope to the named database
* groupID. Optional. Group ID
* groupFullname. Optional. Full name of the group
* userId. Optional. Group to which the user belongs
* cursor. Optional. If set, begin the search at the passed cursor. See “Cursors, length, and pagination.” If not set, start from beginning.
* pageLength.Optional.

Set the number of items to be returned in a page.

Default can be set in *“%installationDir%/Interwoven\WorkSite\WorkSitePro\Config\ DmsSettings.config”*

Add, *“<add key="DMS.PaginatedSearchPageLength" value="25" />”* inside <appSettings> tag.

If this is not mentioned the default will be 25.

Example

dbName=DFTestDB1&groupID=1&groupFullname=\*nrt\*&userId=server\_dev\_admin

Body

No body is passed in the GET version of the call.

Return

Header Content-Type: application/json

##### Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string cursor;

boolean more;

array [[Group](#_Group:), (0,}] Groups;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

#### Method: POST

POST /groups

Only an Admin user could perform this operation.

##### Request JSON:

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

String DomainName;

Boolean Enabled;

Boolean External;

String FullName;

String Name;

String NOS;

}\*

Where,

Name – Name of the Group

FullName – Full Name of the Group

DomainName – Domain to which the Group belongs

Enabled – is the Group Enabled

External – is the Group an External Group

NOS – is a value of [NOS](#_NOS)

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

[Group](#_Group:) Group;

}\*

##### Example Request JSON:

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"",

"userID":""

},

"DatabaseName":"DFTestDB1",

"DomainName":"",

"Enabled":true,

"External":false,

"FullName":"New Test Group FullName",

"Name":"New Test Group",

"NOS":”NOSVirtual”,

"TimeZoneOffset":0,

"Version":3

}

**Body:**

No body is passed in the GET version of the call.

The body passed in the POST call.

Return:

Header Content-Type: application/json

#### Method: PUT

PUT /groups

?dbName=value

&groupName=*value*

*&action=value*

*&user=value*

Only an Admin user could perform this operation.

Parameters:

* dbName. Required. The job is created in the given database.
* groupName. Required. Name of the Group to edit
* action. Required. Specify what action to perform

Holds any one of the following vaues,

action

{

addUser,

deleteUser,

disableGroup,

enableGroup

}

##### Response JSON:

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

[Group](#_Group:) Group;

}\*

##### Example:

/dms/v1/groups?dbName=dftestdb1&groupName=GN111&action=addUser&user=server\_dev\_admin

/dms/v1/groups?dbName=dftestdb1&groupName=GN111&action=disableGroup

### History (only Mock Data available, paginated not available yet)

GET /history

?dbName=*value*

&historyCriteria=*historyCriteria*

&cursor=*value*

#### Discussion

The /history call is the general form for returning history entries. The entries are based on a search, which may simply be a document’s number and version.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbName. Required. The returned groups are limited in scope to the named database
* historyCriteria. Optional. Not a specific parameter, but a set of parameters, the names of these parameters taken from the values of HistoryFieldEnumerators, excepting db. These are used in a search-like manner.
* cursor. Optional. If set, begin the search at the passed cursor. See “Cursors, length, and pagination.” If not set, start from beginning.

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string cursor;

boolean more;

array [HistoryEntry, (0,}] HistoryEntries;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

### User

GET /users

?dbName=*value*

&userCriteria=*userCriteria*

&groupName=*value*

&cursor=*value*

&pageLength=value

**Base URL:** dms/v1

#### Discussion

The /users call is the general form for returning a set of users. The users may be based on a search of users or may be enumerated as a group’s users.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbName. Required. The returned groups are limited in scope to the named database
* userCriteria. Optional. Not a specific parameter, but a set of parameters, the names of these parameters taken from the values of **UserFieldEnumerators**, excepting db. These are used in a search-like manner. If not set, all users under the database or user are returned.
* groupName. Optional. If set, the users are limited in scope to those to which the specified group contains.
* cursor. Optional. If set, begin the search at the passed cursor. See “Cursors, length, and pagination.” If not set, start from beginning.
* pageLength.Optional.

Set the number of items to be returned in a page.

Default can be set in *“%installationDir%/Interwoven\WorkSite\WorkSitePro\Config\ DmsSettings.config”*

Add, *“<add key="DMS.PaginatedSearchPageLength" value="25" />”* inside <appSettings> tag.

If this is not mentioned the default will be 25.

#### Example

dbName=DFTestDB1&UserName=\*server\*

dbName=DFTestDB1&groupNum=1

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string cursor;

boolean more;

array [User, (0,}] Users;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

### createSubscriptionFolders

GET /createSubscriptionFolders

?userID=*value*

**Base URL:** dms/v1

#### Discussion

The /createSubscriptionFolders call creates MyMatters and MyFavourites folder for an user.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* userID. Required. The Worksite User name.

#### Example

userID=test1

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

string MyFavouritesFolderID;

string MyMattersFolderID;

}\*

Most of these values are conventional values (see “Return values”) or are self-explanatory.

### documentHistory

GET /documentHistory

?dbList=*value*

&userID= *value*

&fromDate=*time*

&toDate=*time*

&cursor=value

&pageLength=value

**Base URL:** dms/v1

#### Discussion

The /documentHistory call return a list of Document History entries within a specified time.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbList. Required. One or more database names separated by comma.
* userID. Required. Name of the Worksite user
* fromDate. Required. Date Time value specified in Seconds elapsed from Epoch Time
* toDate. Required. Date Time value specified in Seconds elapsed from Epoch Time
* cursor. Required. Cursor value to start the search from
* pageLength.Optional.

Set the number of items to be returned in a page.

Default can be set in *“%installationDir%/Interwoven\WorkSite\WorkSitePro\Config\ DmsSettings.config”*

Add, *“<add key="DMS.PaginatedSearchPageLength" value="25" />”* inside <appSettings> tag.

If this is not mentioned the default will be 25.

#### Example

#### dbList=dftestdb1&userID=server\_dev\_admin&fromDate=1404234180&toDate=1407862980

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} ServerVersion;

string Cursor;

boolean More;

array [[HistoryEntry](#_HistoryEntry:), (0,}] Users;

}\*

### checkedOutDocumentsForUser

GET /checkedOutDocumentsForUser

?dbList=*value*

&userID= *value*

**Base URL:** dms/v1

#### Discussion

The /checkedOutDocumentsForUser call returns all the Documents that are checked out by an user.

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbList. One or more database names separated by comma. If this is not provided the search will be for all the databases
* userID. Required. Name of the Worksite user

#### Example

#### dbList=dftestdb1,dftestdb2&userID=server\_dev\_admin

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} ServerVersion;

array [[Profile](#_Profile:), (0,}] Profiles;

}\*

### userAccessibleDocuments

**Base URL:** dms/v1

A set of document numbers and a user ID (user name) are passed, and this function would return the subset of those documents numbers to which the specified user would have access. If the user does not have at least read access to the document, then the document number would not be returned in the subset.

This could be called only by an Admin user.

#### Method: POST

POST /userAccessibleDocuments

**Request JSON:**

object {

[AuthenticationInfo](#_AuthenticationInfo) Authentication;

Int32 Version;

Double TimeZoneOffset;

String DatabaseName;

array [[Document](#_Document:), (0,}] Documents;

String UserID;

}\*

Where,

Documents – is an array of [Document](#_Document:) objects; this could have a set of document number and versions on which to do the accessibility test

UserID – the Worksite UserID for which to check the accessible documents.

**Response JSON:**

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} serverVersion;

array [[Document](#_Document:), (0,}] Documents;

}\*

* Documents **–** the subset of documents that the user has atleast read access to.

**Example Request JSON:**

{

"Authentication":{

"appUuid":"8129A423-5B7F-44CA-9541-B756041A8B72",

"domain":"",

"password":"Password2",

"userID":"server\_dev\_admin"

},

"DatabaseName":"DFTestDB1",

"Documents":[{

"Number":7692,

"Version":2

},

{

"Number":4102,

"Version":1

},

{

"Number":2844,

"Version":2

}],

"TimeZoneOffset":0,

"UserID":"server\_dev\_amin",

"Version":3

}

Return:

Header Content-Type: application/json

### savedSearchDetails

#### Method: GET

GET /savedSearchDetails

**Base URL:** dms/v1

#### Discussion

Given the folderID of a Saved Search Folder this API will return the Search Criteria of the folder

#### Headers

The call must include the header Accept: application/json or a wider set of types.

#### Parameters

* dbName. Required. Database Name
* folderID. Required. FolderID of the Saved Search Folder

#### Example

#### dbName=databaseOne&folderID=12345

#### Body

No body is passed in the GET version of the call.

#### Return

Header Content-Type: application/json

Response JSON is of the format

object {

integer {0,} statusCode;

string statusDetail;

integer {0,} ServerVersion;

object {[SearchFolder](#_SearchFolder:)}\*;

}\*

Example:

{

… same as folder..+

"FolderSearchCriteria":[

{"Key":"folderid","Value":"2"}

],

"ProfileSearchCriteria":[

{"Key":"desc","Value":"test\*"},

{"Key":"type","Value":"ANSI"},

{"Key":"systemcreatedate","Value":"1322697600,1430870400"},

{"Key":"containerid","Value":"2"},

{"Key":"subtype","Value":"D,5"}

],

"Query":""

“SearchEmail”:”SearchEmailOnly”

}