DOKMEE DMS CONNECTOR v.1.5

1. PURPOSE

- Provide a simple programming interface for Dokmee DMS system integration
- Required only a single reference DLL (Dokmee.Dms.Connector)

2. HOW TO INITIALIZE CONNECTOR

- **DMS** - DmsConnector connector = **new** DmsConnector(DokmeeApplication.DokmeeDMS);
- **WEB** - DmsConnector connector = **new** DmsConnector(DokmeeApplication.DokmeeWeb);
- **CLOUD** - DmsConnector connector = **new** DmsConnector(DokmeeApplication.DokmeeCloud);

3. HOW TO GET EXISITING CONNECTION FROM CONFIG

- **DMS** - var connection = connector. GetConnectionFromConfig <ConnectionInfo>();
- **WEB** - var connection = connector.GetConnectionFromConfig<string>();
- **CLOUD** – Cloud connection is embedded

4. HOW TO SAVE CONNECTION TO CONFIG

- **DMS** - connector. SaveConnectionToConfig <ConnectionInfo>(**new** ConnectionInfo { });
- **WEB** - connector. SaveConnectionToConfig <string>(“http://localhost/dokmee”);
- **CLOUD** – Cloud connection is embedded so not required to save connection

5. HOW TO REGISTER CONNECTION

- **DMS**
  ```
  ConnectionInfo connInfo = new ConnectionInfo ();
  connInfo.ServerName = "localhost";
  connInfo.UserID = "sa";
  connInfo.Password = "@password1";
  connector.RegisterConnection< ConnectionInfo >(connInfo);
  ```

- **WEB** - connector.RegisterConnection<string>(“http://localhost/dokmee”);
- **CLOUD** – Cloud connection is registered during initialization

6. HOW TO RETRIEVE DOKMEE FILE CABINETS

- **DMS/WEB/CLOUD**

```
var cabinets = connector.Login(new LogonInfo { Username = "admin", Password = "admin" });
```

7. HOW TO INITIALIZE A SPECIFIC FILE CABINET

- **DMS/WEB/CLOUD**

```
connector.RegisterCabinet(cabinetID);
```

8. HOW TO RETRIEVE FOLDER STRUCTURE

- **DMS/WEB/CLOUD**

    I.   Folder @ Root Level:
    ```
    var nodes = connector.GetFilesystem(WebAccess.Data.SubjectTypes.Folder);
    ```

    II.   Folder @ Sub Folder Level:
    ```
    var nodes = connector.GetFilesystem(WebAccess.Data.SubjectTypes.Folder, nodeID);
    ```

9. HOW TO RETRIEVE FOLDER STRUCTURE WITH PAGING

- **DMS/WEB/CLOUD**

    III.   Folder @ Root Level:
    ```
    var nodes = connector. GetFilesystemPaging (WebAccess.Data.SubjectTypes.Folder);
    ```

    IV.   Folder @ Sub Folder Level:
    ```
    var nodes = connector. GetFilesystemPaging (WebAccess.Data.SubjectTypes.Folder, nodeID);
    ```

10. HOW TO CREATE NEW FOLDER

- **DMS/WEB/CLOUD**

```
var newFolder = connector.CreateFolder("SAMPLE", parentFolderID);
```

11. HOW TO GET FILE INDEX INFO

- **DMS/WEB/CLOUD**

```
var indexInfos = connector.GetCabinetIndexInfo();
```

12. HOW TO UPLOAD SINGLE FILE

- **DMS/WEB/CLOUD**

```
var uploadInfo = new UploadInfo { Filename = filePath, IndexInfos = BuildIndexInfos(connector, indexInfos) };
```

```
WebAccess.Data.FileDesc fileDesc = connector.UploadFile(uploadFolder.ID, uploadInfo)
```

13. HOW TO CHECK FOR RIGHTS TO SPECIFIC NODE

- **DMS/WEB/CLOUD**

```
bool isAuthorized = connector.HasRights(DmsRights.Import, uploadFolder.ID.ToString());
```

14. HOW TO VIEW FILE

- **DMS/WEB/CLOUD**

```
var viewer = connector.ViewFile(node.ID.ToString());
Process.Start(viewer);
```

15. HOW TO RENAME A FILE OR FOLDER

- **DMS/WEB/CLOUD**

```
connector.Rename(node.ID.ToString(), "My Test");
```

16. HOW TO DELETE A FILE OR FOLDER

- **DMS/WEB/CLOUD**

```
connector.DeleteNode(node.ID.ToString());
```

17. HOW TO SEARCH A SPECIFIC NODE OR BY SEARCH FIELD TYPE

- **DMS/WEB/CLOUD**

```
// search specific node by id
var node = connector.Search(WebAccess.Data.SearchFieldType.ByNodeID, "413e682b-bf0b-477d-8a65-
2b4254b9b444");

// search by all index
var dmsNodes = connector.Search(WebAccess.Data.SearchFieldType.AllIndex, "doc");
```

18. HOW TO UPDATE INDEX

- **DMS/WEB/CLOUD**

```
// build index
Enumerable<DokmeeIndex> indexes = BuildIndexInfo();
connector.UpdateIndex(nodeID, indexes);

// example
private IEnumerable<DokmeeIndex> BuildIndexInfos(DmsConnector dmsConnector,
                                                 IEnumerable<DokmeeIndex> indexInfos)
{
    var dokmeeIndex = new List<DokmeeIndex>();
    foreach (var i in indexInfos)
    {
        var index = new DokmeeIndex();
```

```
            index.Name = i.Name;
            if (dmsConnector.IsStringType(i.ValueType) && i.IsDefault)
                index.Value = "";
            if (dmsConnector.IsStringType(i.ValueType) && !i.IsDefault)
                index.Value = GetRandomName();
            else if (dmsConnector.IsNumberType(i.ValueType))
                index.Value = GetRandomNumber();
            else if (dmsConnector.IsDateType(i.ValueType))
                index.Value = DateTime.Now.ToString();

            dokmeeIndex.Add(index);
        }

        return dokmeeIndex.AsEnumerable();
    }
```

19. HOW TO MOVE A FILE

- **DMS/WEB/CLOUD**

```
connector.MoveFile(fileID, folderID);
```

20. HOW TO CANCEL UPLOAD

- **DMS/WEB/CLOUD**

```
Connector.CancelUpload();
```

21. HOW TO LOGOUT

- **DMS/WEB/CLOUD**

```
Connector.logout();
```

22. HOW TO GET FILE CABINET INDEX INFO BY CABINET ID

- **DMS/WEB/CLOUD**

```
Connector.GetCabinetIndexInfoByID(cabinetID, subjectType);

// Available subject types
public enum SubjectTypes
{
    Document = 1,
    Folder = 2,
    All = 255,

}
```