**壹键通教育云盘服务平台项目**

**代**

**码**

**段**

**东方博冠（北京）科技有限公司**

**一、数据库底层操作类**

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BLL.YunFile

{

public class ImgList

{

BLL.YunFile.MyFile MyFile = new BLL.YunFile.MyFile();

public List<DateList> ImgDateList( string UserCode )

{

DataSetToList List = new DataSetToList();

using( var db = new Model.NETDISKDBEntities() )

{

string strSql = "exec [SpFileCreateAxis] @CreateId='" + UserCode + "'";

DataSet Dt = MyFile.SqlQueryForDataTatable1( db.Database, strSql );

List<DateList> ImgDateInfolist = List.ToList<DateList>( Dt, 0 );

return ImgDateInfolist.OrderByDescending( a => a.YearS ).ToList();

}

}

public List<Model.YUN\_FileInfo> ImgTitleInfoList( string UserCode,string time )

{

DataSetToList List = new DataSetToList();

using( var db = new Model.NETDISKDBEntities() )

{

string strSql = "SELECT DISTINCT(CONVERT(varchar(10),(SELECT DATEPART(yyyy, FileCreateTime)))+'年'+ CONVERT(varchar(10),(SELECT DATEPART(mm, FileCreateTime)))+'月'+ CONVERT(varchar(10),(SELECT DATEPART(dd, FileCreateTime)))+'日') AS CreateUnitCode,CONVERT(varchar(10),(select COUNT(\*) from YUN\_FileInfo where CONVERT(varchar(10),a.FileCreateTime, 23)=CONVERT(varchar(10), FileCreateTime, 23) and CreateId = '" + UserCode + "' AND FileExtName IN ('.bmp', '.jpeg', '.jpg', '.gif', '.png', '.tif', '.psd', '.dwg') and FileState=1 ))as ShareFileID FROM dbo.YUN\_FileInfo as a WHERE CreateId = '" + UserCode + "' AND FileExtName IN ('.bmp', '.jpeg', '.jpg', '.gif', '.png', '.tif', '.psd', '.dwg') and FileState=1 AND datediff(month,FileCreateTime,'" + time + "')=0 ORDER BY CreateUnitCode DESC";

DataSet Dt = MyFile.SqlQueryForDataTatable1( db.Database, strSql );

List<Model.YUN\_FileInfo> Filelist = List.ToList<Model.YUN\_FileInfo>( Dt, 0 );

return Filelist;

}

}

public List<Model.YUN\_FileInfo> ImgInfoList( string UserCode, string time )

{

DataSetToList List = new DataSetToList();

using( var db = new Model.NETDISKDBEntities() )

{

string strSql = "SELECT FileId,FileName,FileCreateTime,FileUrl,FileState,(CONVERT(varchar(10),(SELECT DATEPART(yyyy, FileCreateTime)))+'年'+ CONVERT(varchar(10),(SELECT DATEPART(mm, FileCreateTime)))+'月'+ CONVERT(varchar(10),(SELECT DATEPART(dd, FileCreateTime)))+'日') AS CreateUnitCode FROM dbo.YUN\_FileInfo as a WHERE CreateId = '" + UserCode + "' AND FileExtName IN ('.bmp', '.jpeg', '.jpg', '.gif', '.png', '.tif', '.psd', '.dwg') and FileState=1 AND datediff(month,FileCreateTime,'" + time + "')=0 ORDER BY CreateUnitCode DESC";

DataSet Dt = MyFile.SqlQueryForDataTatable1( db.Database, strSql );

List<Model.YUN\_FileInfo> Filelist = List.ToList<Model.YUN\_FileInfo>( Dt, 0 );

return Filelist;

}

}

public class DateList

{

public string YearS { get; set; }

public string Moth { get; set; }

}

}

}

**二、我的文件数据库操作类**

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Data.SqlClient;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Web;

namespace BLL.YunFile

{

public class MyFile

{

DataSetToList List = new DataSetToList();

/// <summary>

/// 获取我的网盘数据

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> MyFileLoad( string FileId, string SearchName, string CreateId )

{

string wherestr = "";

if( !string.IsNullOrEmpty( SearchName ) )

{

wherestr = " and FileName like '%" + SearchName + "%'";

}

else

{

wherestr = " and ParentFileId='" + FileId + "'";

}

if( CreateId != null && CreateId != "" )

{

wherestr += " AND [CreateId] = '" + CreateId + "'";

}

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT \* From YUN\_FileInfo" );

strSql.Append( " where FileState=1 " + wherestr + " ORDER BY IsFolder DESC " );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

return result1.ToList();

}

}

/// <summary>

/// 获取学校分享数据

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> ShareFileLoad( string FileId, string SearchName )

{

string wherestr = "";

if( !string.IsNullOrEmpty( SearchName ) )

{

wherestr = " and FileName like '%" + SearchName + "%'";

}

else

{

wherestr = " and ParentFileId='" + FileId + "'";

}

wherestr += " AND ShareTypeId ='1001' and( CreateId='User' or CreateId='Admin') ";

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT \* From YUN\_FileInfo" );

strSql.Append( " where FileState=1 " + wherestr + " ORDER BY IsFolder DESC " );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

return result1.ToList();

}

}

/// <summary>

/// 获取部门分享数据

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> ShareDepartmentLoad( string FileId, string SearchName, string UnitCode )

{

string wherestr = "";

if( !string.IsNullOrEmpty( SearchName ) )

{

wherestr = " and FileName like '%" + SearchName + "%'";

}

wherestr += " AND ShareTypeId ='1002' and CreateUnitCode='" + UnitCode + "' and IsFolder =0";

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT \* From YUN\_FileInfo" );

strSql.Append( " where FileState=1 " + wherestr + " ORDER BY IsFolder DESC " );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

return result1.ToList();

}

}

/// <summary>

/// 分类获取我的网盘数据

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> MyFileShow( string wherestr )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT \* From YUN\_FileInfo" );

strSql.Append( " where FileState=1 and IsFolder=0 " + wherestr + " ORDER BY IsFolder DESC " );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

return result1.ToList();

}

}

/// <summary>

/// 获取我分享的数据

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> MyShareOrDelFileLoad( int pageIndex, int pageSize, string wherestr, string Ord )

{

int startRow = ( pageIndex - 1 ) \* pageSize;

StringBuilder strSql = new StringBuilder();

strSql.Append( " SELECT TOP " + pageSize + " \*,(SELECT ShareTypeName FROM dbo.ShareInfo WHERE ShareTypeId =A.ShareTypeId) as ShareTypeName From (SELECT ROW\_NUMBER() OVER (ORDER BY " + Ord + " desc) AS RowNumber,\* FROM YUN\_FileInfo where" + wherestr + " ) as A " );

strSql.Append( "WHERE RowNumber > " + startRow + " ORDER BY RowNumber ASC " );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DataSet Dt = SqlQueryForDataTatable1( Db.Database, strSql.ToString() );

List<Model.YUN\_FileInfo> result1 = List.ToList<Model.YUN\_FileInfo>( Dt, 0 );

return result1.ToList();

}

}

/// <summary>

/// EF SQL 语句返回 dataTable

/// </summary>

/// <param name="db"></param>

/// <param name="sql"></param>

/// <param name="parameters"></param>

/// <returns></returns>

public DataSet SqlQueryForDataTatable1( Database db, string sql )

{

SqlConnection conn = new System.Data.SqlClient.SqlConnection();

conn = (SqlConnection) db.Connection;

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

cmd.CommandText = sql;

SqlDataAdapter adapter = new SqlDataAdapter( cmd );

DataSet table = new DataSet();

adapter.Fill( table );

conn.Close();//连接需要关闭

conn.Dispose();

return table;

}

/// <summary>

/// 获取总个数

/// </summary>

/// <param name="sql"></param>

/// <returns></returns>

public int GetCount( string sql )

{

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<int> result2 = Db.Database.SqlQuery<int>( "SELECT count(\*) FROM YUN\_FileInfo where " + sql );

return result2.FirstOrDefault();

}

}

/// <summary>

/// 向下 获取文件

/// </summary>

/// <param name="field"></param>

/// <param name="Ids"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> GetFileByDown( string field, string Ids )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT " );

strSql.Append( " \*" );

strSql.Append( " FROM YUN\_FileInfo where " );

strSql.Append( string.Format( "{0} In ({1})", field, Ids ) );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

return result1.ToList();

}

}

/// <summary>

/// 递归向上 获取文件夹名

/// </summary>

/// <param name="parentFileId"></param>

/// <returns></returns>

public void GetFileMapPath( string parentFileId, bool IsShare, ref string path )

{

string isFile = "";

//空值和0 是我的网盘否是最顶部

switch( parentFileId )

{

case "":

case "0":

Cookie C = new Cookie();

if( IsShare )

{

path = "School/" + path;

}

else

{

path = C.GetUserCookie().userCode + "/" + path;

}

break;

default:

Model.YUN\_FileInfo YunFileList = GetFileByUp( "FileId", int.Parse( parentFileId ) );

if( path == "" )

{

path = YunFileList.FileId.ToString();

}

else

{

path = YunFileList.FileId.ToString() + "/" + path;

}

isFile = YunFileList.IsFolder.ToString();

if( isFile == "True" )

{

GetFileMapPath( YunFileList.ParentFileId, IsShare, ref path );

}

break;

}

}

/// <summary>

/// 向上 获取文件夹名

/// </summary>

/// <param name="field"></param>

/// <param name="Ids"></param>

/// <returns></returns>

public Model.YUN\_FileInfo GetFileByUp( string field, int Ids )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT " );

strSql.Append( " \*" );

strSql.Append( " FROM YUN\_FileInfo where " );

strSql.Append( string.Format( "{0} = {1} AND FileState=1 ", field, Ids ) );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

return result1.FirstOrDefault();

}

}

/// <summary>

/// 递归向上 获取文件夹名

/// </summary>

/// <param name="parentFileId"></param>

/// <returns></returns>

public void GetFileMapPathByDel( string parentFileId, bool IsShare, ref string path )

{

string isFile = "";

//空值和0 是我的网盘否是最顶部

switch( parentFileId )

{

case "":

case "0":

Cookie C = new Cookie();

if( IsShare )

{

path = "School/" + path;

}

else

{

path = C.GetUserCookie().userCode + "/" + path;

}

break;

default:

Model.YUN\_FileInfo YunFileList = GetModel( int.Parse( parentFileId ) );

if( path == "" )

{

path = YunFileList.FileId.ToString();

}

else

{

path = YunFileList.FileId.ToString() + "/" + path;

}

isFile = YunFileList.IsFolder.ToString();

if( isFile == "True" )

{

GetFileMapPathByDel( YunFileList.ParentFileId, IsShare, ref path );

}

break;

}

}

/// <summary>

/// 获取文件夹(复制文件/剪切文件)

/// </summary>

/// <param name="field"></param>

/// <param name="Ids"></param>

/// <returns></returns>

public DataTable GetFileList( string ParentFileId, string CreateId, bool Share )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT " );

if( Share )

{

strSql.Append( string.Format( " \*,(SELECT COUNT(\*) FROM dbo.YUN\_FileInfo WHERE ShareTypeId='1001' and IsFolder = 1 AND FileState = 1 AND ParentFileId = A.FileId and( CreateId='User' or CreateId='Admin') )as IsChildFolder", CreateId ) );

strSql.Append( " FROM dbo.YUN\_FileInfo AS A " );

strSql.Append( string.Format( "WHERE ShareTypeId='1001' and IsFolder = 1 AND FileState = 1 AND ParentFileId = '{0}'and( CreateId='User' or CreateId='Admin')", ParentFileId, CreateId ) );

}

else

{

strSql.Append( string.Format( " \*,(SELECT COUNT(\*) FROM dbo.YUN\_FileInfo WHERE CreateId = '{0}' AND IsFolder = 1 AND FileState = 1 AND ParentFileId = A.FileId )as IsChildFolder", CreateId ) );

strSql.Append( " FROM dbo.YUN\_FileInfo AS A " );

strSql.Append( string.Format( "WHERE CreateId ='{1}' and IsFolder = 1 AND FileState = 1 AND ParentFileId = '{0}'", ParentFileId, CreateId ) );

}

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DataSet Dt = List.SqlQueryForDataTatable( Db.Database, strSql.ToString() );

return Dt.Tables[0];

}

}

/// <summary>

/// 获取文件夹(复制文件/剪切文件)

/// </summary>

/// <param name="field"></param>

/// <param name="Ids"></param>

/// <returns></returns>

public List<Model.YUN\_FileInfo> GetFileListByFileIds( string FileIdS )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT " );

strSql.Append( " \* " );

strSql.Append( " FROM dbo.YUN\_FileInfo AS A " );

strSql.Append( string.Format( "WHERE FileId in ({0})", FileIdS ) );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DataSet Dt = List.SqlQueryForDataTatable( Db.Database, strSql.ToString() );

List<Model.YUN\_FileInfo> result1 = List.ToList<Model.YUN\_FileInfo>( Dt, 0 );

return result1;

}

}

/// <summary>

/// 文件删除

/// </summary>

/// <param name="path"></param>

public void FileDelete( string path )

{

//File.Delete(path);

}

/// <summary>

/// 文件删除

/// </summary>

/// <param name="path"></param>

public void DeleteFile( string path )

{

if( File.Exists( path ) )

{

File.Delete( path );

}

}

public void DeleteFolderFile( string path )

{

if( System.IO.Directory.Exists( path ) )

{

System.IO.Directory.Delete( path, true );

}

}

/// <summary>

/// 删除一条数据

/// </summary>

public bool Delete( int delId )

{

Model.YUN\_FileInfo F = GetModel( delId );

F.FileState = false;

F.FileDeleteTime = DateTime.Now;

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbEntityEntry<Model.YUN\_FileInfo> entry = Db.Entry<Model.YUN\_FileInfo>( F );

entry.State = System.Data.Entity.EntityState.Modified;

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 文件重命名

/// </summary>

public bool ReName( int FileId, string FileName )

{

Model.YUN\_FileInfo F = new Model.YUN\_FileInfo() { FileId = FileId, FileName = FileName };

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

Db.YUN\_FileInfo.Attach( F );

Db.Entry( F ).Property( x => x.FileName ).IsModified = true;

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 文件分享

/// </summary>

public bool Share( int FileId, string ShareTypeId, string unitCode )

{

Model.YUN\_FileInfo F = new Model.YUN\_FileInfo() { FileId = FileId, IsShare = true, ShareTypeId = ShareTypeId, CreateUnitCode = unitCode, ShareTime = DateTime.Now };

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

Db.YUN\_FileInfo.Attach( F );

Db.Entry( F ).Property( x => x.IsShare ).IsModified = true;

Db.Entry( F ).Property( x => x.ShareTypeId ).IsModified = true;

Db.Entry( F ).Property( x => x.CreateUnitCode ).IsModified = true;

Db.Entry( F ).Property( x => x.ShareTime ).IsModified = true;

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 文件取消分享

/// </summary>

public bool IsShare( int FileId )

{

Model.YUN\_FileInfo F = new Model.YUN\_FileInfo() { FileId = FileId, IsShare = false };

Model.YUN\_FileInfo FileInfo = GetModel( FileId );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

Db.YUN\_FileInfo.Attach( F );

Db.Entry( F ).Property( x => x.IsShare ).IsModified = true;

int rows = Db.SaveChanges();

if( FileInfo.IsFolder.ToString() == "True" )

{

string FileIdString = FileId.ToString();

var NextFile = from B in Db.YUN\_FileInfo

where B.ParentFileId == FileIdString

select B;

foreach( var item in NextFile.ToList() )

{

IsShare( item.FileId );

}

}

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 文件还原（递归还原文件夹）

/// </summary>

public bool Reduction( int FileId )

{

Model.YUN\_FileInfo FileInfo = GetModel( FileId );

if( !string.IsNullOrEmpty( FileInfo.ParentFileId ) )

{

Model.YUN\_FileInfo ParentFile = GetModel( int.Parse( FileInfo.ParentFileId ) );

if( ParentFile != null )

{

ReductionFlie( FileId );

if( FileInfo.ParentFileId != "" )

{

Reduction( int.Parse( FileInfo.ParentFileId ) );

}

return true;

}

else if( ParentFile == null )

{

ReductionFlieToMu( FileId );

return true;

}

else

{

return false;

}

}

else

{

if( ReductionFlie( FileId ) )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 文件还原

/// </summary>

public bool ReductionFlie( int FileId )

{

Model.YUN\_FileInfo F = new Model.YUN\_FileInfo() { FileId = FileId, FileState = true };

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

Db.YUN\_FileInfo.Attach( F );

Db.Entry( F ).Property( x => x.FileState ).IsModified = true;

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 文件还原到主目录

/// </summary>

public bool ReductionFlieToMu( int FileId )

{

Model.YUN\_FileInfo F = new Model.YUN\_FileInfo() { FileId = FileId, FileState = true, ParentFileId = "" };

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

Db.YUN\_FileInfo.Attach( F );

Db.Entry( F ).Property( x => x.FileState ).IsModified = true;

Db.Entry( F ).Property( x => x.ParentFileId ).IsModified = true;

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

/// <summary>

/// 获取当前人使用空间

/// </summary>

public decimal GetlistByCode( string Code )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT(CASE WHEN SUM(FileSizeKb) Is NULL THEN 0 ELSE SUM(FileSizeKb)END) as Size FROM YUN\_FileInfo where [CreateId]='" + Code + "'" );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<decimal> result1 = Db.Database.SqlQuery<decimal>( strSql.ToString() );

return result1.FirstOrDefault();

}

}

/// <summary>

/// 获取一个对象

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public Model.YUN\_FileInfo GetModel( int FileId )

{

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

var File = from b in Db.YUN\_FileInfo

where b.FileId == FileId

orderby b.FileId

select b;

return File.FirstOrDefault();

}

}

/// <summary>

/// 获取一个对象

/// </summary>

/// <param name="FileId"></param>

/// <returns></returns>

public string GetModelByShareFileID( string FileId )

{

string FileIds = "";

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

var File = from b in Db.YUN\_FileInfo

where b.ShareFileID == FileId

orderby b.FileId

select b;

foreach( var item in File.ToList() )

{

FileIds += item.FileId + ",";

}

return FileIds.Length > 0 ? FileIds.Substring( 0, FileIds.Length - 1 ) : "";

}

}

/// <summary>

/// 向上 获取文件夹名

/// </summary>

/// <param name="field"></param>

/// <param name="Ids"></param>

/// <returns></returns>

public Model.YUN\_FileInfo GetImgNextOrUp( string where )

{

StringBuilder strSql = new StringBuilder();

strSql.Append( "SELECT top 1 \* " );

strSql.Append( " FROM YUN\_FileInfo where " );

strSql.Append( where );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

DbRawSqlQuery<Model.YUN\_FileInfo> result1 = Db.Database.SqlQuery<Model.YUN\_FileInfo>( strSql.ToString() );

if( result1 != null )

{

return result1.FirstOrDefault();

}

else

{

return null;

}

}

}

//清空回收站

public bool ClearDel( string Code )

{

//数据库访问模式

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

var filelist = from b in Db.YUN\_FileInfo.ToList()

where b.CreateId == Code && b.FileState == false

select b;

var Filelist = filelist.ToList();

foreach( var item in Filelist )

{

string WebUrl = HttpContext.Current.Server.MapPath( item.FileUrl );

string FileName = System.IO.Path.GetFileNameWithoutExtension( WebUrl );

switch( item.FileExtName )

{

case ".docx":

case ".doc":

case ".xls":

case ".xlsx":

case ".ppt":

case ".pptx":

string PDfFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + item.FileName + ".pdf";

if( FileHelper.ExitFile( PDfFileUrl ) )

{

DeleteFile( PDfFileUrl );

}

break;

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

string FlvFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".flv";

string jpgFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".jpg";

if( FileHelper.ExitFile( FlvFileUrl ) )

{

DeleteFile( FlvFileUrl );

}

if( FileHelper.ExitFile( jpgFileUrl ) )

{

DeleteFile( jpgFileUrl );

}

break;

default:

break;

}

if( item.IsFolder == true )

{

string FileMapPath = "";

GetFileMapPathByDel( item.FileId.ToString(), false, ref FileMapPath );

DeleteFolderFile( HttpContext.Current.Server.MapPath( "/Upload/Yun/" + FileMapPath ) );

}

else

{

DeleteFile( WebUrl );

}

Db.YUN\_FileInfo.Remove( item );

}

int rows = Db.SaveChanges();

if( rows == Filelist.Count() )

{

return true;

}

else

{

return false;

}

}

}

//定时清理数据

public bool ClearFile()

{

string str4 = AppDomain.CurrentDomain.BaseDirectory; //获取基目录，它由程序集冲突解决程序用来探测程序集。

string str7 = AppDomain.CurrentDomain.SetupInformation.ApplicationBase;

//数据库访问模式

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

var filelist = from b in Db.YUN\_FileInfo.ToList()

where b.FileDeleteTime < DateTime.Now.AddDays( -10 ) && b.FileState == false

select b;

var Filelist = filelist.ToList();

foreach( var item in Filelist )

{

string WebUrl = str4 + item.FileUrl.Replace( "\\", "#$" ).Replace( "/", "\\" ).Replace( "#$", "\\" );

string FileName = System.IO.Path.GetFileNameWithoutExtension( WebUrl );

switch( item.FileExtName )

{

case ".docx":

case ".doc":

case ".xls":

case ".xlsx":

case ".ppt":

case ".pptx":

string PDfFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + item.FileName + ".pdf";

if( FileHelper.ExitFile( PDfFileUrl ) )

{

DeleteFile( PDfFileUrl );

}

break;

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

string FlvFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".flv";

string jpgFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".jpg";

if( FileHelper.ExitFile( FlvFileUrl ) )

{

DeleteFile( FlvFileUrl );

}

if( FileHelper.ExitFile( jpgFileUrl ) )

{

DeleteFile( jpgFileUrl );

}

break;

default:

break;

}

if( item.IsFolder == true )

{

string FileMapPath = "";

GetFileMapPathByDel( item.FileId.ToString(), false, ref FileMapPath );

DeleteFolderFile( HttpContext.Current.Server.MapPath( "/Upload/Yun/" + FileMapPath ) );

}

else

{

DeleteFile( WebUrl );

}

DeleteFile( WebUrl );

Db.YUN\_FileInfo.Remove( item );

}

int rows = Db.SaveChanges();

if( rows == Filelist.Count() )

{

return true;

}

else

{

return false;

}

}

}

//删除数据

public bool TrueDel( int fileId )

{

//数据库访问模式

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

var File = from b in Db.YUN\_FileInfo

where b.FileId == fileId

orderby b.FileId

select b;

Model.YUN\_FileInfo FileModel = File.FirstOrDefault();

string WebUrl = HttpContext.Current.Server.MapPath( FileModel.FileUrl );

string FileName = System.IO.Path.GetFileNameWithoutExtension( WebUrl );

switch( FileModel.FileExtName )

{

case ".docx":

case ".doc":

case ".xls":

case ".xlsx":

case ".ppt":

case ".pptx":

string PDfFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileModel.FileName + ".pdf";

if( FileHelper.ExitFile( PDfFileUrl ) )

{

DeleteFile( PDfFileUrl );

}

break;

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

string FlvFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".flv";

string jpgFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".jpg";

if( FileHelper.ExitFile( FlvFileUrl ) )

{

DeleteFile( FlvFileUrl );

}

if( FileHelper.ExitFile( jpgFileUrl ) )

{

DeleteFile( jpgFileUrl );

}

break;

default:

break;

}

if( FileModel.IsFolder == true )

{

}

else

{

DeleteFile( WebUrl );

}

Db.YUN\_FileInfo.Remove( FileModel );

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

//分享删除数据

public bool ShareTrueDel( int fileId )

{

//数据库访问模式

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

var File = from b in Db.YUN\_FileInfo

where b.FileId == fileId

orderby b.FileId

select b;

Model.YUN\_FileInfo FileModel = File.FirstOrDefault();

string WebUrl = HttpContext.Current.Server.MapPath( FileModel.FileUrl );

string FileName = System.IO.Path.GetFileNameWithoutExtension( WebUrl );

switch( FileModel.FileExtName )

{

case ".docx":

case ".doc":

case ".xls":

case ".xlsx":

case ".ppt":

case ".pptx":

string PDfFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileModel.FileName + ".pdf";

if( FileHelper.ExitFile( PDfFileUrl ) )

{

DeleteFile( PDfFileUrl );

}

break;

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

string FlvFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".flv";

string jpgFileUrl = System.IO.Path.GetDirectoryName( WebUrl ) + "\\" + FileName + ".jpg";

if( FileHelper.ExitFile( FlvFileUrl ) )

{

DeleteFile( FlvFileUrl );

}

if( FileHelper.ExitFile( jpgFileUrl ) )

{

DeleteFile( jpgFileUrl );

}

break;

default:

break;

}

if( FileModel.IsFolder == true )

{

}

else

{

DeleteFile( WebUrl );

}

Db.YUN\_FileInfo.Remove( FileModel );

int rows = Db.SaveChanges();

if( rows > 0 )

{

return true;

}

else

{

return false;

}

}

}

public decimal GetDiskSize( string userCode )

{

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

decimal UserSize = 0;

var UserRole = from b in Db.UserRoleInfo

where b.UserId == userCode

select b;

Model.UserRoleInfo User = UserRole.FirstOrDefault();

if( User != null )

{

UserSize = decimal.Parse( User.DiskSize.ToString() ) \* 1024;

}

else

{

UserSize = 500 \* 1024;

}

return UserSize;

}

}

}

}

**三、文件操作js类**

var ReNameFileId = "";//存储要重命名的文件id

var DFBGFlie = { Flie: {} }

DFBGFlie.Flie.FlieOperation = {

AjaxHtml: function () {

var url = arguments[0];

var data = arguments[1];

var callback = arguments[2];

var async = arguments[3];

var datatype = arguments.length > 4 ? arguments[4] : "html";

var complete = arguments.length > 5 ? arguments[5] : function () { };

$.ajax({

type: "POST",

url: url,

data: data,

dataType: datatype,

async: async,

success: function (msg) {

callback(msg);

},

complete: function (XMLHttpRequest, textStatus) {

complete(XMLHttpRequest, textStatus);

}

});

},

AjaxList: function () {

var url = arguments[0];

var data = arguments[1];

var callback = arguments[2];

$.ajax({

type: "POST",

url: url,

data: data,

dataType: "json",

beforeSend: function (XMLHttpRequest) {

},

success: function (msg, status) {

callback(msg, status);

},

complete: function (XMLHttpRequest, textStatus) {

},

error: function (e, x) {

}

});

},

//显示

ImgShow: function () {

///父级遮罩层隐藏

parent.$(".b-panel").fadeOut(500);

parent.$(".module").fadeOut(500);

parent.$(".module").html("");

},

//分享提示框

ShareTransfer: function () {

var \_base = DFBGFlie.Flie.FlieOperation;

var FileId = arguments.length > 0 ? arguments[0] : "";

var ShareType = arguments.length > 1 ? arguments[1] : "";//非空返回列表页

\_base.AjaxList("../Ajax/GetShare", "", function (data) {

var Str = "";

Str += "<div id=\"ReNameId\" style=\" min-width: 650px; \" class=\"ui-draggable\">";

Str += "<h1 id=\"popup\_title\" class=\"widgettitle YIdong\" style=\"cursor: move;\">分享文件</h1>";

Str += "<span onclick=\"$.closezhezhao('popup\_container2','popup\_overlay2')\" class=\"icon-fullscreen1\"></span>";

Str += "<div id=\"popup\_content2\" class=\"widgetcontent\">";

Str += "<div style=\"text-align:center;\"> ";

if (data.length > 0) {

for (var i = 0; i < data.length; i++) {

if (data[i].ShareTypeId == "1001") {

Str += "<input type=\"button\" onclick=\"DFBGFlie.Flie.FlieOperation.SchoolShare('" + FileId + "')\" value=\"" + data[i].ShareTypeName + "\" />";

} else {

Str += "<input type=\"button\" onclick=\"DFBGFlie.Flie.FlieOperation.ShareMethod('" + FileId + "','" + data[i].ShareTypeId + "','" + ShareType + "')\" value=\"" + data[i].ShareTypeName + "\" />";

}

}

}

Str += "</div>";

Str += "</div>";

Str += "</div>";

parent.$("#popup\_container2").append(Str);

$.dingwei('popup\_container2', 'popup\_overlay2', 'parent');

}

);

},

//分享

ShareMethod: function (FileId, ShareTypeId) {

var ShareType = arguments.length > 2 ? arguments[2] : "";//分享类型 非空返回列表页

DFBGFlie.Flie.FlieOperation.OnShare(FileId, ShareTypeId, ShareType);

},

//分享

OnShare: function () {

var FileId = arguments.length > 0 ? arguments[0] : "";

var ShareTypeId = arguments.length > 1 ? arguments[1] : "";

var ShareType = arguments.length > 2 ? arguments[2] : "";

var swich = getCookie("ShowType");

//判断如果是大图标展示，获取大图标展示中选择的id

if (FileId == "") {

if (swich == "grid") {

var cssId = window.frames["left"].$(".item-active");

for (var i = 0; i < cssId.length; i++) {

FileId += cssId[i].id + ",";

}

}

else {

//获取要分享的文件id(获取list列表展示中的id)

var list = window.frames["left"].$("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value + ",";

}

}

}

}

if (FileId == "" || FileId == undefined) {

alert("请选择要分享的文件!");

return false;

}

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/ShareMethod", { FileId: FileId, ShareTypeId: ShareTypeId }, function (data) {

if (data == "suc") {

alert("分享成功！");

if (ShareTypeId == "1003") {

var url = SiShareUrl(FileId);

$("#popup\_content2").html("<div style=\" padding: 20px 15px;text-align: center;\" id=\"popup\_message\">" + url + "</div><div id=\"popup\_panel\" style=\" padding: 10px 0px 20px;text-align: center;\"> <input type=\"button\" onclick=\"copyToClipboard('" + url + "')\" value=\"复制地址\" class=\"widgetcontent\" style=\"min-width: 100px\" id=\"popup\_ok\"></div>");

} else if (ShareType == "") {

window.location.href = window.location.href;

} else {

$.closezhezhao('popup\_container2', 'popup\_overlay2');

}

} else {

alert("分享失败！")

}

}, false);

},

//取消分享确定按钮

NotShare: function () {

var FileId = arguments.length > 0 ? arguments[0] : "";

var ShareType = arguments.length > 0 ? arguments[1] : "";//分享类型 非空返回列表页

var list = $("input[name='cheName']");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value + ",";

}

}

if (FileId == "" || FileId == undefined) {

alert("请选择要取消分享的文件!");

return false;

}

if (confirm("确定要取消分享吗？")) {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/IsShareMethods", { FileId: FileId }, function (data) {

if (data == "suc") {

alert("取消分享成功！");

if (ShareType == "") {

window.location.href = "/MyFile/MyShare";

}

} else {

alert("取消分享失败！")

}

}, false);

}

},

//还原确定按钮

Reduction: function () {

var FileId = arguments.length > 0 ? arguments[0] : "";

var ShareType = arguments.length > 0 ? arguments[1] : "";//分享类型 非空返回列表页

var list = $("input[name='cheName']");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value + ",";

}

}

if (FileId == "" || FileId == undefined) {

alert("请选择要取消分享的文件!");

return false;

}

if (confirm("确定要还原吗？")) {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/Reduction", { FileId: FileId }, function (data) {

if (data == "suc") {

alert("还原成功！");

if (ShareType == "") {

window.location.href = "/MyFile/Mydel";

}

} else {

alert("还原失败！")

}

}, false);

}

},

//批量删除

DelMore: function (Id, whereTransfer) {

var FileId = "";

if (Id != "" && Id != undefined) {

FileId = Id;

}

else {

var swich = getCookie("ShowType");

//判断如果是大图标展示，获取大图标展示中选择的id

if (swich == "grid") {

var cssId = window.frames["left"].$(".item-active");

for (var i = 0; i < cssId.length; i++) {

FileId += cssId[i].id + ",";

}

}

else {

//获取要删除的文件id(list展示中的id)

var list = window.frames["left"].$("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value + ",";

}

}

}

}

if (FileId == "" || FileId == undefined) {

alert("请选择要删除的文件!");

return false;

}

else {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/DelMore", { FileIds: FileId }, function (data) {

if (data == "suc") {

if (whereTransfer == "parent") {

$.closezhezhao('popup\_container', 'popup\_overlay');

window.frames["left"].window.location.reload();

} else {

window.location.href = window.location.href;

}

}

}, false);

}

},

//删除提示框

Transfer: function (Id, whereTransfer) {

//whereTransfer (this 当前的，parent父级的)

var container, kuan = "";

if (whereTransfer == "parent") {

container = parent.$("#popup\_container");

} else {

container = $("#popup\_container");

}

container.html(kuan);

kuan = kuan + " <h1 id=\"popup\_title\"class=\"widgettitle YIdong\" style=\"cursor: move;\">提示</h1><div id=\"popup\_content\" class=\"widgetcontent\">";

kuan = kuan + " <div id=\"popup\_message\" style=\" padding: 20px 15px;text-align: center;\">确定删除吗？</div> <div style=\" padding: 10px 0px 20px;text-align: center;\" id=\"popup\_panel\">";

kuan = kuan + " <input type=\"button\" id=\"popup\_ok\" style=\"min-width: 100px\" class=\"widgetcontent\" value=\"确定\" onclick=\"DFBGFlie.Flie.FlieOperation.DelMore('" + Id + "','" + whereTransfer + "')\">";

kuan = kuan + " <input type=\"button\" id=\"popup\_cancel\" style=\"min-width: 100px\" class=\"widgetcontent\" onclick=\"$.closezhezhao('popup\_container','popup\_overlay')\" value=\"取消\"></div></div>";

container.append(kuan);

$.dingwei('popup\_container', 'popup\_overlay', whereTransfer);

},

//选多个文件下载

DownFileMore: function () {

var FileId = "";

var swich = getCookie("ShowType");

//判断如果是大图标展示，获取大图标展示中选择的id

if (swich == "grid") {

var cssId = $(".item-active");

for (var i = 0; i < cssId.length; i++) {

FileId += cssId[i].id + ",";

}

}

else {

//获取要下载的文件id(list列表)

var list = $("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value + ",";

}

}

}

if (FileId == "" || FileId == undefined) {

alert("请选择要下载的文件!");

return false;

}

else {

try {

var url = "/File/DownFileMore?FileIds=" + FileId;

var elemIF = document.createElement("iframe");

elemIF.src = url;

elemIF.style.display = "none";

document.body.appendChild(elemIF);

}

catch (e) {

alert("下载失败！");

}

}

},

//选多个文件下载

DownLoad: function (FileId) {

try {

var url = "/File/DownFile?FileId=" + FileId;

var elemIF = document.createElement("iframe");

elemIF.src = url;

elemIF.style.display = "none";

document.body.appendChild(elemIF);

}

catch (e) {

alert("下载失败！");

}

},

//新建文件夹

NewsFile: function () {

$(".aGMvFtb").hide();

//获取显示类型（大图标显示/list列表显示）

var showType = getCookie("ShowType");

var html = "";

if (showType == "grid") {

html += "<div class=\"grid-view-item open-enable\" style=\"display: block;\">";

html += "<div class=\"FileShow\">";

html += "<div class=\"fileicon dir-large\">";

html += "</div>";

html += "</div>";

html += "<div class=\"file-name\"><input class=\"GadHyA\" onblur=\"DFBGFlie.Flie.FlieOperation.FileOnblur('a1','sp1')\" type=\"text\" value=\"新建文件夹\" id=\"a1\" /><span id=\"sp1\"></span></div> </div>";

$("#FileShow").append(html);

//$("#a1").focus();//获取焦点$("#a1")[0].focus();

//将光标移到文本最后面

MoveCursor("a1");

}

else {//列表

html += "<dd class=\"g-clearfix list-view-item open-enable hover-item item-active\">";

html += "<input type=\"checkbox\" class=\"checkbox1\" name=\"cheName\" />";

html += "<div class=\"fileicon dir-small\"></div>";

html += "<div style=\"width: 60%\" class=\"file-name\">";

html += "<div class=\"text\">";

html += "<a class=\"filename\" href=\"#\"><input class=\"filename\" onblur=\"DFBGFlie.Flie.FlieOperation.FileOnblur('a1','sp1')\" type=\"text\" value=\"新建文件夹\" id=\"a1\" /><span id=\"sp1\"></span></div>";

html += "<div class=\"operate\">";

html += "<div style=\"position: absolute; top: 0px; padding-top: 0px; line-height: normal;\">";

//分享

html += "<a href=\"javascript:void(0);\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"分享\" class=\"icon icon-share-blue\"></em> </span></a>";

//下载

html += "<a href=\"javascript:void(0);\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"下载\" class=\"icon icon-download-blue\"></em></span></a>";

html += "<span class=\"g-dropdown-button\">";

//更多

html += "<a href=\"javascript:void(0);\" class=\"g-button\"><span class=\"g-button-right\"></span></a>";

html += "<span class=\"menu\" style=\"width: 64px;display:none;\">";

html += "<a href=\"javascript:void(0);\" class=\"g-button-menu rename\">重命名</a>";

html += "<a href=\"javascript:void(0);\" class=\"g-button-menu delete\">删除</a>";

html += "</span></span>";

html += " </div>";

html += "</div>";

html += "</div>";

html += "<div style=\"width: 16%\" class=\"file-size\">-</div>";

var date = new Date();

html += "<div style=\"width: 23%\" class=\"ctime\">" + date.getFullYear() + "年" + (date.getMonth() + 1) + "月" + date.getDate() + "日"; "</div>";

html += "</dd>";

$("#FileShowList").append(html);

//$("#a1").focus();//获取焦点$("#a1")[0].focus()

//将光标移到文本最后面

MoveCursor("a1");

}

},

//给创建的文件夹命名

FileOnblur: function (inputId, spanId) {

var Name = $("#" + inputId + "").val();

if (Name.trim() == "") {

$(window.document).off();

Name = "新建文件夹";

}

var Share = $("#Share").val();

if (Share != undefined) {

Share == "1001";

} else {

Share = "";

}

$("#" + spanId + "").html(Name);

$("#" + inputId + "").remove();

//将新建的文件夹保存到库中

var ParentFileId = $("#HidFileId").val();//获取父级文件夹id

DFBGFlie.Flie.FlieOperation.AjaxHtml("/MyFile/AddFile", { ParentFileId: ParentFileId, FileName: Name, Share: Share }, function (data) {

if (data.indexOf("Suc") > -1) {

window.location.reload();

} else {

alert("创建失败！")

}

}, false);

},

//重命名确定按钮

OnReName: function (FileId) {

var name = $("#ReNameInfo").val();

if (name.trim() == "") {

alert("请输入文件夹名称");

$("#ReNameInfo").focus();

return;

} else {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/ReNameMethod", { FileId: FileId, Name: name },

function (data) {

if (data == "suc") {

$.closezhezhao('popup\_container', 'popup\_overlay');

window.frames["left"].location.reload();

} else {

alert("重命名失败！")

}

}, false);

}

},

OnReNameFor: function () {

var FileId = "";

var OldName = "";

var swich = getCookie("ShowType");

//判断如果是大图标展示，获取大图标展示中选择的id

if (swich == "grid") {

var cssId = $(".item-active");

for (var i = 0; i < cssId.length; i++) {

FileId = cssId[i].id.replace("add", "");

OldName = $("#" + cssId[i].id).children(".file-name").find(".filename").attr("title");

}

}

else {

//获取要删除的文件id(list展示中的id)

var list = $("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value;

OldName = $(list[i]).next().next().children(".text").children("a").text();

}

}

}

if (FileId != "") {

var \_Base = DFBGFlie.Flie.FlieOperation;

ReNameMethod(FileId, "", OldName);

}

},

GetFileSize: function (Size) {

if (Size > 1048576) {

Size = (parseFloat(Size) / 1024 / 1024).toFixed(2);

return Size + "G";

} else if (Size > 1024) {

Size = (parseFloat(Size) / 1024).toFixed(2);

return Size + "MB";

} else if (Size == "0") {

return "-";

} else {

return Size + "KB";

}

},

//根据类型显示

FileShowControl: function (FileExtName, FileId) {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -77.5px;\"></div>");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-loading\"></i><span class=\"tip-msg\">正在打开文件，请稍候…</span></div></span></div>");

$(".module-tip").css("display", "block");

$(document).on('mousemove', function (e) { });

var \_base = DFBGFlie.Flie.FlieOperation;

switch (FileExtName) {

case ".jpg":

case ".png":

case ".jpeg":

case ".gif":

case ".bmp":

var parentFileId = $.toString($("#HidFileId").val());

DFBGFlie.Flie.FlieOperation.AjaxHtml("/Details/Img", { FileId: FileId, ParentFileId: parentFileId }, function (data) {

if (data != "") {

\_base.Layer();

parent.$(".module").append(data);

parent.$(".b-panel").fadeIn(1000);

parent.$(".module").fadeIn(1000);

}

}, true);

break;

case ".docx":

case ".doc":

case ".xls":

case ".xlsx":

case ".ppt":

case ".pptx":

DFBGFlie.Flie.FlieOperation.AjaxHtml("/Details/Office", { FileId: FileId }, function (data) {

if (data != "") {

\_base.Layer();

parent.$("#popup\_container2").html(data);

parent.$("#popup\_container2").css({ "position": "absolute", "top": 0, "left": 0, "right": "0px", "bottom": "0px", "display": "block", "max-height": "100%" });

parent.$("#popup\_overlay").css({ "display": "block" });

}

}, true);

break;

case ".pdf":

DFBGFlie.Flie.FlieOperation.AjaxHtml("/Details/Pdf", { FileId: FileId }, function (data) {

if (data != "") {

\_base.Layer();

parent.$("#popup\_container2").html(data);

parent.$("#popup\_container2").css({ "position": "absolute", "top": 0, "left": 0, "right": "0px", "bottom": "0px", "display": "block", "max-height": "100%" });

parent.$("#popup\_overlay").css({ "display": "block" });

}

}, true);

break;

case ".flv":

case ".mp4":

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

DFBGFlie.Flie.FlieOperation.AjaxHtml("/Details/Video", { FileId: FileId }, function (data) {

if (data != "") {

\_base.Layer();

parent.$("#popup\_container").html(data);

$.dingwei('popup\_container', 'popup\_overlay', 'parent');

}

}, false);

break;

case ".mp3":

case ".wma":

case ".wav":

DFBGFlie.Flie.FlieOperation.AjaxHtml("/Details/Audio", { FileId: FileId }, function (data) {

if (data != "") {

\_base.Layer();

parent.$("#popup\_container").html(data);

$.dingwei('popup\_container', 'popup\_overlay', 'parent');

}

}, false);

break;

case ".txt":

case ".rar":

case ".zip":

case ".exe":

case ".psd":

case ".xmind":

\_base.Layer();

DFBGFlie.Flie.FlieOperation.DownLoad(FileId);

break;

default:

\_base.Layer();

DFBGFlie.Flie.FlieOperation.DownLoad(FileId);

break;

}

},

//移动文件

MoveShow: function (MoveType) {

//MoveType 1复制 2移动

DFBGFlie.Flie.FlieOperation.AjaxHtml("/Details/FileMove", { MoveType: MoveType }, function (data) {

if (data != "") {

parent.$("#popup\_container").append(data);

$.dingwei('popup\_container', 'popup\_overlay', 'parent');

}

}, true);

},

//加载目录信息

Move: function (treeview, obj, Pl, Share) {

var url = "/Details/folderData";

var par = {

ParentFileId: treeview,

Share: Share

}

EDUCAjax(par, function () {

},

function (re) {

var date = re;

var \_base = DFBGFlie.Flie.FlieOperation;

var Next = "";

for (var i = 0; i < date.length; i++) {

Next += "<li>";

Next += "<div style=\"padding-left:" + (parseInt(Pl) + 15) + "px\" class=\"treeview-node " + (parseInt(date[i]["IsChildFolder"]) > 0 ? "" : "treenode-empty") + "\" \_childfolder=\"" + date[i]["IsChildFolder"] + "\" >";

Next += "<span class=\"treeview-node-handler\">";

Next += "<em class=\"b-in-blk plus icon-operate \"></em> ";

Next += "<dfn class=\"b-in-blk treeview-ic\"></dfn> ";

Next += "<span class=\"treeview-txt\" treeview=\"" + date[i]["FileId"] + "\">" + date[i]["FileName"] + "</span>";

Next += "</span>";

Next += "</div>";

Next += "<ul class=\"treeview treeview-content treeview-collapse\"></ul>";

Next += "</li>";

}

$(obj).next().html(Next);

\_base.treeview(Share);

}, url, "json", false, "数据加载中，请耐心等待...", false);

},

//移动文件弹出层添加文件夹

MoveAddFile: function () {

var AddFile = "";

var Folder = $("#plus-createFolder");

var \_base = DFBGFlie.Flie.FlieOperation;

if (Folder.length > 0) {

$(".\_disk\_id\_4").focus().select();

} else {

var Pl = $(".treeview-node-on").css("padding-left").replace("px", "");

var treeview = $(".treeview-node-on").children().children("span").attr("treeview");

AddFile += "<li id=\"plus-createFolder\">";

AddFile += "<div style=\"padding-left:" + (parseInt(Pl) + 15) + "px\" class=\"treeview-node treenode-empty\">";

AddFile += "<span class=\"treeview-node-handler\">";

AddFile += "<em class=\"b-in-blk plus sprite-ic2 icon-operate\"></em> ";

AddFile += "<dfn class=\"b-in-blk treeview-ic\"></dfn>";

AddFile += " <span class=\"plus-create-folder\">";

AddFile += "<input type=\"text\" value=\"新建文件夹\" class=\"input \_disk\_id\_4\">";

AddFile += "<span class=\"sure \_disk\_id\_2\" pId=" + treeview + " ></span>";

AddFile += "<span class=\"cancel \_disk\_id\_3\"></span></span></span>";

AddFile += "</div><ul \_pl=\"75\" class=\"treeview treeview-content treeview-collapse\"></ul></li>";

$(".treeview-node-on").next("ul").append(AddFile);

$(".\_disk\_id\_4").focus().select();

$(".\_disk\_id\_3").click(function () {

$(this).parent().parent().parent().parent("li").remove();

});

$(".\_disk\_id\_2").click(function () {

var Name = $(".\_disk\_id\_4").val();

var pId = $(this).attr("pId");

if (Name.trim() != "") {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/MyFile/AddFile", { ParentFileId: pId, FileName: Name }, function (data) {

if (data.indexOf("Suc") > -1) {

$(".\_disk\_id\_4").parent().removeClass("plus-create-folder").addClass("treeview-txt").html(Name).attr("treeview", data.replace("Suc", ""));

$("#plus-createFolder").removeAttr("id");

$(".treeview-node-on").removeClass("treenode-empty").addClass("\_minus").attr("\_childfolder", "1").children().children("em").toggleClass("minus");

var \_base = DFBGFlie.Flie.FlieOperation;

\_base.treeview();

}

}, false);

}

})

}

},

//展开/不展开

treeview: function (Share) {

$(".treeview-node").off().hover(function () {

$(this).toggleClass("treeview-node-hover");

}).click(function () {

$(".treeview-node").removeClass("treeview-node-on");

$(this).addClass("treeview-node-on").next("ul").toggleClass("treeview-collapse");

if (parseInt($(this).attr("\_childfolder")) > 0) {

var ul = $(this).next().children();

if (ul.length == 0) {

var treeview = $(this).children().children("span").attr("treeview");

var Pl = $(this).css("padding-left").replace("px", "");

var \_base = DFBGFlie.Flie.FlieOperation;

\_base.Move(treeview, this, Pl, Share);

}

$(this).toggleClass("\_minus").children().children("em").toggleClass("minus");

}

})

},

//移动文件

MoveFile: function () {

var MoveFileId = "";

var treeviewId = $(".treeview-node-on").children().children("span").attr("treeview");

//MoveType 1复制 2移动

var MoveType = $("#fileTreeDialog").attr("\_movetype");

var MoveStr = "";

var swich = getCookie("ShowType");

var Share = window.frames["left"].$("#Share").val();

if (Share != undefined) {

Share = true;

} else {

Share = false;

}

//判断如果是大图标展示，获取大图标展示中选择的id

if (swich == "grid") {

var cssId = window.frames["left"].$(".item-active");

for (var i = 0; i < cssId.length; i++) {

MoveFileId += cssId[i].id + ",";

}

}

else {

//获取要删除的文件id(list展示中的id)

var list = window.frames["left"].$("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

MoveFileId += list[i].value + ",";

}

}

}

if (MoveFileId == "") {

return;

}

var url = "../File/FileMoveTo";

var par = { FileIdList: MoveFileId, WhereId: treeviewId, MoveType: MoveType, Share: Share }

if (MoveType == "1") {

MoveStr = "复制";

} else {

MoveStr = "移动";

}

EDUCAjax(par, function () {

$.closezhezhao('popup\_container', 'popup\_overlay');

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -77.5px;\"></div>");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-loading\"></i><span class=\"tip-msg\">正在" + MoveStr + "文件，请稍候…</span></div></span></div>");

$(".module-tip").css("display", "block");

},

function (data) {

if (data == "suc") {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -48.5px\"></div>");

} else {

$(".module-tip").css("margin-left", "-48.5px");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-success\"></i><span class=\"tip-msg\">文件" + MoveStr + "成功</span></div>");

$(".module-tip").css("display", "block");

window.frames["left"].location.reload();

setTimeout("$(\".module-tip\").hide()", 5000);

} else {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -48.5px\"></div>");

} else {

$(".module-tip").css("margin-left", "-48.5px");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-success\"></i><span class=\"tip-msg\">文件" + MoveStr + "失败</span></div>");

$(".module-tip").css("display", "block");

window.frames["left"].location.reload();

setTimeout("$(\".module-tip\").hide()", 5000);

}

}, url, "html", false, "", false);

},

Layer: function () {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -48.5px\"></div>");

} else {

$(".module-tip").css("margin-left", "-48.5px");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-success\"></i><span class=\"tip-msg\">文件打开成功</span></div>");

$(".module-tip").css("display", "block");

setTimeout("$(\".module-tip\").hide()", 2000);

},

//清空回收站

ClearDel: function () {

if (confirm("确定要清空吗？")) {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/ClearDel", {}, function (data) {

if (data == "suc") {

alert("清空成功！");

window.location.href = "/MyFile/Mydel";

} else {

alert("清空失败！")

}

}, false);

}

},

TrueDel: function () {

var FileId = arguments.length > 0 ? arguments[0] : "";

if (FileId == "") {

var list = $("input[name='cheName']");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileId += list[i].value + ",";

}

}

if (FileId == "" || FileId == undefined) {

alert("请选择要永久删除的文件!");

return false;

}

}

if (confirm("确定要永久删除吗？")) {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/TrueDel", { FileIds: FileId }, function (data) {

if (data == "suc") {

alert("删除成功！");

window.location.href = "/MyFile/Mydel";

} else {

alert("删除失败！")

}

}, false);

}

},

SetContainerHeight: function () {

var ShowCookieType = getCookie("ShowType");

if (ShowCookieType == "List") {

$(".list-view-container").css({ "height": $(window).height() - 124 });

} else if (ShowCookieType == "grid") {

$(".list-view-container").css({ "height": $(window).height() - 81 });

}

},

///学校分享

SchoolShare: function () {

$.closezhezhao('popup\_container2', 'popup\_overlay2');

var FileId = arguments.length > 0 ? arguments[0] : "";

DFBGFlie.Flie.FlieOperation.AjaxHtml("/SchoolShare/SchoolFileMove", { FileId: FileId }, function (data) {

if (data != "") {

parent.$("#popup\_container").html("");

parent.$("#popup\_container").append(data);

$.dingwei('popup\_container', 'popup\_overlay', 'parent');

}

}, true);

},

///学校分享

SchoolShareFile: function () {

var MoveFileId = arguments.length > 0 ? arguments[0] : "";

var treeviewId = $(".treeview-node-on").children().children("span").attr("treeview");

if (MoveFileId == "") {

var MoveStr = "";

var swich = getCookie("ShowType");

//判断如果是大图标展示，获取大图标展示中选择的id

if (swich == "grid") {

var cssId = window.frames["left"].$(".item-active");

for (var i = 0; i < cssId.length; i++) {

MoveFileId += cssId[i].id + ",";

}

}

else {

//获取要删除的文件id(list展示中的id)

var list = window.frames["left"].$("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

MoveFileId += list[i].value + ",";

}

}

}

}

if (MoveFileId == "") {

return;

}

var url = "../SchoolShare/SchoolShareFile";

var par = { FileIdList: MoveFileId, WhereId: treeviewId }

EDUCAjax(par, function () {

$.closezhezhao('popup\_container', 'popup\_overlay');

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -77.5px;\"></div>");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-loading\"></i><span class=\"tip-msg\">正在分享文件，请稍候…</span></div></span></div>");

$(".module-tip").css("display", "block");

},

function (data) {

if (data == "suc") {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -48.5px\"></div>");

} else {

$(".module-tip").css("margin-left", "-48.5px");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-success\"></i><span class=\"tip-msg\">文件分享成功</span></div>");

$(".module-tip").css("display", "block");

window.frames["left"].location.reload();

setTimeout("$(\".module-tip\").hide()", 5000);

}

}, url, "html", false, "", false);

},

isSelect: function () {

var FileisSelect = false;

var cssId = $(".item-active");

if (cssId.length > 0) {

FileisSelect = true;

}

if (FileisSelect) {

$(".number").html(cssId.length);

$(".FlieOperation").css("display", "inline-block");

} else {

$(".FlieOperation").hide();

}

}

}

//获取焦点后将光标移到文本的末尾

function MoveCursor(Id) {

var obj = document.getElementById(Id);

obj.value = obj.value;

obj.focus();

var len = obj.value.length;

if (document.selection) {

var sel = obj.createTextRange();

sel.moveStart('character', len);

sel.collapse();

sel.select();

} else if (typeof obj.selectionStart == 'number'

&& typeof obj.selectionEnd == 'number') {

obj.selectionStart = obj.selectionEnd = len;

}

}

//给文件重命名

function ReNameMethod(FileId, hidId, OldName) {

var Str = "";

Str += "<div id=\"ReNameId\" style=\" min-width: 300px; max-width: 300px;\" class=\"ui-draggable\">";

Str += "<h1 id=\"popup\_title\" class=\"widgettitle YIdong\" style=\"cursor: move;\">重命名</h1>";

Str += "<span onclick=\"$.closezhezhao('popup\_container','popup\_overlay')\" class=\"icon-fullscreen1\"></span>";

Str += "<div id=\"popup\_content2\" class=\"widgetcontent\">";

Str += "<div id=\"popup\_message2\">";

Str += "名称：";

Str += "<input type=\"text\" id=\"ReNameInfo\" class=\"input1\" style=\"width: 200px; border: 1px solid #2a84e9; height: 21px; padding: 2px 5px; width: 200px; \" value=\"" + OldName + "\" />";

Str += "</div>";

Str += "<div style=\"text-align:center;\"> <input type=\"button\" value=\"确定\" onclick=\"DFBGFlie.Flie.FlieOperation.OnReName('" + FileId + "')\" /></div>";

Str += "</div>";

Str += "</div>";

parent.$("#popup\_container").append(Str);

$.dingwei('popup\_container', 'popup\_overlay', 'parent');

}

function SiShareUrl(FileId) {

var host = "http://" + window.location.host;

var num1 = GetRandomNum(5);

var num2 = GetRandomNum(5);

var num3 = GetRandomNum(5);

return host + "/SchoolShare/ShareDetails?url=" + encodeURIComponent(encodeURIComponent(num1 + FileId + num2 + num3));

}

var chars = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'];

function GetRandomNum(n) {

var res = "";

for (var i = 0; i < n ; i++) {

var id = Math.ceil(Math.random() \* 35);

res += chars[id];

}

return res;

}

function copyToClipboard(txt) {

if (window.clipboardData) {

window.clipboardData.clearData();

clipboardData.setData("Text", txt);

alert("复制成功！");

} else if (navigator.userAgent.indexOf("Opera") != -1) {

window.location = txt;

} else if (window.netscape) {

try {

netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect");

} catch (e) {

alert("被浏览器拒绝！\n请在浏览器地址栏输入'about:config'并回车\n然后将 'signed.applets.codebase\_principal\_support'设置为'true'");

}

var clip = Components.classes['@mozilla.org/widget/clipboard;1'].createInstance(Components.interfaces.nsIClipboard);

if (!clip)

return;

var trans = Components.classes['@mozilla.org/widget/transferable;1'].createInstance(Components.interfaces.nsITransferable);

if (!trans)

return;

trans.addDataFlavor("text/unicode");

var str = new Object();

var len = new Object();

var str = Components.classes["@mozilla.org/supports-string;1"].createInstance(Components.interfaces.nsISupportsString);

var copytext = txt;

str.data = copytext;

trans.setTransferData("text/unicode", str, copytext.length \* 2);

var clipid = Components.interfaces.nsIClipboard;

if (!clip)

return false;

clip.setData(trans, null, clipid.kGlobalClipboard);

alert("复制成功！");

}

}

**四、我的全部文件操作js类**

var DFBG = { FlieOA: {} }

var maxnum = 0;

var SumListNum = 0;

var SumGridNum = 0;

//打开文件上传窗口

var \_layerIndex;

function showLayer() {

\_layerIndex = $.layer({

type: 2,

maxmin: true,

title: '文件上传',

area: ['700px', '415px'],

fix: true,

border: [2, 0.3, '#000'],

iframe: {

src: '/File/Upload',

scrolling: 'auto'

},

close: function (index) {

layer.close(\_layerIndex);

}

});

//注册关闭方法

if (!window.closeUpPageHandler) {

window.closeUpPageHandler = function (files) {

setTimeout(function () {

layer.close(\_layerIndex);

}, 100);

}

}

}

var OptionFile = (function (opt) {

var o = {

length: 0

};

var o = $.extend(o, opt),

\_body = $('body'),

boxBg = '<div id="select"></div>';

return {

actionLock: false, //移动锁定

releaseTarget: false, //释放锁定

keyCode: null, //当前按键 键值

//鼠标按下操作

optionDown: function (length, type, evt) {

this.releaseTarget = false;

this.getImgList(length);

var currentX = evt.pageX;

var currentY = evt.pageY;

$('#select').css({

top: currentY + 1,

left: currentX + 1

})

},

//鼠标移动操作

optionMoving: function (length, type, evt) {

if (this.actionLock) {

this.optionDown(length, type, evt);

}

},

getImgList: function (length) {

if (length > 10) {

length = parseInt(length / 10);

}

if (length > 100) {

length = parseInt(length / 100);

}

if (!this.actionLock) {

\_body.append(boxBg);

var movingBox = '<em>' + length + '</em>';

$("#select").html(movingBox);

}

this.actionLock = true;

},

//放开鼠标操作(回调函数，返回按键键值和当前目标)

closeOption: function (func) {

var \_this = this;

$(document).keydown(function (event) {

\_this.keyCode = event.keyCode;

$(document).on('mouseup', function (e) {

if (!\_this.releaseTarget) {

$('.moving-box').remove();

\_this.actionLock = false;

$(document).unbind('mousemove');

\_this.releaseTarget = true;

func(e, \_this.keyCode); //返回当前 释放的 目标元素 ， 和按键code

$(document).unbind('keydown');

\_this.keyCode = null;

}

})

});

$(document).trigger("keydown");

$(document).keyup(function (event) {

$(document).unbind('keyup');

$(document).unbind('keydown');

\_this.keyCode = null;

})

}

}

})

DFBG.FlieOA.Flie = {

//我的网盘页面数据加载

MyFileLoad: function (FileId, Name) {

var \_base = DFBGFlie.Flie.FlieOperation;

var \_base2 = DFBG.FlieOA.Flie;

\_base.AjaxList("/MyFile/MyFileLoad", { FileId: FileId, SearchName: Name },

function (re, status) {

var data = re.model;

if (status == "success") {

var htmlInfo = "";//大图标展示

var htmlList = "";//列表展示

if (data.length != 0) {

//共多少个文件

$("#SumNum").html("已全部加载，共" + data.length + "个");

$("#SumUse").html("磁盘空间已使用：" + DFBGFlie.Flie.FlieOperation.GetFileSize(re.SumSize) + " 未使用：" + DFBGFlie.Flie.FlieOperation.GetFileSize(parseFloat(re.UserSize) - parseFloat(re.SumSize)));

for (var i = 0; i < data.length; i++) {

var model = data[i];

htmlInfo += "<div id=\"add" + model.FileId + "\" class=\"grid-view-item open-enable\" style=\"display: block;\">";

htmlInfo += " <div class=\"FileShow\">";

//\*\*\*\*图标\*\*\*

//判断是否是文件夹，如果是文件夹连接为打开文件夹，如果不是文件夹，连接为下载连接

if (model.IsFolder == true) {

htmlInfo += " <div onclick=\"DFBG.FlieOA.Flie.FileClick('" + model.FileId + "','" + model.FileName + "')\">";

//------ 修改文件夹样式

}

else {

htmlInfo += " <div onclick=\"DFBGFlie.Flie.FlieOperation.FileShowControl('" + model.FileExtName + "','" + model.FileId + "')\">";

}

//判断文件的类型，调用不同的文件图片样式

if (model.IsFolder == true) {//文件夹

htmlInfo += "<span class=\"fileicon dir-large\">";

}

else {

switch (model.FileExtName) {

case ".jpg":

case ".png":

case ".jpeg":

case ".gif":

case ".bmp":

//图片

htmlInfo += "<span class=\"fileicon fileicon-large-pic\">";

htmlInfo += "<img class=\"thumb fileicon-all\" style=\"width:100px;height:100px;\" data-url=\"/File/ImgSuoLue?FileId=" + model.FileId + "\" src=\"\">";

break;

case ".docx":

case ".doc":

htmlInfo += "<span class=\"fileicon fileicon-large-doc\">";

break;

case ".xls":

case ".xlsx":

htmlInfo += "<span class=\"fileicon fileicon-large-xls\">";

break;

case ".txt":

htmlInfo += "<span class=\"fileicon fileicon-large-txt\">";

break;

case ".rar":

htmlInfo += "<span class=\"fileicon fileicon-large-rar\">";

break;

case ".zip":

htmlInfo += "<span class=\"fileicon fileicon-large-zip\">";

break;

case ".pdf":

htmlInfo += "<span class=\"fileicon fileicon-large-pdf\">";

break;

case ".flv":

case ".mp4":

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

htmlInfo += "<span class=\"fileicon fileicon-large-video\">";

break;

case ".exe":

htmlInfo += "<span class=\"fileicon fileicon-large-exe\">";

break;

case ".psd":

htmlInfo += "<span class=\"fileicon fileicon-large-psd\">";

break;

case ".mp3":

case ".wma":

case ".wav":

htmlInfo += "<span class=\"fileicon fileicon-large-mp3\">";

break;

case ".ppt":

case ".pptx":

htmlInfo += "<span class=\"fileicon fileicon-large-ppt\">";

break;

case ".xmind":

htmlInfo += "<span class=\"fileicon fileicon-large-xmind\">";

break;

default:

htmlInfo += "<span class=\"fileicon fileicon-all\">";

break;

}

}

htmlInfo += "</span>";

htmlInfo += "</div>";

htmlInfo += "<label onmousedown=\"ClickSelCss('add" + model.FileId + "')\" class=\"checkbox\"></label>";

htmlInfo += "</div>";

if (model.FileName.length > 10) {

htmlInfo += "<div class=\"file-name\"><span title=\"" + model.FileName + "\" class=\"filename\">" + model.FileName.substr(0, 6) + "..." + model.FileExtName + "</span></div> ";

}

else {

htmlInfo += "<div class=\"file-name\"><span title=\"" + model.FileName + "\" href=\"javascript:void(0);\" class=\"filename\">" + model.FileName + "</span></div>";

}

htmlInfo += "</div>";

//\*\*\*\*图标\*\*\*

//\*\*\*\*List方式展示\*\*\*

htmlList += "<div class=\"g-clearfix list-view-item open-enable\">";

//判断文件的类型调不同的图标样式

if (model.IsFolder == true) {//文件夹

//---GYF 新增

htmlList += "<input type=\"checkbox\" onclick=\"YzCheAll()\" class=\"checkbox1\" name=\"cheName\" value=\"" + model.FileId + "\" />";

//--------

htmlList += "<div class=\"fileicon dir-small\"></div>";

//htmlList += "<span class=\"checkbox\"></span>";

}

else {

//非文件夹放选择框

htmlList += "<input type=\"checkbox\" onclick=\"YzCheAll()\" class=\"checkbox1\" name=\"cheName\" value=\"" + model.FileId + "\" />";

//图片

switch (model.FileExtName) {

case ".jpg":

case ".png":

case ".jpeg":

case ".jpg":

case ".gif":

case ".bmp":

//图片

htmlList += "<div class=\"fileicon fileicon-small-pic\"></div>";

break;

case ".docx":

case ".doc":

htmlList += "<div class=\"fileicon fileicon-small-doc\"></div>";

break;

case ".xls":

case ".xlsx":

htmlList += "<div class=\"fileicon fileicon-small-xls\"></div>";

break;

case ".txt":

htmlList += "<div class=\"fileicon fileicon-small-txt\"></div>";

break;

case ".rar":

htmlList += "<div class=\"fileicon fileicon-small-rar\"></div>";

break;

case ".zip":

htmlList += "<div class=\"fileicon fileicon-small-zip\"></div>";

break;

case ".pdf":

htmlList += "<div class=\"fileicon fileicon-small-pdf\"></div>";

break;

case ".flv":

case ".mp4":

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

htmlList += "<div class=\"fileicon fileicon-small-video\"></div>";

break;

case ".exe":

htmlList += "<div class=\"fileicon fileicon-small-exe\"></div>";

break;

case ".psd":

htmlList += "<div class=\"fileicon fileicon-small-psd\"></div>";

break;

case ".mp3":

case ".wma":

case ".wav":

htmlList += "<div class=\"fileicon fileicon-small-mp3\"></div>";

break;

case ".ppt":

case ".pptx":

htmlList += "<div class=\"fileicon fileicon-small-ppt\"></div>";

break;

case ".xmind":

htmlList += "<div class=\"fileicon fileicon-small-xmind\"></div>";

break;

default:

htmlList += "<div class=\"fileicon fileicon-small\"></div>";

break;

}

}

htmlList += "<div style=\"width: 60%\" class=\"file-name\">";

htmlList += "<div class=\"text\">";

//判断是否是文件夹，如果是文件夹连接为打开文件夹

if (model.IsFolder == true) {

htmlList += "<a onclick=\"DFBG.FlieOA.Flie.FileClick('" + model.FileId + "','" + model.FileName + "')\" class=\"filename\" href=\"#\" title=\"" + model.FileName + "\">" + model.FileName + "</a></div>";

}

else {

htmlList += "<a onclick=\"DFBGFlie.Flie.FlieOperation.FileShowControl('" + model.FileExtName + "','" + model.FileId + "')\" class=\"filename\" href=\"#\" title=\"" + model.FileName + "\">" + model.FileName + "</a></div>";

}

htmlList += "<div class=\"operate\">";

htmlList += "<div style=\"position: absolute; top: 0px; padding-top: 0px; line-height: normal;\">";

//（文件夹不放下载按钮和分享按钮）

if (model.IsFolder != true) {

//分享

htmlList += "<a onclick=\"DFBGFlie.Flie.FlieOperation.ShareTransfer(" + model.FileId + ")\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"分享\" class=\"icon icon-share-blue\"></em> </span></a>";

//下载

htmlList += "<a href=\"/File/DownFile?FileId=" + model.FileId + "\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"下载\" class=\"icon icon-download-blue\"></em></span></a>";

}

else {

//放空标签占位

htmlList += "<a onclick=\"DFBGFlie.Flie.FlieOperation.ShareTransfer(" + model.FileId + ")\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"分享\" class=\"icon icon-share-blue\"></em> </span></a>";

//htmlList += "<a class=\"g-button\"></a><a class=\"g-button\"></a>";

}

htmlList += "<span class=\"g-dropdown-button Baiparent\">";

//更多

htmlList += "<a onclick=\"MoreMethod(" + model.FileId + ")\" href=\"#\" class=\"g-button\"><span class=\"g-button-right\"></span></a>";

htmlList += "<span class=\"menu menu" + model.FileId + "\" onmouseout=\"\" style=\"width: 64px;display:none;\">";

htmlList += "<a href=\"#\" onclick=\"ReNameMethod(" + model.FileId + ",'menu" + model.FileId + "','" + model.FileName + "')\" class=\"g-button-menu rename\">重命名</a>";

htmlList += "<a href=\"#\" onclick=\"DFBGFlie.Flie.FlieOperation.Transfer(" + model.FileId + ",'parent')\" class=\"g-button-menu delete\">删除</a>";

htmlList += "</span></span>";

htmlList += " </div>";

htmlList += "</div>";

htmlList += "</div>";

htmlList += "<div style=\"width: 16%\" class=\"file-size\">" + DFBGFlie.Flie.FlieOperation.GetFileSize(model.FileSizeKb) + "</div>";

var date = new Date(parseInt(model.FileCreateTime.replace("/Date(", "").replace(")/", ""), 10));

htmlList += "<div style=\"width: 23%\" class=\"ctime\">" + date.Format("yyyy年MM月dd日 hh:mm") + "</div>";

htmlList += "</div>";

htmlList += "</div>";

}

} else {

htmlList += "<div class=\"aGMvFtb\">";

htmlList += "<div class=\"hqvyGW05 wegnv3JM\">";

htmlList += "<p class=\"ggEQwFb\">";

htmlList += " 您还没上传过文件哦，点击";

htmlList += "<span class=\"cjujvy90 upload-wrapper\" onclick='showLayer();'>";

htmlList += " 上传";

htmlList += " </span>按钮～";

htmlList += "</p>";

htmlList += "</div>";

htmlList += "</div>";

htmlInfo = htmlList;

}

htmlInfo += " <div style=\"clear: both\"></div>";

//\*\*\*List列表显示默认加一个文件夹放项目文件

$("#FileShowList").html(htmlList);

//\*\*\*图标显示默认加一个文件夹放项目文件

$("#FileShow").html(htmlInfo);

$(".module-list-view .list-view-item").hover(function () {

$(this).children(".file-name").children(".operate").toggle();

});

if ($(".list-view-item").length > 3) {

$(".list-view-item:last .operate .menu").css({ "left": "-40px", "top": "-50px" });

}

var optionImg = new OptionFile();

$(".module-grid-view .grid-view li").on("mousedown", function (e) {

if ($(this).hasClass('item-active')) {

e.preventDefault();

e.stopPropagation();

}

var activelength = $(".item-active").length;

var loop = setTimeout(function () {

optionImg.optionDown(activelength, 1, e);

$(document).mousemove(function (e) {

optionImg.optionMoving(activelength, 1, e);

optionImg.closeOption(function (e, keycode) {

var target = e.target.className;//目标位置 可以判断目标不同位置执行不同操作

$("#select").remove();

if (target == "filename" || target == "fileicon dir-large" || target == "grid-view-item open-enable") {

var MoveFileId = "";

var treeviewId = "";

var swich = getCookie("ShowType");

if (swich == "grid") {

var cssId = $(".item-active");

for (var i = 0; i < cssId.length; i++) {

MoveFileId += cssId[i].id + ",";

}

}

switch (target) {

case "filename":

treeviewId = $(e.target).parent().parent().attr("id").replace("add", "");

break;

case "fileicon dir-large":

treeviewId = $(e.target).parent().parent().parent().attr("id").replace("add", "");

break;

case "grid-view-item open-enable":

treeviewId = $(e.target).attr("id").replace("add","");

break;

default:

break;

}

if (treeviewId != "" ) {

\_base2.TuoMoVe(MoveFileId, treeviewId);

}

}

});

});

}, 200);

$(document).mouseup(function () {

clearTimeout(loop);

});

})

\_base.SetContainerHeight();

$("img").load(function () {

//图片默认隐藏

$(this).hide();

//使用fadeIn特效

$(this).stop().fadeIn("5000");

});

// 异步加载图片，实现逐屏加载图片

$(".grid-view-item").scrollLoading();

}

}

);

},

MyGridFileShowLoad: function (Type, Name, pageSize, pageIndex) {

var \_base = DFBGFlie.Flie.FlieOperation;

var \_base2 = DFBG.FlieOA.Flie;

\_base.AjaxList("/MyFile/MyFileShowLoad", { Type: Type, SearchName: Name, pageSize: pageSize, p: pageIndex },

function (re, status) {

var data = re.model;

if (status == "success") {

var htmlInfo = "";//列表展示

if (data.length != 0) {

//共多少个文件

\_base2.GetSumNum(re.SumCount, data.length, pageSize, pageIndex, "grid");

for (var i = 0; i < data.length; i++) {

var model = data[i];

htmlInfo += "<div id=\"add" + model.FileId + "\" href=\"javascript:;\" class=\"grid-view-item open-enable\" style=\"display: block;\">";

htmlInfo += " <div class=\"FileShow\">";

//\*\*\*\*图标\*\*\*

//判断是否是文件夹，如果是文件夹连接为打开文件夹，如果不是文件夹，连接为下载连接

if (model.IsFolder == true) {

htmlInfo += " <div onmousedown=\"DFBG.FlieOA.Flie.FileClick('" + model.FileId + "','" + model.FileName + "')\">";

//------ 修改文件夹样式

}

else {

htmlInfo += " <div onmousedown=\"DFBGFlie.Flie.FlieOperation.FileShowControl('" + model.FileExtName + "','" + model.FileId + "')\">";

}

//判断文件的类型，调用不同的文件图片样式

if (model.IsFolder == true) {//文件夹

htmlInfo += "<span class=\"fileicon dir-large\">";

}

else {

switch (model.FileExtName) {

case ".jpg":

case ".png":

case ".jpeg":

case ".gif":

case ".bmp":

//图片

htmlInfo += "<span class=\"fileicon fileicon-large-pic\">";

htmlInfo += "<img class=\"thumb fileicon-all\" style=\"width:100px;height:100px;\" data-url=\"/File/ImgSuoLue?FileId=" + model.FileId + "\" src=\"\">";

break;

case ".docx":

case ".doc":

htmlInfo += "<span class=\"fileicon fileicon-large-doc\">";

break;

case ".xls":

case ".xlsx":

htmlInfo += "<span class=\"fileicon fileicon-large-xls\">";

break;

case ".txt":

htmlInfo += "<span class=\"fileicon fileicon-large-txt\">";

break;

case ".rar":

htmlInfo += "<span class=\"fileicon fileicon-large-rar\">";

break;

case ".zip":

htmlInfo += "<span class=\"fileicon fileicon-large-zip\">";

break;

case ".pdf":

htmlInfo += "<span class=\"fileicon fileicon-large-pdf\">";

break;

case ".flv":

case ".mp4":

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

htmlInfo += "<span class=\"fileicon fileicon-large-video\">";

break;

case ".exe":

htmlInfo += "<span class=\"fileicon fileicon-large-exe\">";

break;

case ".psd":

htmlInfo += "<span class=\"fileicon fileicon-large-psd\">";

break;

case ".mp3":

case ".wma":

case ".wav":

htmlInfo += "<span class=\"fileicon fileicon-large-mp3\">";

break;

case ".ppt":

case ".pptx":

htmlInfo += "<span class=\"fileicon fileicon-large-ppt\">";

break;

case ".xmind":

htmlInfo += "<span class=\"fileicon fileicon-large-xmind\">";

break;

default:

htmlInfo += "<span class=\"fileicon fileicon-all\">";

break;

}

}

htmlInfo += "</span>";

htmlInfo += "</div>";

htmlInfo += "<label onmousedown=\"ClickSelCss('add" + model.FileId + "')\" class=\"checkbox\"></label>";

htmlInfo += "</div>";

if (model.FileName.length > 10) {

htmlInfo += "<div class=\"file-name\"><span title=\"" + model.FileName + "\" class=\"filename\">" + model.FileName.substr(0, 6) + "..." + model.FileExtName + "</span></div> ";

}

else {

htmlInfo += "<div class=\"file-name\"><span title=\"" + model.FileName + "\" href=\"javascript:void(0);\" class=\"filename\">" + model.FileName + "</span></div>";

}

htmlInfo += "</div>";

}

$("#FileShow").append(htmlInfo);

\_base.SetContainerHeight();

$("img").load(function () {

//图片默认隐藏

$(this).hide();

//使用fadeIn特效

$(this).stop().fadeIn("5000");

});

// 异步加载图片，实现逐屏加载图片

$(".grid-view-item").scrollLoading();

}

else {

\_base2.GetCont0();

}

}

});

},

MyListFileShowLoad: function (Type, Name, pageSize, pageIndex) {

var \_base = DFBGFlie.Flie.FlieOperation;

var \_base2 = DFBG.FlieOA.Flie;

\_base.AjaxList("/MyFile/MyFileShowLoad", { Type: Type, SearchName: Name, pageSize: pageSize, p: pageIndex },

function (re, status) {

var data = re.model;

if (status == "success") {

var htmlList = "";//列表展示

if (data.length != 0) {

//共多少个文件

\_base2.GetSumNum(re.SumCount, data.length, pageSize, pageIndex, "List");

for (var i = 0; i < data.length; i++) {

var model = data[i];

//\*\*\*\*List方式展示\*\*\*

htmlList += "<div class=\"g-clearfix list-view-item open-enable\">";

//判断文件的类型调不同的图标样式

if (model.IsFolder == true) {//文件夹

//---GYF 新增

htmlList += "<input type=\"checkbox\" onclick=\"YzCheAll()\" class=\"checkbox1\" name=\"cheName\" value=\"" + model.FileId + "\" />";

//--------

htmlList += "<div class=\"fileicon dir-small\"></div>";

//htmlList += "<span class=\"checkbox\"></span>";

}

else {

//非文件夹放选择框

htmlList += "<input type=\"checkbox\" onclick=\"YzCheAll()\" class=\"checkbox1\" name=\"cheName\" value=\"" + model.FileId + "\" />";

//图片

switch (model.FileExtName) {

case ".jpg":

case ".png":

case ".jpeg":

case ".jpg":

case ".gif":

case ".bmp":

//图片

htmlList += "<div class=\"fileicon fileicon-small-pic\"></div>";

break;

case ".docx":

case ".doc":

htmlList += "<div class=\"fileicon fileicon-small-doc\"></div>";

break;

case ".xls":

case ".xlsx":

htmlList += "<div class=\"fileicon fileicon-small-xls\"></div>";

break;

case ".txt":

htmlList += "<div class=\"fileicon fileicon-small-txt\"></div>";

break;

case ".rar":

htmlList += "<div class=\"fileicon fileicon-small-rar\"></div>";

break;

case ".zip":

htmlList += "<div class=\"fileicon fileicon-small-zip\"></div>";

break;

case ".pdf":

htmlList += "<div class=\"fileicon fileicon-small-pdf\"></div>";

break;

case ".flv":

case ".mp4":

case ".mkv":

case ".rmvb":

case ".avi":

case ".swf":

case ".wmv":

case ".3gp":

htmlList += "<div class=\"fileicon fileicon-small-video\"></div>";

break;

case ".exe":

htmlList += "<div class=\"fileicon fileicon-small-exe\"></div>";

break;

case ".psd":

htmlList += "<div class=\"fileicon fileicon-small-psd\"></div>";

break;

case ".mp3":

case ".wma":

case ".wav":

htmlList += "<div class=\"fileicon fileicon-small-mp3\"></div>";

break;

case ".ppt":

case ".pptx":

htmlList += "<div class=\"fileicon fileicon-small-ppt\"></div>";

break;

case ".xmind":

htmlList += "<div class=\"fileicon fileicon-small-xmind\"></div>";

break;

default:

htmlList += "<div class=\"fileicon fileicon-small\"></div>";

break;

}

}

htmlList += "<div style=\"width: 60%\" class=\"file-name\">";

htmlList += "<div class=\"text\">";

//判断是否是文件夹，如果是文件夹连接为打开文件夹

if (model.IsFolder == true) {

htmlList += "<a onclick=\"DFBG.FlieOA.Flie.FileClick('" + model.FileId + "','" + model.FileName + "')\" class=\"filename\" href=\"#\" title=\"" + model.FileName + "\">" + model.FileName + "</a></div>";

}

else {

htmlList += "<a onclick=\"DFBGFlie.Flie.FlieOperation.FileShowControl('" + model.FileExtName + "','" + model.FileId + "')\" class=\"filename\" href=\"#\" title=\"" + model.FileName + "\">" + model.FileName + "</a></div>";

}

htmlList += "<div class=\"operate\">";

htmlList += "<div style=\"position: absolute; top: 0px; padding-top: 0px; line-height: normal;\">";

//（文件夹不放下载按钮和分享按钮）

if (model.IsFolder != true) {

//分享

htmlList += "<a onclick=\"DFBGFlie.Flie.FlieOperation.ShareTransfer(" + model.FileId + ")\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"分享\" class=\"icon icon-share-blue\"></em> </span></a>";

//下载

htmlList += "<a href=\"/File/DownFile?FileId=" + model.FileId + "\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"下载\" class=\"icon icon-download-blue\"></em></span></a>";

}

else {

//放空标签占位

htmlList += "<a onclick=\"DFBGFlie.Flie.FlieOperation.ShareTransfer(" + model.FileId + ")\" class=\"g-button\" style=\"\"><span class=\"g-button-right\"><em title=\"分享\" class=\"icon icon-share-blue\"></em> </span></a>";

//htmlList += "<a class=\"g-button\"></a><a class=\"g-button\"></a>";

}

htmlList += "<span class=\"g-dropdown-button Baiparent\">";

//更多

htmlList += "<a onclick=\"MoreMethod(" + model.FileId + ")\" href=\"#\" class=\"g-button\"><span class=\"g-button-right\"></span></a>";

htmlList += "<span class=\"menu menu" + model.FileId + "\" style=\"width: 64px;display:none;\">";

htmlList += "<a href=\"#\" onclick=\"ReNameMethod(" + model.FileId + ",'menu" + model.FileId + "','" + model.FileName + "')\" class=\"g-button-menu rename\">重命名</a>";

htmlList += "<a href=\"#\" onclick=\"DFBGFlie.Flie.FlieOperation.Transfer(" + model.FileId + ",'parent')\" class=\"g-button-menu delete\">删除</a>";

htmlList += "</span></span>";

htmlList += " </div>";

htmlList += "</div>";

htmlList += "</div>";

htmlList += "<div style=\"width: 16%\" class=\"file-size\">" + DFBGFlie.Flie.FlieOperation.GetFileSize(model.FileSizeKb) + "</div>";

var date = new Date(parseInt(model.FileCreateTime.replace("/Date(", "").replace(")/", ""), 10));

htmlList += "<div style=\"width: 23%\" class=\"ctime\">" + date.Format("yyyy年MM月dd日 hh:mm") + "</div>";

htmlList += "</div>";

htmlList += "</div>";

}

$("#FileShowList").append(htmlList);

\_base.SetContainerHeight();

$(".module-list-view .list-view-item").hover(function () {

$(this).children(".file-name").children(".operate").toggle();

});

if ($(".list-view-item").length > 3) {

$(".list-view-item:last .operate .menu").css({ "left": "-40px", "top": "-50px" });

}

} else {

\_base2.GetCont0();

}

}

}

);

},

GetSumNum: function (SumCount, PartCount, pageSize, pageIndex, ShowType) {

maxnum = parseInt(SumCount / pageSize);

var ShowCookieType = getCookie("ShowType");

maxnum = parseFloat(SumCount % pageSize) > 0 ? maxnum + 1 : maxnum;

if (ShowType == "List") {

if (maxnum == pageIndex) {

$("#SumListNum").html("已全部加载，共" + SumCount + "个");

} else {

SumListNum += PartCount;

$("#SumListNum").html("已加载" + SumListNum + "个资源！");

}

}

else {

if (maxnum == pageIndex) {

$("#SumGridNum").html("已全部加载，共" + SumCount + "个");

} else {

SumGridNum += PartCount;

$("#SumGridNum").html("已加载" + SumGridNum + "个资源！");

}

}

if (ShowCookieType == "List") {

$("#SumListNum").show();

$("#SumGridNum").hide();

} else {

$("#SumGridNum").show();

$("#SumListNum").hide();

}

},

GetCont0: function () {

var htmlList = "<div class=\"aGMvFtb\">";

htmlList += "<div class=\"hqvyGW05 wegnv3JM\">";

htmlList += "<p class=\"ggEQwFb\">";

htmlList += " 您还没上传过文件哦";

htmlList += "</p>";

htmlList += "</div>";

htmlList += "</div>";

$("#FileShowList").html(htmlList);

$("#FileShow").html(htmlList);

},

FileClick: function (fileId) {

window.location.href = "/MyFile/MyFile?FileId=" + fileId;

},

InFileOnlode: function (fileId) {

DFBGFlie.Flie.FlieOperation.AjaxHtml("/File/FileALLNavigation", { parentFileId: fileId }, function (data) {

if (data != "") {

$("#TitleIds").append(data);

}

}, true);

},

isSelect: function () {

var FileisSelect = false;

var RenameStade = 0;

var swich = getCookie("ShowType");

//判断如果是大图标展示，获取大图标展示中选择的id

if (swich == "List") {

//获取要分享的文件id(获取list列表展示中的id)

var list = $("input[type=checkbox][name=cheName]");

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

FileisSelect = true;

RenameStade++;

}

}

} else {

var cssId = $(".item-active");

if (cssId.length > 0) {

FileisSelect = true;

}

RenameStade = cssId.length;

}

if (FileisSelect) {

$(".FlieOperation").css("display", "inline-block");

} else {

$(".FlieOperation").hide();

}

if (RenameStade == 1) {

$(".Renamed").show();

} else {

$(".Renamed").hide();

}

},

TuoMoVe: function (MoveFileId, treeviewId) {

var url = "../File/FileMoveTo";

var par = { FileIdList: MoveFileId, WhereId: treeviewId, MoveType: 1, Share: false }

EDUCAjax(par, function () {

$.closezhezhao('popup\_container', 'popup\_overlay');

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -77.5px;\"></div>");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-loading\"></i><span class=\"tip-msg\">正在复制文件，请稍候…</span></div></span></div>");

$(".module-tip").css("display", "block");

},

function (data) {

if (data == "suc") {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -48.5px\"></div>");

} else {

$(".module-tip").css("margin-left", "-48.5px");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-success\"></i><span class=\"tip-msg\">文件复制成功</span></div>");

$(".module-tip").css("display", "block");

window.location.reload();

setTimeout("$(\".module-tip\").hide()", 5000);

} else {

if ($(".module-tip").length == 0) {

$("body").append("<div class=\"module-tip\" style=\"top: 190px; left: 50%; margin-left: -48.5px\"></div>");

} else {

$(".module-tip").css("margin-left", "-48.5px");

}

$(".module-tip").html("<div class=\"tip-inner\"><i class=\"tip-icon tip-icon-success\"></i><span class=\"tip-msg\">文件复制失败</span></div>");

$(".module-tip").css("display", "block");

window.location.reload();

setTimeout("$(\".module-tip\").hide()", 5000);

}

}, url, "html", false, "", false);

}

}

parent.window.onresize = function () {

var swich = getCookie("ShowType");

if (swich == "List") {

$(".list-view-container").css({ "height": $(window).height() - 124 });

}

else {

$(".list-view-container").css({ "height": $(window).height() - 81 });

}

}

//全部文件

function AllFileMethod() {

window.location.href = "/MyFile/MyFile?FileId=";

}

//搜索文件

function SearchFileMethod() {

//将搜索框的值存在cookie中

setCookie("SearchInfo", $("#SearchId").val(), "d1");

window.location.reload();

}

//点击选择图标方式

function SelShowType(swich) {

setCookie("ShowType", swich, "d1");

if (swich == "List") {

$(".list-grid-switch").removeClass("grid-switched-on");

$(".list-grid-switch").addClass("list-switched-on");

$("#TypeShow").val("List");

$("#FileShow").hide();

$(".FileShowList").css({ "display": "block", "height": "100%" });

$("#FileShow").css({ "display": "none", "height": "auto" });

$("#FileShow").parent().css({ "height": "auto" });

$("#FileShowList").parent().css({ "height": "100%" });

$(".list-view-container").css({ "height": $(window).height() - 124 });

$("#SumListNum").show();

$("#SumGridNum").hide();

}

else {

$(".list-grid-switch").removeClass("list-switched-on");

$(".list-grid-switch").addClass("grid-switched-on");

$("#TypeShow").val("Tu");

$("#FileShow").show();

$(".FileShowList").css({ "display": "none", "height": "100%" });

$("#FileShowList").parent().css({ "height": "auto" });

$("#FileShow").css({ "display": "block", "height": "auto" });

$("#FileShow").parent().css({ "height": "100%" });

$(".list-view-container").css({ "height": $(window).height() - 81 });

$("#SumGridNum").show();

$("#SumListNum").hide();

}

DFBG.FlieOA.Flie.isSelect();

}

//样式左上角的复选框点击事件（未选中则点击选中，已选中则点击取消）

function ClickSelCss(elem) {

$("#" + elem).toggleClass("item-active");

DFBG.FlieOA.Flie.isSelect();

}

//list显示页面点击更多按钮

function MoreMethod(Id) {

$(".menu" + Id).show();

}

//全选方法

function AllCheck() {

var AllCheckId = document.getElementById("checkAll");//全选按钮id

var che = document.getElementsByName("cheName");//复选框name值

if (AllCheckId.checked == true) {

for (var i = 0; i < che.length; i++) {

che[i].checked = "checked";

}

}

else {

for (var i = 0; i < che.length; i++) {

che[i].checked = false;

}

}

DFBG.FlieOA.Flie.isSelect();

}

//验证当前页的复选框是否已经全选

function YzCheAll() {

//判断当前是否全选，动态给全选框状态

var list = $("input[type=checkbox][name=cheName]");

var ListCount = 0;

for (var i = 0; i < list.length; i++) {

if (list[i].checked) {

ListCount++;

}

}

if (list.length != 0) {

if ((ListCount == list.length)) {

$("#checkAll").attr("checked", "true");

} else {

$("#checkAll").removeAttr("checked");

}

DFBG.FlieOA.Flie.isSelect();

}

else {

$("#checkAll").removeAttr("checked");

}

}

**五、中间逻辑操作类**

using BLL;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity.Infrastructure;

using System.Drawing;

using System.Drawing.Imaging;

using System.IO;

using System.IO.Compression;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace EastElite.Controllers

{

public class FileController : Controller

{

BLL.Cookie GetCookie = new BLL.Cookie();

BLL.YunFile.MyFile YunFile = new BLL.YunFile.MyFile();

public static string VIR\_PATH = "/Upload/Yun";

public ActionResult Upload()

{

return View();

}

public ActionResult UploadMethod()

{

string fullFileName;

GetCookie.ExistCookie();

// 如果是文件上传请求

if( !string.IsNullOrWhiteSpace( this.Request.Params["name"] ) )

{

int chunk = Request.Params["chunk"] != null ? int.Parse( Request.Params["chunk"] ) : 0;

int chunks = Request.Params["chunks"] != null ? int.Parse( Request.Params["chunks"] ) : 0;

string fileName = Request.Params["uploadfilename"] != null ? Request.Params["uploadfilename"] : "1.jpg";

string parentFileId = Request.Params["parentFileId"] != null ? Request.Params["parentFileId"] : "";

bool Share = Request.Params["Share"] != null ? Request.Params["Share"].ToString() != "" ? true : false : false;

var Size = chunks \* 1024;

string KZname = System.IO.Path.GetExtension( fileName ).ToLower();//扩展名

//生成一个新的文件名称

if( chunk == 0 )

{

fullFileName = string.Format( "{0}", UploadPath( parentFileId, Share ) + "\\" + Guid.NewGuid() + KZname );

}

else

{

fullFileName = Session["filename"].ToString();

}

Session["filename"] = fullFileName;

FileStream fs = new FileStream( fullFileName, chunk == 0 ? FileMode.OpenOrCreate : FileMode.Append );

//write our input stream to a buffer

Byte[] buffer = null;

if( Request.ContentType == "application/octet-stream" && Request.ContentLength > 0 )

{

buffer = new Byte[Request.InputStream.Length];

Request.InputStream.Read( buffer, 0, buffer.Length );

}

else if( Request.ContentType.Contains( "multipart/form-data" ) && Request.Files.Count > 0 && Request.Files[0].ContentLength > 0 )

{

buffer = new Byte[Request.Files[0].InputStream.Length];

Request.Files[0].InputStream.Read( buffer, 0, buffer.Length );

}

else

{

return Content( "error" );

}

//write the buffer to a file.

fs.Write( buffer, 0, buffer.Length );

fs.Close();

string ReturnFileUrl = fullFileName.Substring( FileHelper.CurrentBaseDir.Length );

if( !ReturnFileUrl.StartsWith( @"\" ) )

{

ReturnFileUrl = @"\" + ReturnFileUrl;

}

ReturnFileUrl = ReturnFileUrl.Replace( @"\", @"/" );

string ReturnFileName = System.IO.Path.GetFileNameWithoutExtension( fileName.ToString() ) + KZname;

return Content( string.Format( "{0}|{1}|{2}", "上传成功！", ReturnFileUrl, ReturnFileName ) );

}

return Content( "error" );

}

/// <summary>

/// 上传文件时判断是否超过个人空间

/// </summary>

/// <param name="?">当前上传文件大小</param>

/// <returns></returns>

public bool IsOutFileSize( float FileSize )

{

bool result = false;

BLL.Cookie.TeUser U = GetCookie.GetUserCookie();

////获取当前人的空间大小

decimal UserSize = YunFile.GetDiskSize( U.userCode );

decimal SumSize = YunFile.GetlistByCode( U.userCode );

// 当前人的空间大小 + 当前上传文件大小

decimal num = SumSize + decimal.Parse( FileSize.ToString() );

// 是否超出500 M

if( num > UserSize )

{

result = true;

}

else

{

result = false;

}

return result;

}

/// <summary>

/// 获取文件上传的路径

/// </summary>

public string UploadPath( string parentFileId, bool Share )

{

//获取当前登陆人（当前登录人有对应文件如果不存在则创建）

//Cookie 验证

BLL.Cookie.TeUser U = GetCookie.GetUserCookie();

string url = "";

string GetFileMapPath = "";

switch( parentFileId )

{

case "":

case "0":

if( Share )

{

url = VIR\_PATH + "/School";

}

else

{

url = VIR\_PATH + "/" + U.userCode;

}

break;

default:

YunFile.GetFileMapPath( parentFileId, Share, ref GetFileMapPath );

url = VIR\_PATH + "/" + GetFileMapPath;

break;

}

if( !System.IO.Directory.Exists( Server.MapPath( url ) ) )

{

//不存在创建文件

System.IO.Directory.CreateDirectory( Server.MapPath( url ) );

}

return Server.MapPath( url );

}

/// <summary>

/// 向数据库插入上件的文件

/// </summary>

/// <returns></returns>

public JsonResult AddMyYunFiles( string files, string parentFileId )

{

Session["filename"] = null;

try

{

BLL.Cookie.TeUser U = GetCookie.GetUserCookie();

var json = new System.Web.Script.Serialization.JavaScriptSerializer();

var fileAddresses = json.Deserialize<Model.YUN\_FileInfo[]>( ( files ?? "[]" ) );

if( fileAddresses != null && fileAddresses.Length != default( Int32 ) )

{

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

for( int i = default( Int32 ); i < fileAddresses.Length; i++ )

{

Model.YUN\_FileInfo model = new Model.YUN\_FileInfo();

model.FileName = fileAddresses[i].FileName;

model.FileExtName = System.IO.Path.GetExtension( fileAddresses[i].FileName );

model.FileSizeKb = Convert.ToInt32( fileAddresses[i].FileSizeKb );

model.FileCreateTime = DateTime.Now;

model.IsFolder = false;

model.FileUrl = fileAddresses[i].FileUrl;

model.CreateId = U.userCode;

model.FileState = true;

model.IsShare = false;

model.ParentFileId = ( parentFileId == "0" ? "" : parentFileId );

model.CreateName = U.userName;

model.CreateUnitCode = U.unitCode;

model.ShareTypeId = fileAddresses[i].ShareTypeId;

if( !string.IsNullOrEmpty( model.ShareTypeId ) )

{

model.IsShare = true;

model.ShareTime = DateTime.Now;

model.CreateId = "Admin";

}

Db.YUN\_FileInfo.Add( model );

}

Db.SaveChanges();

}

return Json( new { s = true } );

}

return Json( new { s = false } );

}

catch( Exception ex )

{

return Json( new { s = false, msg = ex.Message } );

}

}

/// <summary>

/// 文件下载(直接点击下载，一次下载一个文件)

/// </summary>

/// <param name="filePath"></param>

/// <param name="fileName"></param>

/// <returns></returns>

public ActionResult DownFile( string FileId )

{

List<Model.YUN\_FileInfo> YunFileList = YunFile.GetFileByDown( "FileId", FileId.Trim( ',' ) );

string browser = Request.UserAgent.ToUpper();//获得浏览器用户代理信息

//if (browser.Contains("MS") == true && browser.Contains("IE") == true)//ie模式

if( browser.Contains( "FIREFOX" ) == false )//判断是否为火狐

{

YunFileList[0].FileName = Server.UrlEncode( YunFileList[0].FileName );

}

string absoluFilePath = Server.MapPath( System.Configuration.ConfigurationManager.AppSettings["AttachmentPath"] + YunFileList[0].FileUrl );

return new FileResult( absoluFilePath, YunFileList[0].FileName );

}

/// <summary>

/// 文件下载(选择多个文件)

/// </summary>

/// <param name="filePath"></param>

/// <param name="fileName"></param>

/// <returns></returns>

public ActionResult DownFileMore( string FileIds )

{

if( FileIds.Contains( "add" ) )

{

FileIds = FileIds.Replace( "add", "" );

}

List<Model.YUN\_FileInfo> YunFileList = YunFile.GetFileByDown( "FileId", FileIds.Trim( ',' ) );

string browser = "", absoluFilePath = "";

if( YunFileList.Count == 1 && YunFileList[0].IsFolder.ToString() != "True" )

{

browser = Request.UserAgent.ToUpper();//获得浏览器用户代理信息

//if (browser.Contains("MS") == true && browser.Contains("IE") == true)//ie模式

if( browser.Contains( "FIREFOX" ) == false )//判断是否为火狐

{

YunFileList[0].FileName = Server.UrlEncode( YunFileList[0].FileName );

}

absoluFilePath = Server.MapPath( System.Configuration.ConfigurationManager.AppSettings["AttachmentPath"] + YunFileList[0].FileUrl );

return new FileResult( absoluFilePath, YunFileList[0].FileName );

}

string[] file = new string[] { };

int sumnumber = 0;

if( YunFileList.Count > 0 )

{

for( var i = 0; i < YunFileList.Count; i++ )

{

if( YunFileList[i].IsFolder.ToString() == "True" )

{

List<Model.YUN\_FileInfo> YunFileList2 = YunFile.GetFileByDown( "ParentFileId", YunFileList[i].FileId.ToString() );

if( YunFileList2.Count > 0 )

{

foreach( var dr in YunFileList2 )

{

Array.Resize<string>( ref file, sumnumber + 1 );

file[sumnumber] = Server.MapPath( dr.FileUrl.ToString() );

sumnumber++;

}

}

}

else

{

Array.Resize<string>( ref file, sumnumber + 1 );

file[sumnumber] = Server.MapPath( YunFileList[i].FileUrl.ToString() );

sumnumber++;

}

}

}

BLL.Cookie.TeUser U = GetCookie.GetUserCookie();

string zipUrl = "/Upload/Yun/" + U.userCode + "/压缩.zip";

string fileName = "下载文件压缩包.zip";

ZipInfo.ZipSyem zip = new ZipInfo.ZipSyem();

zip.Zip( Server.MapPath( zipUrl ), 6, file );

browser = Request.UserAgent.ToUpper();//获得浏览器用户代理信息

if( browser.Contains( "FIREFOX" ) == false )//判断是否为火狐

{

fileName = Server.UrlEncode( fileName );

}

absoluFilePath = Server.MapPath( System.Configuration.ConfigurationManager.AppSettings["AttachmentPath"] + zipUrl );

return new FileResult( absoluFilePath, fileName );

}

/// <summary>

/// 删除文件

/// </summary>

/// <param name="FileIds"></param>

/// <returns></returns>

public string DelMore( string FileIds )

{

string result = "";

FileIds = FileIds.Trim( ',' );

if( FileIds.Contains( "add" ) )

{

FileIds = FileIds.Replace( "add", "" );

}

string isFile = "";

//删除磁盘中的数据

List<Model.YUN\_FileInfo> YunFileList = YunFile.GetFileByDown( "FileId", FileIds.Trim( ',' ) );

for( var z = 0; z < YunFileList.Count; z++ )

{

//判断当前需要删的是否是文件夹，如果是文件夹则需删除文件夹下的所有文件

string path = Server.MapPath( YunFileList[z].FileUrl.ToString() );

isFile = YunFileList[z].IsFolder.ToString();

if( isFile != "True" )

{

YunFile.FileDelete( path );

}

}

//删除数据库中的数据

if( FileIds.Contains( ',' ) )

{

//批量删除

string[] array = FileIds.Split( ',' );

for( var i = 0; i < array.Length; i++ )

{

if( YunFile.Delete( Convert.ToInt32( array[i] ) ) )

{

result = "suc";

}

}

}

else

{

if( YunFile.Delete( Convert.ToInt32( FileIds ) ) )//删除一个文件

{

result = "suc";

}

}

//删除文件夹下数据库中的文件数据

if( isFile == "True" )

{

DeleteFileDg( FileIds );

}

return result;

}

//清空回收站

public string ClearDel()

{

BLL.Cookie.TeUser U = GetCookie.GetUserCookie();

if( YunFile.ClearDel( U.userCode ) )

{

return "suc";

}

else

{

return "Err";

}

}

//真正删除文件

public string TrueDel( string FileIds )

{

BLL.Cookie.TeUser U = GetCookie.GetUserCookie();

string result = "";

FileIds = FileIds.Trim( ',' );

if( FileIds.Contains( ',' ) )

{

string[] array = FileIds.Split( ',' );

for( var i = 0; i < array.Length; i++ )

{

if( YunFile.TrueDel( int.Parse( array[i].ToString() ) ) )

{

result = "suc";

}

}

}

else

{

if( YunFile.TrueDel( int.Parse( FileIds ) ) )///真正删除文件

{

result = "suc";

}

}

return result;

}

/// <summary>

/// 递归删除文件夹下所有文件(数据库中的的文件)

/// </summary>

/// <param name="dir"></param>

public void DeleteFileDg( string Fileid )

{

List<Model.YUN\_FileInfo> YunFileList = YunFile.GetFileByDown( "ParentFileId", Fileid.ToString() );

var id = "";

for( var d = 0; d < YunFileList.Count; d++ )

{

id = YunFileList[d].FileId.ToString();

if( YunFileList[d].IsFolder.ToString() == "False" )

{

YunFile.FileDelete( Server.MapPath( YunFileList[d].FileUrl.ToString() ) );//删磁盘文件

}

YunFile.Delete( Convert.ToInt32( YunFileList[d].FileId ) );//删库中数据

DeleteFileDg( id );

}

}

public string FileALLNavigation( string parentFileId )

{

string str = "<li>";

if( parentFileId == "0" || parentFileId == "" )

{

str += "<a href=\"#\" >返回上一级</a>";

}

else

{

Model.YUN\_FileInfo YunFileList = YunFile.GetFileByUp( "FileId", int.Parse( parentFileId ) );

if( YunFileList != null )

{

str += "<a href=\"/MyFile/MyFile?FileId=" + YunFileList.ParentFileId + "\" >返回上一级</a>";

}

}

str += "<span class=\"historylistmanager-separator\">|</span>";

str += "<a onclick=\"AllFileMethod()\" href=\"#\">全部文件</a>";

str += "</li>";

return str + FileOnClick( parentFileId, false );

}

public string ShareFileALLNavigation( string parentFileId )

{

string str = "<li>";

if( parentFileId == "" || parentFileId == "0" )

{

str += "<a href=\"#\" >返回上一级</a>";

}

else

{

Model.YUN\_FileInfo YunFileList = YunFile.GetFileByUp( "FileId", int.Parse( parentFileId ) );

if( YunFileList != null )

{

str += "<a href=\"/SchoolShare/School?FileId=" + YunFileList.ParentFileId + "\" >返回上一级</a>";

}

}

str += "<span class=\"historylistmanager-separator\">|</span>";

str += "<a onclick=\"DFBGShare.FlieShare.Flie.ShareFileMethod()\" href=\"#\">全部文件</a>";

str += "</li>";

return str + FileOnClick( parentFileId, true );

}

/// <summary>

/// 获取文件上传的路径

/// </summary>

string FileHtml = "";

public string FileOnClick( string parentFileId, bool Share )

{

string isFile = "";

if( parentFileId == "" || parentFileId == "0" )

{

return "";

}

else

{

Model.YUN\_FileInfo YunFileList = YunFile.GetFileByUp( "FileId", int.Parse( parentFileId ) );

isFile = YunFileList.IsFolder.ToString();

if( Share )

{

FileHtml = "<li><span class=\"historylistmanager-separator-gt\">&gt;</span><a href=\"/SchoolShare/School?FileId=" + YunFileList.FileId + "\"><span title=\"ping\">" + YunFileList.FileName + "</span></a></li>" + FileHtml;

}

else

{

FileHtml = "<li><span class=\"historylistmanager-separator-gt\">&gt;</span><a href=\"/MyFile/MyFile?FileId=" + YunFileList.FileId + "\"><span title=\"ping\">" + YunFileList.FileName + "</span></a></li>" + FileHtml;

}

if( isFile == "True" )

{

FileOnClick( YunFileList.ParentFileId, Share );

}

}

return FileHtml;

}

/// <summary>

/// 文件重命名

/// </summary>

/// <returns></returns>

public string ReNameMethod( int FileId, string Name )

{

string result = "";

if( YunFile.ReName( FileId, Name ) )

{

result = "suc";

}

return result;

}

/// <summary>

/// 分享文件

/// </summary>

/// <returns></returns>

public string ShareMethod( string FileId, string ShareTypeId )

{

string result = "";

FileId = FileId.Trim( ',' );

if( FileId.Contains( "add" ) )

{

FileId = FileId.Replace( "add", "" );

}

if( FileId.Contains( ',' ) )

{

//批量分享

string[] array = FileId.Split( ',' );

for( var i = 0; i < array.Length; i++ )

{

if( YunFile.Share( int.Parse( array[i].ToString() ), ShareTypeId, GetCookie.GetUserCookie().unitCode ) )

{

result = "suc";

}

}

}

else

{

if( YunFile.Share( int.Parse( FileId ), ShareTypeId, GetCookie.GetUserCookie().unitCode ) )//分享一个文件

{

result = "suc";

}

}

return result;

}

/// <summary>

/// 文件取消分享

/// </summary>

/// <returns></returns>

public string IsShareMethods( string FileId )

{

string result = "";

FileId = FileId.Trim( ',' );

if( FileId.Contains( "add" ) )

{

FileId = FileId.Replace( "add", "" );

}

if( FileId.Contains( ',' ) )

{

//批量取消分享

string[] array = FileId.Split( ',' );

for( var i = 0; i < array.Length; i++ )

{

if( YunFile.IsShare( int.Parse( array[i].ToString() ) ) )

{

string ShareFileS = YunFile.GetModelByShareFileID( array[i].ToString() );

TrueDel( ShareFileS );

result = "suc";

}

}

}

else

{

if( YunFile.IsShare( int.Parse( FileId ) ) )

{

string ShareFileS = YunFile.GetModelByShareFileID( FileId );

TrueDel( ShareFileS );

result = "suc";

}

}

return result;

}

//缩略图加载

public FileContentResult ImgSuoLue( int FileId )

{

Model.YUN\_FileInfo FileInfo = YunFile.GetModel( FileId );

using( FileStream fs2 = new FileStream( Server.MapPath( FileInfo.FileUrl ), FileMode.Open, FileAccess.Read ) )

{

string Type = System.IO.Path.GetExtension( FileInfo.FileUrl ).ToLower().Replace( ".", "" );

//生成184\*180的缩略图

Image photo = Image.FromStream( fs2, true );

switch( Type )

{

case "gif":

MemoryStream gif = ImageShrink.ShrinkGifToStream( photo, 184, 180 );

photo.Dispose();

fs2.Close();

return File( gif.ToArray(), "image/gif" );

case "jpg":

MemoryStream jpeg = ImageShrink.ShrinkToStream( photo, 184, 180 );

photo.Dispose();

fs2.Close();

return File( jpeg.ToArray(), "image/jpeg" );

case "png":

MemoryStream png = ImageShrink.ShrinkToStream( photo, 184, 180 );

photo.Dispose();

fs2.Close();

return File( png.ToArray(), "image/png" );

case "bmp":

MemoryStream bmp = ImageShrink.ShrinkToStream( photo, 184, 180 );

photo.Dispose();

fs2.Close();

return File( bmp.ToArray(), "image/bmp" );

default:

MemoryStream jpg = ImageShrink.ShrinkToStream( photo, 184, 180 );

photo.Dispose();

fs2.Close();

return File( jpg.ToArray(), "image/jpeg" );

}

}

}

//大图像变小加载

public FileContentResult ImgLow( int FileId, int W, int H )

{

Model.YUN\_FileInfo FileInfo = YunFile.GetModel( FileId );

using( FileStream fs2 = new FileStream( Server.MapPath( FileInfo.FileUrl ), FileMode.Open, FileAccess.Read ) )

{

string Type = System.IO.Path.GetExtension( FileInfo.FileUrl ).ToLower().Replace( ".", "" );

//生成缩略图

Image photo = Image.FromStream( fs2, true );

int Width = photo.Width > W ? W : photo.Width;

int Height = photo.Height > H ? H : photo.Height;

switch( Type )

{

case "gif":

MemoryStream gif = ImageShrink.ShrinkGifToStream( photo, Width, Height );

photo.Dispose();

return File( gif.ToArray(), "image/gif" );

case "jpg":

MemoryStream jpeg = ImageShrink.ShrinkToStream( photo, Width, Height );

photo.Dispose();

return File( jpeg.ToArray(), "image/jpeg" );

case "png":

MemoryStream png = ImageShrink.ShrinkToStream( photo, Width, Height );

photo.Dispose();

return File( png.ToArray(), "image/png" );

case "bmp":

MemoryStream bmp = ImageShrink.ShrinkToStream( photo, Width, Height );

photo.Dispose();

return File( bmp.ToArray(), "image/bmp" );

default:

MemoryStream jpg = ImageShrink.ShrinkToStream( photo, Width, Height );

photo.Dispose();

return File( jpg.ToArray(), "image/jpeg" );

}

}

}

public ActionResult TxtDownload( string path, string FileName )

{

string txtname = "~/" + path.Split( '/' )[path.Split( '/' ).Length - 1].ToString();

string Txtpath = path.Substring( 0, path.Length - txtname.Length );

return File( new FileStream( AppDomain.CurrentDomain.BaseDirectory + path, FileMode.Open ), "text/plain", FileName );

}

/// <summary>

/// 移动文件、复制文件

/// </summary>

/// <param name="FileIdList">要移动的文件Id集合</param>

/// <param name="WhereId">目标文件夹</param>

/// <param name="MoveType">类型 1复制 2移动</param>

/// <returns></returns>

public string FileMoveTo( string FileIdList, string WhereId, string MoveType, bool Share )

{

if( FileIdList.Contains( "add" ) )

{

FileIdList = FileIdList.Replace( "add", "" );

}

int re; bool Copy = false;

List<Model.YUN\_FileInfo> YunFileList = YunFile.GetFileByDown( "FileId", FileIdList.Trim( ',' ) );

using( Model.NETDISKDBEntities Db = new Model.NETDISKDBEntities() )

{

foreach( var item in YunFileList )

{

if( WhereId == item.FileId.ToString() )

{

return "";

}

item.ParentFileId = WhereId;

item.FileState = true;

item.IsShare = false;

if( item.IsFolder == true )

{

Copy = true;

}

string FileUrl = item.FileUrl;

string FileMapPath = "";

YunFile.GetFileMapPath( WhereId, Share, ref FileMapPath );

FileMapPath = ( FileMapPath == "" ? "" : FileMapPath );

item.FileUrl = "/Upload/Yun/" + FileMapPath + "/" + Path.GetFileName( item.FileUrl );

//已有文件

if( FileHelper.ExitFile( Server.MapPath( item.FileUrl ) ) )

{

item.FileUrl = "/Upload/Yun/" + FileMapPath + "/" + Guid.NewGuid() + Path.GetExtension( item.FileUrl );

}

//文件复制

if( FileHelper.CopyFile( Server.MapPath( FileUrl ), Server.MapPath( item.FileUrl ), 1024 \* 1024 ) )

{

Copy = true;

}

int OldId = item.FileId;

//移动、复制

switch( MoveType )

{

case "1":

Db.YUN\_FileInfo.Add( item );

Db.SaveChanges();

if( item.IsFolder == true )

{

List<Model.YUN\_FileInfo> DownFileList = YunFile.GetFileByDown( "ParentFileId", OldId.ToString() );

for( int i = 0; i < DownFileList.Count; i++ )

{

FileMoveTo( DownFileList[i].FileId.ToString(), item.FileId.ToString(), MoveType, Share );

}

}

break;

case "2":

if( Copy )

{

DbEntityEntry<Model.YUN\_FileInfo> entry = Db.Entry<Model.YUN\_FileInfo>( item );

entry.State = System.Data.Entity.EntityState.Modified;

string OldFileMapPath = "";

YunFile.GetFileMapPath( OldId.ToString(), Share, ref OldFileMapPath );

Db.SaveChanges();

if( item.IsFolder == true )

{

if( !System.IO.Directory.Exists( Server.MapPath( "/Upload/Yun/" + FileMapPath ) ) )

{

//不存在创建文件

System.IO.Directory.CreateDirectory( Server.MapPath( "/Upload/Yun/" + FileMapPath ) );

}

List<Model.YUN\_FileInfo> DownFileList = YunFile.GetFileByDown( "ParentFileId", OldId.ToString() );

for( int i = 0; i < DownFileList.Count; i++ )

{

FileMoveTo( DownFileList[i].FileId.ToString(), item.FileId.ToString(), MoveType, Share );

}

if( System.IO.Directory.Exists( Server.MapPath( "/Upload/Yun/" + OldFileMapPath ) ) )

{

System.IO.Directory.Delete( Server.MapPath( "/Upload/Yun/" + OldFileMapPath ) );

}

}

else

{

YunFile.DeleteFile( Server.MapPath( FileUrl ) );

}

}

break;

default:

break;

}

}

}

return "suc";

}

//还原文件

public string Reduction( string FileId )

{

string result = "";

FileId = FileId.Trim( ',' );

if( FileId.Contains( "add" ) )

{

FileId = FileId.Replace( "add", "" );

}

if( FileId.Contains( ',' ) )

{

//批量还原

string[] array = FileId.Split( ',' );

for( var i = 0; i < array.Length; i++ )

{

if( YunFile.Reduction( int.Parse( array[i].ToString() ) ) )

{

result = "suc";

}

}

}

else

{

if( YunFile.Reduction( int.Parse( FileId ) ) )

{

result = "suc";

}

}

return result;

}

}

public class FileResult : ActionResult

{

/// <summary>

/// 该类继承了ActionResult，通过重写ExecuteResult方法，进行文件的下载

/// </summary>

private readonly string \_filePath;//文件路径

private readonly string \_fileName;//文件名称

public FileResult( string filePath, string fileName )

{

\_filePath = filePath;

\_fileName = fileName;

}

public override void ExecuteResult( ControllerContext context )

{

string fileName = \_fileName;

HttpResponseBase response = context.HttpContext.Response;

if( File.Exists( \_filePath ) )

{

FileStream fs = null;

byte[] fileBuffer = new byte[1024 \* 100];//每次读取1024字节大小的数据

try

{

using( fs = File.OpenRead( \_filePath ) )

{

long totalLength = fs.Length;

response.ContentType = "application/octet-stream";

response.AddHeader( "Content-Disposition", "attachment; filename=" + fileName );

while( totalLength > 0 && response.IsClientConnected )//持续传输文件

{

int length = fs.Read( fileBuffer, 0, fileBuffer.Length );//每次读取1024个字节长度的内容

fs.Flush();

response.OutputStream.Write( fileBuffer, 0, length );//写入到响应的输出流

response.Flush();//刷新响应

totalLength = totalLength - length;

}

response.Close();//文件传输完毕，关闭相应流

}

}

catch( Exception ex )

{

response.Write( ex.Message );

}

finally

{

if( fs != null )

fs.Close();//最后记得关闭文件流

}

}

}

}

}