using Dal;

using DTO;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Bll

{

public class CategorysFunctions

{

static DataEntities d = new DataEntities();

static List<int> li = new List<int>();

//• הוספת קטגוריה.

//v

public static void AddCategory(CategoryDTO cd)

{

Category c= new Category();

c = CategoryDTO.ConvertFromDTO(cd);

d.Category.Add(c);

d.SaveChanges();

}

//• החזרת קטגורית אב ע"פ קוד קטגוריה.

//x

public static int GetBasicCategoryByCategoryId(int id)

{

int basicId = d.Category.Where(x => x.CategoryId == id).Select(x=>x.BasicCategoryId).FirstOrDefault();

return basicId;

}

//החזרת רשימת קטגוריות על פי רשימת קודים

public static List<Category> GetListCategoryByListCategoryId(List<int> idl)

{

List<Category> lc = new List<Category>();

for (int i = 0; i < idl.Count(); i++)

{

Category c = new Category();

c = d.Category.FirstOrDefault(x => x.CategoryId == idl[i]);

lc.Add(c);

}

return lc;

}

public static void DeleteCategoryFromSpezificListCategoryById(List<Category> slc,int id)

{

int i;

for ( i= 0; i < slc.Count() && slc[i].CategoryId != id; i++);

if(i < slc.Count())

{

slc.Remove(slc[i]);

}

}

//• החזרת רשימת קטגוריות צאצאים ע"פ קטגורית אב חוץ מהבן שכבר נכתב.

//x

public static List<int> GetCategorysListByBasicCategoryId(int id)

{

foreach (var item in d.Category)

{

if (item.BasicCategoryId == id)

{

li.Add(item.CategoryId);

return GetCategorysListByBasicCategoryId(item.CategoryId);

}

}

return li;

}

}

}