# ENTITY FRAMEWORK UTILIZANDO STORED PROCEDURE

GET

public static Result GetSinFotoFirma(int idGrupo)

        {

            Result result = new Result();

            try

            {

                using (CAEEntities context = new CAEEntities())

                {

                    var alumnos = context.AlumnoGetSinFotoFirma().ToList();

                    result.Objects = new List<object>();

                    if (alumnos != null)

                    {

                        foreach (var obj in alumnos)

                        {

                            ML.Alumno alumno = new ML.Alumno();

                            alumno.Matricula = obj.Matricula;

                            alumno.Nombre = obj.Nombre;

                            alumno.ApellidoPaterno = obj.ApellidoPaterno;

                            alumno.ApellidoMaterno = obj.ApellidoMaterno;

                            alumno.CURP = obj.CURP;

                            alumno.Telefono= obj.Telefono;

                            result.Objects.Add(alumno);

                        }

                        result.Correct = true;

                    }

                    else

                    {

                        result.Correct = false;

                        result.ErrorMessage = "No se encontraron registros.";

                    }

                }

            }

            catch (Exception ex)

            {

                result.Correct = false;

                result.ErrorMessage = ex.Message;

            }

            return result;

        }

Add:

public static ML.Result Add(ML.PlantelCarreraMateria plantelCarreraMateria)

{

ML.Result result = new ML.Result();

try

{

using (DL.CAEEntities context = new DL.CAEEntities())

{

var query = context.PlantelCarreraMateriaAdd(plantelCarreraMateria.PlantelCarrera.Id, plantelCarreraMateria.Materia.IdMateria, plantelCarreraMateria.Usuario.IdUser);

if (query >= 1)

{

result.Correct = true;

}

else

{

result.Correct = false;

result.ErrorMessage = "No se insertó el registro";

}

result.Correct = true;

}

}

catch (Exception ex)

{

result.Correct = false;

result.ErrorMessage = ex.Message;

}

return result;

}

Update:

public static Result Update (ML.Credencial credencial) //12/12/2019 No se elimina la credencial, se cambia

{

Result result = new Result();

try

{

using (DL.CAEEntities context = new DL.CAEEntities())

{

var updateResult = context.CredencialUpdateStatus(credencial.Matricula);

if (updateResult >= 1)

{

result.Correct = true;

}

else

{

result.Correct = false;

result.ErrorMessage = "No se actualizó el status de la credencial";

}

}

}

catch (Exception ex)

{

Error.Add(ex);

result.Correct = false;

result.ErrorMessage = ex.Message;

}

return result;

}

public static Result Update(ML.Credencial credencial){

Result result = new Result();

try

{

using (DL.CAEEntities context = new DL.CAEEntities())

{

var updateResult = context.CredencialUpdate(credencial.Matricula);

if (updateResult >= 1)

{

result.Correct = true;

}

else

{

result.Correct = false;

result.ErrorMessage = "No se actualizó el status de la credencial";

}

}

}

catch (Exception ex)

{

result.Correct = false;

result.ErrorMessage = ex.Message;

}

return result;

}

Delete:

public static Result Delete(ML.PlantelReporteCredencial plantelReporteCredencial)

{

Result result = new Result();

try

{

using (CAEEntities context = new CAEEntities())

{

Var query = context.PlantelReporteCredencialDelete(plantelReporteCredencial.IdPlantelReporteCredencial);

if (query >= 1)

{

result.Correct = true;

}

else

{

result.Correct = false;

result.ErrorMessage = "No se eliminó el registro";

}

result.Correct = true;

}

}

catch (Exception ex)

{

result.Correct = false;

result.ErrorMessage = ex.Message;

}

return result;

}

BL.Producto

public static ML.Result GetByIdEF(ML.Producto producto)

{

ML.Result result = new ML.Result();

try

{

using (DL\_EF.EcommerceJRodriguezEntities context = new DL\_EF.EcommerceJRodriguezEntities())

{

var query = context.ProductoGetById(producto.IdProducto).FirstOrDefault();

producto.Subcategoria = new ML.Subcategoria();

producto.Subcategoria.IdSubcategoria = query.IdSubcategoria;

producto.Proveedor = new ML.Proveedor();

producto.Proveedor.IdProveedor = query.IdProveedor;

producto.Nombre = query.Nombre;

producto.Precio = query.Precio;

result.Object = producto;

if (query != null)

{

result.Correct = true;

}

else

{

result.Correct = false;

result.ErrorMessage = "No se encontraron registros.";

}

}

}

catch (Exception e)

{

result.Correct = false;

result.ErrorMessage = e.Message;

}

return result;

}

# ENTITY FRAMEWORK UTILIZANDO LINQ

GET:

public static Result GetPlantelCarrera()

{

Result result = new Result();

try

{

using (CAEEntities context = new CAEEntities())

{

var query = (from plantelCarrera in context.Plantel\_Carrera

join plantel in context.Plantels on plantelCarrera.IdPlantel equals plantel.IdPlantel

join carrera in context.Carreras on plantelCarrera.IdCarrera equals carrera.IdCarrera

where plantelCarrera.IdCarrera == carrera.IdCarrera

where plantelCarrera.IdPlantel == plantel.IdPlantel

select new { NombreCorto = plantel.NombreCorto, Nombre = carrera.Nombre, IdPlantelCarrera = plantelCarrera.IdPlantelCarrera });

result.Objects = new List<object>();

if (query != null && query.ToList().Count > 0)

{

foreach (var obj in query)

{

ML.PlantelCarrera pc = new ML.PlantelCarrera();

pc.NombreCorto = obj.NombreCorto;

pc.Nombre = obj.Nombre;

pc.Id = obj.IdPlantelCarrera;

result.Objects.Add(pc);

}

result.Correct = true;

}

else

{

result.Correct = false;

result.ErrorMessage = "No se encontraron registros";

}

}

}

catch (Exception ex)

{

result.Correct = false;

result.ErrorMessage = ex.Message;

}

return result;

}

ADD:

public static void Add (ML.Grupo grupo)

        {

            Result result = new Result();

            try

            {

                using (CAEEntities context = new CAEEntities())

                {

                    DL.Grupo grupoDL = new DL.Grupo();

                    grupoDL.Nombre = grupo.Nombre;

                    grupoDL.Generacion = grupo.Generacion;

                    grupoDL.IdTurno = grupo.IdTurno;

                    grupoDL.IdPlantelCarrera = grupo.IdPlanteCarrera;

                    context.Grupoes.Add(grupoDL);

                    context.SaveChanges();

                }

            }

            catch (Exception ex)

            {

                Error.Add(ex);

                result.Correct = false;

                result.ErrorMessage = ex.Message;

            }

        }

UPDATE

public static Result Update (ML.Grupo grupo)

        {

            Result result = new Result();

            try

            {

                using (CAEEntities context = new CAEEntities())

                {

                    var query = (from a in context.Grupoes

                                 where a.IdGrupo == grupo.IdGrupo

                                 select a).SingleOrDefault();

                    if (query != null)

                    {

                        query.Nombre = grupo.Nombre;

                        query.Generacion = grupo.Generacion;

                        query.IdTurno = grupo.IdTurno;

                        query.IdPlantelCarrera = grupo.IdPlanteCarrera;

                        context.SaveChanges();

                        result.Correct = true;

                    }

                    else

                    {

                        result.Correct = false;

                        result.ErrorMessage = "No se encontró el grupo" + grupo.IdGrupo;

                    }

                }

            }

DELETE

public static Result Delete (ML.Grupo grupo)

        {

            Result result = new Result();

            try

            {

                using (CAEEntities context = new CAEEntities())

                {

                    var query = (from a in context.Grupoes

                                 where a.IdGrupo == grupo.IdGrupo

                                 select a).First();

                    context.Grupoes.Remove(query);

                    context.SaveChanges();

                }

            }

            catch (Exception ex)

            {

                Error.Add(ex);

                result.Correct = false;

                result.ErrorMessage = ex.Message;

            }

            return result;

        }