namespace Core.DAL.Adapter

{

/// <summary>

/// adapter for tax return

/// </summary>

internal class TaxReturnAdapter

{

#region Field Names

internal const string TAX\_RETURN\_ID = "TaxReturnId";

internal const string BUSINESS\_ID = "BusinessId";

internal const string BATCH\_ID = "BatchId";

internal const string PERIOD\_ENDING\_DATE = "PeriodEndingDate";

internal const string POST\_MARK\_DATE = "PostMarkDate";

internal const string PAYMENT = "Payment";

internal const string PAYMENT\_TYPE\_ID = "PaymentTypeId";

internal const string IS\_MASTER = "IsMaster";

internal const string IS\_DIRTY = "IsDirty";

internal const string SYSTEM\_TIMESTAMP = "SystemTimestamp";

#endregion

/// <summary>

/// Gets the tax return sections.

/// </summary>

/// <returns></returns>

internal static ReadOnlyCollection<TaxReturnSection> GetTaxReturnSections(int? taxReturnId)

{

var sections = new List<TaxReturnSection>();

using (var conn = new SqlConnection(DataAccess.ConnectionString))

{

conn.Open();

using (var cmd = DataAccess.CreateCommand("[Core].[TaxReturnSectionSelectByTaxReturnId]"))

{

cmd.Connection = conn;

cmd.Parameters.Add(new SqlParameter(TAX\_RETURN\_ID, SqlDbType.Int)).Value = taxReturnId;

using (var rdr = cmd.ExecuteReader())

while (rdr.Read())

{

var taxReturnSection = new TaxReturnSection();

TaxReturnSectionAdapter.ReadTaxReturnSection(ref taxReturnSection, rdr);

sections.Add(taxReturnSection);

}

}

}

return new ReadOnlyCollection<TaxReturnSection>(sections);

}

/// <summary>

/// Saves the specified tax return section.

/// </summary>

/// <param name="taxReturnSection">The tax return section.</param>

internal static void Save(ref TaxReturnSection taxReturnSection)

{

if (taxReturnSection.TaxReturnSectionId == null)

Insert(ref taxReturnSection);

else

Update(ref taxReturnSection);

}

/// <summary>

/// Inserts the specified tax return section.

/// </summary>

/// <param name="taxReturnSection">The tax return section.</param>

private static void Insert(ref TaxReturnSection taxReturnSection)

{

using (var conn = new SqlConnection(DataAccess.ConnectionString))

{

conn.Open();

using (var cmd = DataAccess.CreateCommand("[Core].[TaxReturnSectionInsert]"))

{

cmd.Connection = conn;

SetSqlParameters(taxReturnSection, cmd);

taxReturnSection.TaxReturnSectionId = (int)cmd.ExecuteScalar();

}

}

}

/// <summary>

/// Updates the specified tax return section.

/// </summary>

/// <param name="taxReturnSection">The tax return section.</param>

private static void Update(ref TaxReturnSection taxReturnSection)

{

throw new **NotImplementedException**();

}

/// <summary>

/// Sets the SQL parameters.

/// </summary>

/// <param name="taxReturnSection">The tax return section.</param>

/// <param name="cmd">The CMD.</param>

private static void SetSqlParameters(TaxReturnSection taxReturnSection, SqlCommand cmd)

{

cmd.Parameters.Add(new SqlParameter(TAX\_RETURN\_SECTION\_ID, SqlDbType.Int)).Value = (object) taxReturnSection.TaxReturnSectionId ?? DBNull.Value;

cmd.Parameters.Add(new SqlParameter(TAX\_RETURN\_ID, SqlDbType.Int)).Value = taxReturnSection.TaxReturnId;

cmd.Parameters.Add(new SqlParameter(TAX\_CATEGORY\_ID, SqlDbType.Int)).Value = taxReturnSection.TaxCategoryId;

}