

```
using System;
using System.Data;
using Caissa.Classes;

namespace Caissa.Persistence
{
    public class SourceTypeDB
    {
        public int source_type_id
        {
            get;
            set;
        }

        #region Insert Source Type
        private void Insert(SourceType type)
        {
            string storedprocedure = "sp_InsertSourceType";

            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

            command = Mapped.Command(storedprocedure, connection);
            command.CommandType = CommandType.StoredProcedure;
            command.Parameters.Add(Mapped.Parameter("p_sot_title", type.Title,
                DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sot_description",
                type.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sot_id", "", DbType.Int32,
                ParameterDirection.Output));
            command.ExecuteNonQuery();

            int id = 0;

            foreach (var parameter in command.Parameters)
            {
                IDbDataParameter p = (IDbDataParameter)parameter;
                if (p.Direction == ParameterDirection.Output)
                    if (p.ParameterName == "p_sot_id")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            source_type_id = id;
        }
        #endregion

        #region Update Source Type
        private void Update(SourceType type)
        {
            string storedprocedure = "sp_UpdateSourceType";

            System.Data.IDbConnection connection;
```

```
System.Data.IDbCommand command;

connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

command = Mapped.Command(storedprocedure, connection);
command.CommandType = CommandType.StoredProcedure;
command.Parameters.Add(Mapped.Parameter("p_sot_id", type.Id,
    DbType.Int32, ParameterDirection.Input));
command.Parameters.Add(Mapped.Parameter("p_sot_title", type.Title,
    DbType.String, ParameterDirection.Input));
command.Parameters.Add(Mapped.Parameter("p_sot_description",
    type.Description, DbType.String, ParameterDirection.Input));
command.ExecuteNonQuery();

connection.Close();
}
#endregion

#region Delete Source Type
private void Delete(int type_id)
{
    string storedprocedure = "sp_DeleteSourceType";

    System.Data.IDbConnection connection;
    System.Data.IDbCommand command;

    connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

    command = Mapped.Command(storedprocedure, connection);
    command.CommandType = CommandType.StoredProcedure;
    command.Parameters.Add(Mapped.Parameter("p_sot_id", type_id,
        DbType.Int32, ParameterDirection.Input));
    command.ExecuteNonQuery();

    connection.Close();
}
#endregion

#region Select Source Type
public DataSet Select()
{
    System.Data.IDbConnection connection;
    System.Data.IDbCommand command;
    System.Data.IDataAdapter adapter;

    DataSet ds = new DataSet();

    string sql = string.Empty;

    connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

    sql = "select sot_id, sot_title, sot_description"
        + " from tbl_source_type";

    command = Mapped.Command(sql, connection);
```

---

```
        adapter = Mapped.Adapter(command);
        adapter.Fill(ds);

        connection.Close();

        return ds;
    }
#endregion

public SourceTypeDB()
{
    //
    // TODO: Add constructor logic here
    //
}
}
```