Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут ім. Ігоря Сікорського”

Кафедра автоматизації проектування енергетичних процесів і систем

Звіт

з лабораторної роботи №5

з дисципліни:

«Проектування систем з розподіленими бащами даних»

Виконали:

група ТР-01мп  
Кривошеев Е.В.

Лєсніченко Р.А.

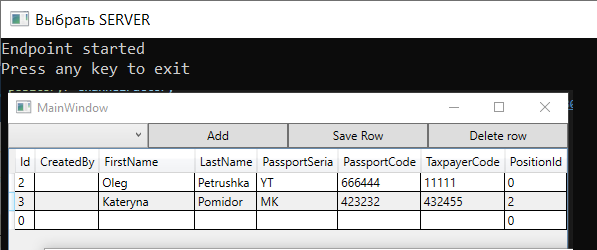
Перевірив:

Варава І.А.

Київ – 2020

Завдання: Триланкова система CORBA.

Хід роботи



Код клієнта

public class MainViewModel : INotifyPropertyChanged

{

IEmployeeRepository \_repository;

public ObservableCollection<Employee> \_rows;

public ObservableCollection<Employee> Rows { get => \_rows; set { \_rows = value; OnPropertyChanged(); } }

private Employee \_selectedRow;

public Employee SelectedRow

{

get => \_selectedRow; set { \_selectedRow = value;OnPropertyChanged(); }

}

private ICommand \_saveSelectedRowCommand;

public ICommand SaveSelectedRowCommand { get => \_saveSelectedRowCommand; private set { \_saveSelectedRowCommand = value; OnPropertyChanged(); } }

private ICommand \_deleteSelectedRowCommand;

public ICommand DeleteSelectedRowCommand { get => \_deleteSelectedRowCommand; private set { \_deleteSelectedRowCommand = value; OnPropertyChanged(); } }

private ICommand \_addNewRowCommand;

public ICommand AddNewRowCommand { get => \_addNewRowCommand; private set { \_addNewRowCommand = value;OnPropertyChanged(); } }

private bool \_isInsertMode = false;

public MainViewModel()

{

try

{

AddNewRowCommand = new RelayCommand(AddNew);

SaveSelectedRowCommand = new RelayCommand(SaveSelected);

DeleteSelectedRowCommand = new RelayCommand(DeleteSelected);

// Регистрируем канал IIOP.

IiopClientChannel channel = new IiopClientChannel();

ChannelServices.RegisterChannel(channel);

// Адрес CORBA-сервера.

string addressString = "iiop://localhost:1234/hello";

// Получаем ссылку на клиентский прокси.

\_repository = (IEmployeeRepository)RemotingServices.Connect(typeof(IEmployeeRepository), addressString);

var objectsSerialized = \_repository.GetObjects();

Rows = new ObservableCollection<Employee>(JsonConvert.DeserializeObject<IList<Employee>>(objectsSerialized) );

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void SaveSelected(object obj = null)

{

if (SelectedRow == null)

{

MessageBox.Show("Can`t save. Row doesn`t selected");

return;

}

else if (\_isInsertMode)

{

\_repository.Create(SelectedRow);

\_isInsertMode = false;

MessageBox.Show("Item created");

}

else

{

\_repository.Update(SelectedRow);

MessageBox.Show("Item updated");

}

}

public void AddNew(object obj = null)

{

\_isInsertMode = true;

Rows.Add(new Employee());

}

private void DeleteSelected(object obj = null)

{

\_repository.Remove(SelectedRow);

Rows.Remove(SelectedRow);

MessageBox.Show("Item deleted");

}

public event PropertyChangedEventHandler PropertyChanged;

public void OnPropertyChanged([CallerMemberName]string prop = "")

{

if (PropertyChanged != null)

PropertyChanged(this, new PropertyChangedEventArgs(prop));

}

}

Код сервера

static void Main(string[] args)

{

Console.Title = "SERVER";

try

{

int serverPort = 1234;

IiopChannel channel = new IiopChannel(serverPort);

ChannelServices.RegisterChannel(channel);

EmployeeRepository repository = new EmployeeRepository();

string objectURI = "hello";

RemotingServices.Marshal(repository, objectURI);

Console.WriteLine("Endpoint started");

Console.WriteLine("Press any key to exit");

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.ReadKey();

}

Репозиторій:

public class EmployeeRepository : MarshalByRefObject, IEmployeeRepository

{

private string \_connectionString;

private DbContextOptions<LvivFilialDBContext> dbOptions;

public override object InitializeLifetimeService()

{

return null;

}

public EmployeeRepository()

{

\_connectionString = "Server = localhost\\SQLExpress; Database = LvivFilialDB; Trusted\_Connection = True;";

dbOptions = new DbContextOptionsBuilder<LvivFilialDBContext>()

.UseSqlServer(\_connectionString).Options;

}

public void Create(Employee obj)

{

if (obj == null) throw new ArgumentNullException();

using (var context = new LvivFilialDBContext(dbOptions))

{

context.Add(obj);

context.SaveChanges();

}

}

public void Remove(Employee obj)

{

if (obj == null) throw new ArgumentNullException();

using (var context = new LvivFilialDBContext(dbOptions))

{

context.Remove(obj);

context.SaveChanges();

}

}

public string GetObjects()

{

using (var context = new LvivFilialDBContext(dbOptions))

{

return JsonConvert.SerializeObject(context.Employee.ToList());

//return context.Employee.ToList();

}

}

public void Update(Employee obj)

{

if (obj == null) throw new ArgumentNullException();

using (var context = new LvivFilialDBContext(dbOptions))

{

context.Update(obj);

context.SaveChanges();

}

}

}