Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  |  |  | М.О. Димухаметов |
|  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ15–16Б, 031510065 |  |  |  | Т.В. Радионов |
|  | номер группы, зачетной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2018

**ОГЛАВЛЕНИЕ**

[1 Постановка задачи 3](#_Toc527580196)

[2 Листинг 3](#_Toc527580197)

[3 Примеры работы программы 6](#_Toc527580198)

[4 Выводы 7](#_Toc527580199)

# Постановка задачи

*Бизнес-операция D*

Эта бизнес-операция может отвечать на два последующих запроса информации. Чтобы это сделать, она использует базу данных сотрудников компании (согласно документации ниже).

1. Определить по ID сотрудника и вернуть 1 или 0 в зависимости от того, является ли сотрудник родственником владельца.
2. По email адресу определить и вернуть 1 или 0 в зависимости от того, является ли адрес внешним.

*Бизнес-операция E*

Бизнес-операция записывает данные в два разных формата файла. Компонент, вызвавший бизнес-операцию E, должен передать ей:

1. Имя файла, в который должно записаться письмо.
   1. Если файл уже существует, добавить туда письмо.
2. Требуемое содержание.

Два формата файла:

1. Файл адресов
   1. Обязательные данные:
      1. Дата и время, когда запись попала в файл
      2. Email адрес.
      3. Является ли адрес внешним.
      4. Является ли отправитель родственником владельца.
   2. Каждый элемент информации разделен пробелом.
2. Файл писем
   1. Обязательные данные:
      1. Имя сотрудника.
      2. Company ID сотрудника.
      3. Email адрес.
      4. Текст письма.
   2. Каждый элемент информации помещается на отдельную строку.
      1. Дата и время, когда запись попала в файл, также помещается в отдельной строке до остальной информации.

# Листинг

Бизнес-операция D

Class BPOperation.GetEmployeeDetails Extends Ens.BusinessOperation  
{  
  
Parameter ADAPTER = "EnsLib.SQL.OutboundAdapter";  
  
Property Adapter As EnsLib.SQL.OutboundAdapter;  
  
Parameter INVOCATION = "Queue";  
  
XData MessageMap  
{  
<MapItems>  
  <MapItem MessageType="Msg.ReqEmployeeID">  
    <Method>CheckOnRelationshipToOwnerById</Method>  
  </MapItem>  
  <MapItem MessageType="Msg.ReqEmployeeEmail">  
    <Method>CheckOnExternalEmail</Method>  
  </MapItem>  
</MapItems>  
}  
  
Method CheckOnRelationshipToOwnerById(req As Msg.ReqEmployeeID, Output res As Msg.ResEmployeeRelationshipToOwner) As %Status  
{  
 set res = ##class(Msg.ResEmployeeRelationshipToOwner).%New()  
   
 #dim result As EnsLib.SQL.GatewayResultSet  
 set st = ..Adapter.ExecuteQuery(.result,"select RelationshipToOwner from SQLUser.Employee where id = ?", req.ID)  
 if $$$ISERR(st) set ^err = $system.Status.GetErrorText(st)  
 if result.Next()  
 {  
 set st = result.%Get("RelationshipToOwner")  
 if st = "None" set res.RelationshipToOwner = 0  
 else  set res.RelationshipToOwner = 1  
 }  
 else  
 {  
 return $$$ERROR($$$GeneralError, "Employee with id: "\_req.ID\_" not found!")  
 }  
   
 return $$$OK  
}  
  
Method CheckOnExternalEmail(req As Msg.ReqEmployeeEmail, Output res As Msg.ResEmployeeExternalEmail) As %Status  
{  
 set res = ##class(Msg.ResEmployeeExternalEmail).%New()  
   
 #dim result As EnsLib.SQL.GatewayResultSet  
 set st = ..Adapter.ExecuteQuery(.result,"select Email from SQLUser.Employee where Email = ?", req.Email)  
 if $$$ISERR(st) set ^err = $system.Status.GetErrorText(st)  
 if result.Next() = 1  
 {  
 set res.ExternalEmail = 0  
 }  
 elseif  result.Next() = 0 // Записей не найдено  
 {  
 set res.ExternalEmail = 1  
 }  
 else  
 {  
 return $$$ERROR($$$GeneralError, "Exception!")  
 }  
   
 return $$$OK  
}  
  
}

Бизнес-операция E

Class BPOperation.WriteFile Extends Ens.BusinessOperation  
{  
  
Parameter ADAPTER = "EnsLib.File.OutboundAdapter";  
  
Property Adapter As EnsLib.File.OutboundAdapter;  
  
Parameter INVOCATION = "Queue";  
  
XData MessageMap  
{  
<MapItems>  
  <MapItem MessageType="Msg.ReqFileData">  
    <Method>AddData</Method>  
  </MapItem>  
</MapItems>  
}  
  
Method AddData(req As Msg.ReqFileData, Output res As Msg.ResWriteSuccess) As %Status  
{  
 #dim e As %Exception.StatusException  
 set res = ##class(Msg.ResWriteSuccess).%New()  
 try   
 {  
 $$$ThrowOnError(..Adapter.PutLine("addresses.txt", req.SendTime))  
 $$$ThrowOnError(..Adapter.PutLine("addresses.txt", req.EmployeeEmail \_ ";" \_ req.ExternalEmail \_ ";" \_ req.RelationshipToOwner \_ $$$NL))  
 $$$ThrowOnError(..Adapter.PutLine("letters.txt", req.EmployeeName \_ ";" \_ req.EmployeeID \_ ";" \_ req.EmployeeEmail \_ ";" \_ req.Text \_ $$$NL))  
 }  
 catch e  
 {  
  
 set res.WriteSuccess = 0  
 return $System.Status.AppendStatus(e.AsStatus(),$$$ERROR($$$GeneralError,"SAFAFSDGDS"))  
 }  
 set res.WriteSuccess = 1  
 return $$$OK  
}  
  
}

# Примеры работы программы

Бизнес-операция D

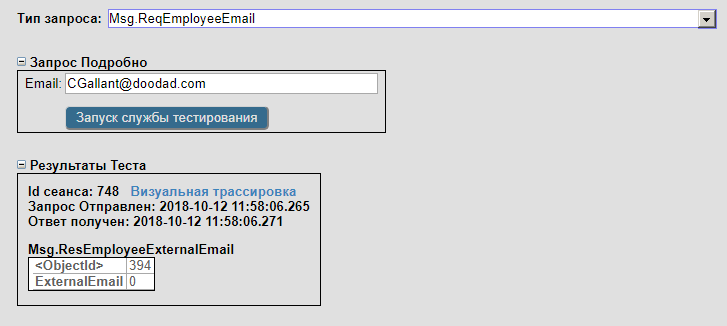


Рисунок 1 – Тестирование операции проверки на внешний адрес эл. Почты

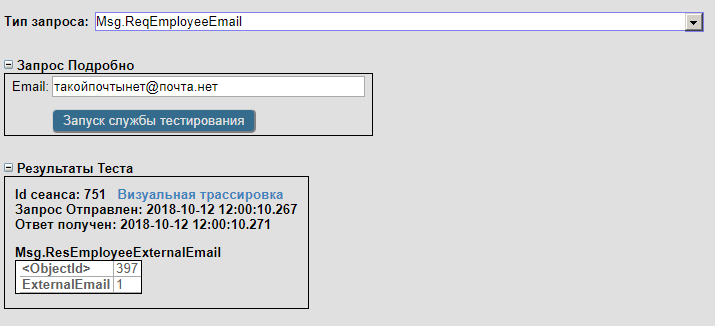


Рисунок 2 – Тестирование операции проверки на внешний адрес эл. Почты

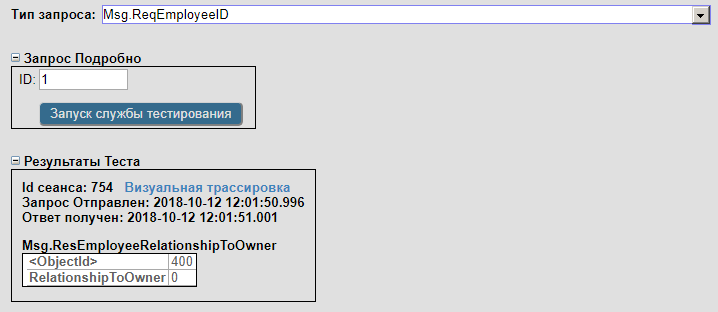


Рисунок 3- Тестирование операции проверки родственности

# Выводы

В ходе выполнения практической работы были получены навыки по созданию и управлению бизнес-операциями в Ensemble, а именно были реализованы файловый и SQL адаптеры, описаны сообщения типа запрос-ответ, а также создана и настроена продукция, которая содержит в себе данные операции. Поставленная задача была выполнена в полном размере.