Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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](#_Toc531791064)

[2 Листинг 4](#_Toc531791065)

[3 Примеры работы программы 9](#_Toc531791066)

[4 Выводы 10](#_Toc531791067)

# Постановка задачи

1. Необходимо разработать одну службу, которая отслеживает изменения в файле BlackList.txt. По добавлению новой записи, необходимо считать последнюю запись о сотруднике (имя сотрудника; company ID сотрудника; email адрес; текст письма).
2. Полученную информацию необходимо отправить в бизнес процесс (WFDelete). Данный бизнес процесс должен согласовать с ген.директором и директором об удалении сотрудника из компании. Если кто-нибудь из них не согласился, то бизнес-процесс завершается. Если оба соглашаются, то необходимо передать в бизнес-операцию (DeleteEmployee) CompanyID сотрудника.
3. Бизнес-операция (DeleteEmployee) по полученному идентификатору сотрудника должна проставить статус в базу данных о том, что сотрудник удален (саму запись удалять НЕ НУЖНО!)

# Листинг

Бизнес-сервис TrackingFileChangesInBlackList

Class BService.TrackingFileChangesInBlackList Extends Ens.BusinessService  
{  
  
Parameter ADAPTER = "EnsLib.File.InboundAdapter";  
  
Property Adapter As EnsLib.File.InboundAdapter;  
  
Method OnProcessInput(pInput As %Stream.Object, Output pOutput As %RegisteredObject) As %Status  
{  
 #dim e As %Exception.StatusException  
 set ..Adapter.DeleteFromServer = $$$NO  
 $$$TRACE("Entering into BlackList.txt ...")  
 set reader1 = pInput.ReadLine()  
 if (pInput.AtEnd)  
 {  
 $$$TRACE("No letters in file")  
 return $$$OK  
 }  
 set reader2 = pInput.ReadLine()  
 if (pInput.AtEnd)  
 {  
 $$$TRACE("Only one letter in file")  
 }  
 else  
 {  
 while ('pInput.AtEnd)  
 {  
 set reader1 = reader2  
 set reader2 = pInput.ReadLine()   
 }  
 $$$TRACE("More then one letter in file")  
 }  
 set req1 = ##class(Msg.ReqLetter).%New()  
 set req1.EmployeeName = $piece(reader1, ";", 1)  
 set req1.EmployeeID = $piece(reader1, ";", 2)  
 set req1.EmployeeEmail = $piece(reader1, ";", 3)  
 set req1.Text = $piece(reader1, ";", 4)  
 $$$TRACE("Read letter in request object: " \_ req1.EmployeeName \_ "|" \_ req1.EmployeeID \_ "|" \_ req1.EmployeeEmail \_ "|" \_ req1.Text)  
 try  
 {  
 $$$ThrowOnError(..SendRequestAsync("BProcess.WFDelete", req1, 1))  
 }  
 catch e  
 {  
 return e.AsStatus()  
 }  
 return $$$OK  
}  
  
}

Бизнес-процесс WFDelete

Class BProcess.WFDelete Extends Ens.BusinessProcess [ ClassType = persistent ]  
{  
  
Property AnswerCEO As %Integer [ InitialExpression = -1 ];  
  
Property AnswerCO As %Integer [ InitialExpression = -1 ];  
  
Method OnRequest(req As Msg.ReqLetter, Output res As Msg.ResSuccess) As %Status  
{  
 #dim e As %Exception.StatusException  
 set res = ##class(Msg.ResSuccess).%New()  
 try   
 {  
 set req1 = ##class(EnsLib.Workflow.TaskRequest).%New()  
 set req1.%Actions = "Yes,No"  
 set req1.%Subject = "Suspicious letter"  
 set req1.%Message = "Employee" \_ req.EmployeeName \_ "sent letter with next content: '" \_  
 req.Text \_ "' on Email: " \_ req.EmployeeEmail \_ ". Delete employee?"  
 set req2 = req1  
 $$$TRACE("Send workflow task for CEO and CO")  
 $$$ThrowOnError(..SendRequestAsync("CEO", req1, 1, "AnswerCEO"))  
 $$$ThrowOnError(..SendRequestAsync("CO", req2, 1, "AnswerCO"))  
 }  
 catch e  
 {  
 return e.AsStatus()  
 }  
 return $$$OK  
}  
  
Method OnResponse(req As Msg.ReqLetter, ByRef res As Msg.ResSuccess, callreq As Ens.Request, callres As Ens.Response, key As %String) As %Status  
{  
 if key = "AnswerCEO"  
 {  
 set res1 = ##class(EnsLib.Workflow.TaskResponse).%New()  
 set res1 = callres  
 if (res1.%Action = "Yes")  
 {  
 set ..AnswerCEO = 1  
 $$$TRACE("Deleting confirmed by CEO")  
 }  
 elseif (res1.%Action = "No")  
 {  
 set ..AnswerCEO = 0  
 $$$TRACE("Deleting rejected by CEO")  
 }  
 }  
 elseif key = "AnswerCO"  
 {  
 set res2 = ##class(EnsLib.Workflow.TaskResponse).%New()  
 set res2 = callres  
 if (res2.%Action = "Yes")  
 {  
 set ..AnswerCO = 1  
 $$$TRACE("Deleting confirmed by CO")  
 }  
 elseif (res2.%Action = "No")  
 {  
 set ..AnswerCO = 0  
 $$$TRACE("Deleting rejected by CO")  
 }  
 }  
 elseif key = "ChangeRecordOnDeletedById"  
 {  
 set res3 = ##class(Msg.ResSuccess).%New()  
 set res3 = callres  
 set res.Success = res3.Success  
 }  
 if (..AnswerCEO = 1) && (..AnswerCO = 1)  
 {  
 #dim e As %Exception.StatusException  
 set ..AnswerCEO = -1  
 set ..AnswerCO = -1  
 set req3 = ##class(Msg.ReqEmployeeID).%New()  
 set req3.ID = req.EmployeeID  
 try  
 {  
 $$$TRACE("Start operation 'DeleteEmployee'")  
 $$$ThrowOnError(..SendRequestAsync("BOperation.DeleteEmployee", req3, 1, "ChangeRecordOnDeletedById"))  
 }  
 catch e  
 {  
 return e.AsStatus()  
 }  
 }  
 elseif (..AnswerCEO = 0) ! (..AnswerCO = 0)  
 {  
 set ..AnswerCEO = -1  
 set ..AnswerCO = -1  
 $$$TRACE("Deleting employee " \_ req.EmployeeName \_ " was rejected by CEO or CO")  
 set res.Success = 1  
 }  
 return $$$OK  
}  
  
Storage Default  
{  
<Data name="WFDeleteDefaultData">  
<Subscript>"WFDelete"</Subscript>  
<Value name="1">  
<Value>AnswerCEO</Value>  
</Value>  
<Value name="2">  
<Value>AnswerCO</Value>  
</Value>  
</Data>  
<DefaultData>WFDeleteDefaultData</DefaultData>  
<Type>%Library.CacheStorage</Type>  
}  
  
}

Бизнес-операция DeleteEmployee

Class BOperation.DeleteEmployee Extends Ens.BusinessOperation  
{  
  
Parameter INVOCATION = "Queue";  
  
Parameter ADAPTER = "EnsLib.SQL.OutboundAdapter";  
  
Property Adapter As EnsLib.SQL.OutboundAdapter;  
  
XData MessageMap  
{  
<MapItems>  
  <MapItem MessageType="Msg.ReqEmployeeID">  
    <Method>ChangeRecordOnDeletedById</Method>  
  </MapItem>  
</MapItems>  
}  
  
Method ChangeRecordOnDeletedById(req As Msg.ReqEmployeeID, Output res As Msg.ResSuccess) As %Status  
{  
 set res = ##class(Msg.ResSuccess).%New()  
 #dim result As EnsLib.SQL.GatewayResultSet  
 set st = ..Adapter.ExecuteUpdate(,"update SQLUser.Employee set Deleted=1 where id = ?", req.ID)  
 $$$TRACE("Employee with id = " \_ req.ID \_ " was deleted")  
 if $$$ISERR(st)  
 {  
 set ^err = $system.Status.GetErrorText(st)  
 set res.Success = 0  
 }  
 else  
 {  
 set res.Success = 1  
 }  
 return $$$OK  
}  
  
}

# Примеры работы программы



Рисунок 1 – Трассировка событий



Рисунок 2 – Пример задачи workflow у CEO

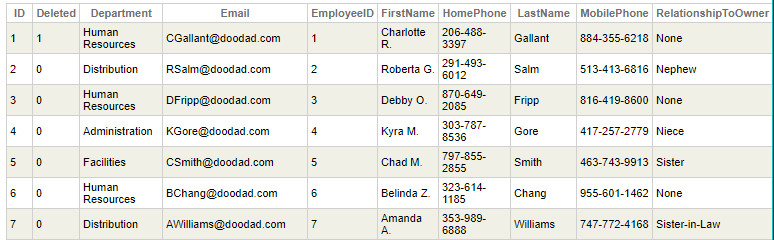


Рисунок 3 – Пример выполнения операции удаления сотрудника в БД

# Выводы

В ходе выполнения практической работы были получены навыки по созданию и управлению задачами workflow в Ensemble, а именно были созданы пользователи, созданы роли workflow и назначены пользователи на них, реализован вызов workflow-операции с формированным для него request и response. Также дополнительно подключен UI-проект для просмотра задач workflow и управления ими. Поставленная задача была выполнена в полном размере.