

# Deleting Data from a REST API (DELETE method)

## Table of Contents

<b>Introduction.....</b>	<b>2</b>
Connect to an Environment	2
<b>Get to know the scenario.....</b>	<b>5</b>
Install the Contacts application	5
Business Case Overview	5
<b>Consume a DELETE REST API Method.....</b>	<b>7</b>
<b>Add the Delete option to the User Interface.....</b>	<b>10</b>
<b>End Lab.....</b>	<b>17</b>

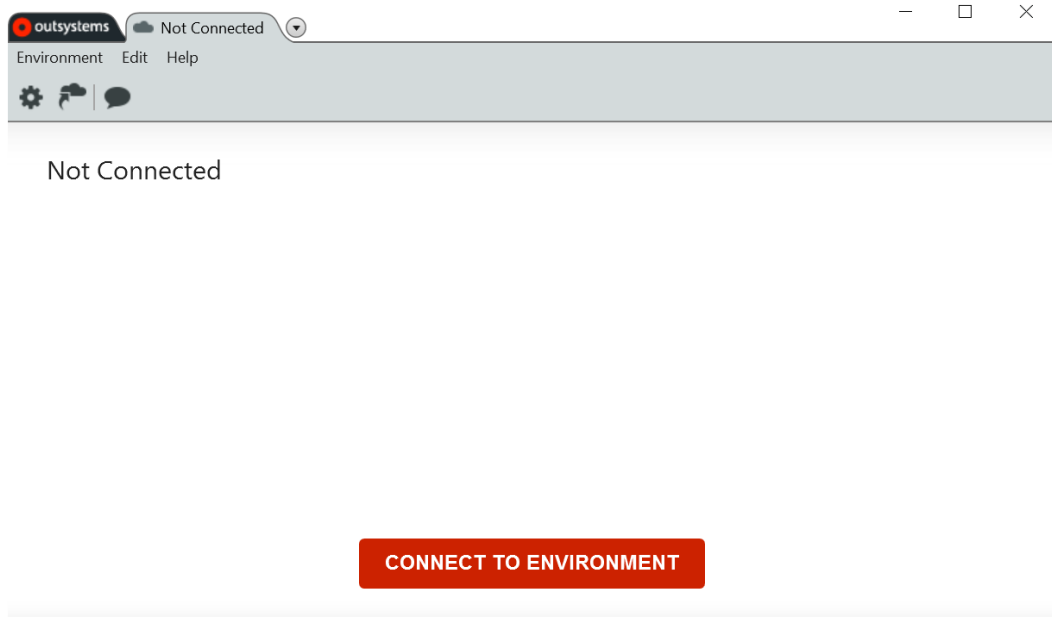
# Introduction

In this lab, we are going to integrate with a REST Web Service to allow to delete data in an external system. We are going to use the DELETE method to delete information of the Contacts in the external system.

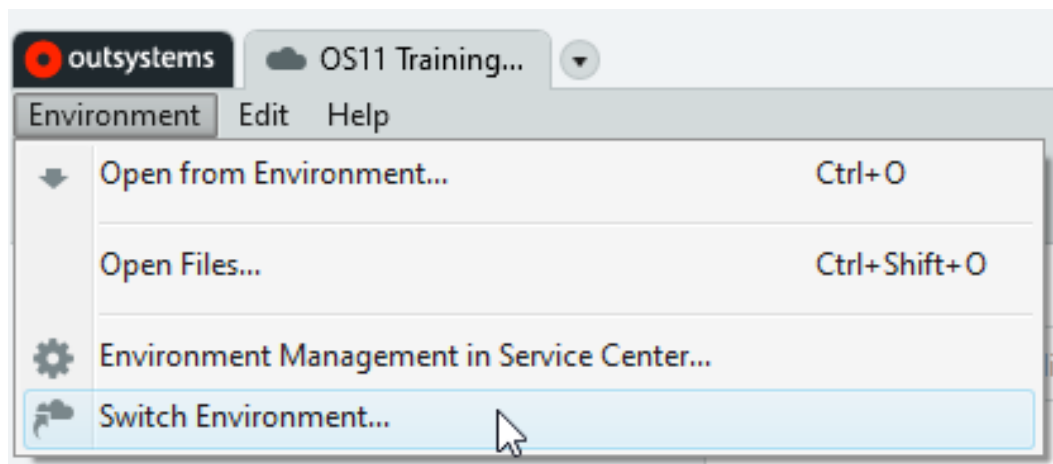
## Connect to an Environment

When we open Service Studio for the first time, we will need to connect to an **environment** where the OutSystems platform server generates, optimizes, compiles, and deploys OutSystems applications.

- 1) Open Service Studio and access the **Connect to Environment** dialog. This can be done in two ways.
  - a) If you are not logged in to any environment, click on **Connect to Environment**.

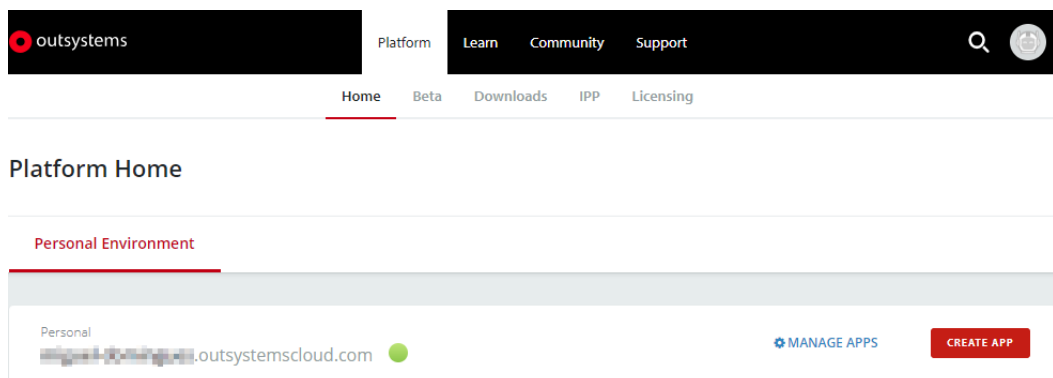


- b) If you are already logged in to an environment, select the **Switch Environment...** option from the Environment menu at the top.

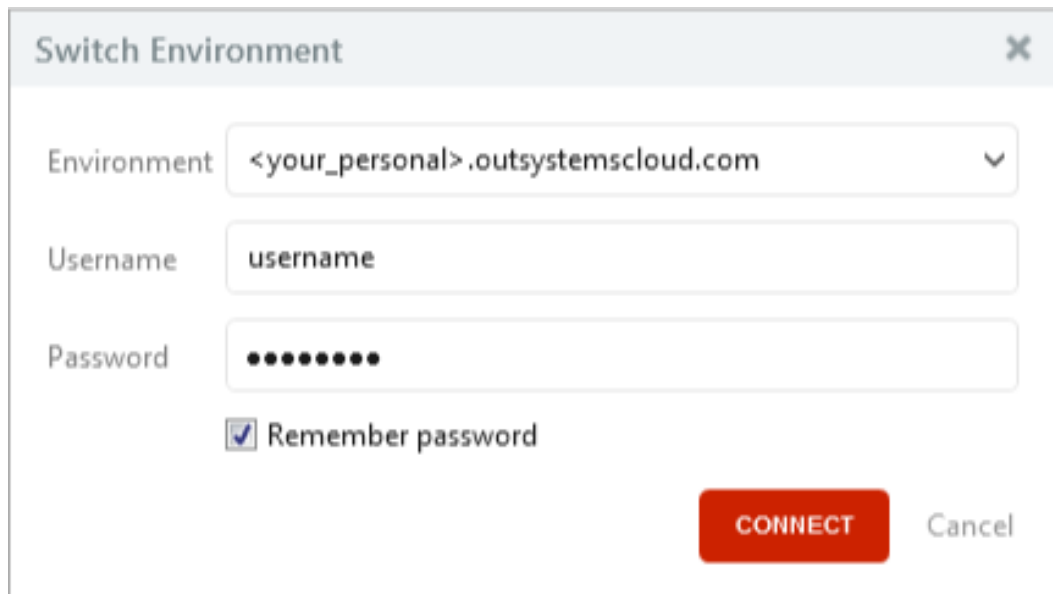


- 2) Connect to your OutSystems personal environment.

- a) If you are using your Personal Environment, you can find its address in the OutSystems [website](#) and log in.
- b) Under the **Platform** tab and then under the **Personal Environment** tab the environment address (or **Server Address**) can be found.



- c) Back in Service Studio, use that Environment and login with your OutSystems community email (username) and password.



The image shows a 'Switch Environment' dialog box with a close button (X) in the top right corner. It contains three input fields: 'Environment' with a dropdown menu showing '<your\_personal>.outsystemscloud.com', 'Username' with the text 'username', and 'Password' with masked characters '••••••••'. Below the password field is a checkbox labeled 'Remember password' which is checked. At the bottom right, there is a red 'CONNECT' button and a 'Cancel' button.

Switch Environment

Environment <your\_personal>.outsystemscloud.com

Username username

Password ••••••••

☒ Remember password

CONNECT Cancel

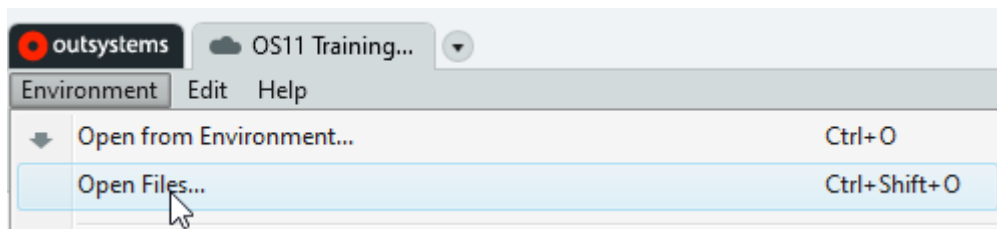
# Get to know the scenario

## Install the Contacts application

- 1) If you have completed the **Reading Data from a REST Web Service (GET)** exercise lab before you can skip this section and jump to the next page. Otherwise, follow the instructions below to install the quick start application.

Open and publish the **REST Contacts - DELETE.oap** in your personal environment. The oap file can be found in the Resources folder.

- 1) In the Applications, open the Environment menu and select Open Files...




- 2) In the Open dialog, change the File Type dropdown option to **OutSystems Application Pack (\*.oap)** and then open the REST Contacts - DELETE.oap.
- 3) Click Proceed when asked.
- 4) Wait for the installation to complete and then proceed.

## Business Case Overview

The REST Contacts application is a simple application, only containing a couple of Screens, Preparations, and some Screen Actions. We already list, create and update Employees on the screens. Now, we'll create the logic to delete a Contact.

The quick start application is simply to speed up the setup part and start working right away with the external REST web service.

 REST Contacts Login

## Contacts

[+ Add New Contact](#)

Name	Gender	Occupation	Company
Joaquin Viola	Male	Periodontist	Pay 'N Pak
Richard Natividad	Male	Set and exhibit designer	The Jolly Farmer
Christina Andresen	Female	Nurse	Whitlocks Auto Supply

# Consume a DELETE REST API Method

In OutSystems, integration with REST Services can be straightforward. OutSystems help us to generate all the methods and data structures needed to integrate with an external system.

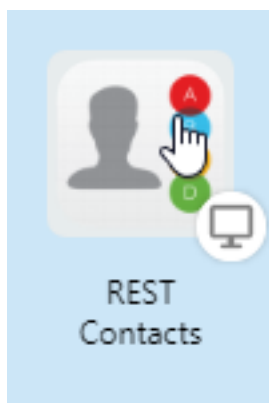
The DELETE request is known as a method to delete existing data from the external system.

Before you consume any REST API it's important to gather all the information you need from the REST API documentation. Information such as the expected structures and some examples maybe be really useful when consuming an external service. In this lab, the documentation for the external REST Web Service is available [here](#).

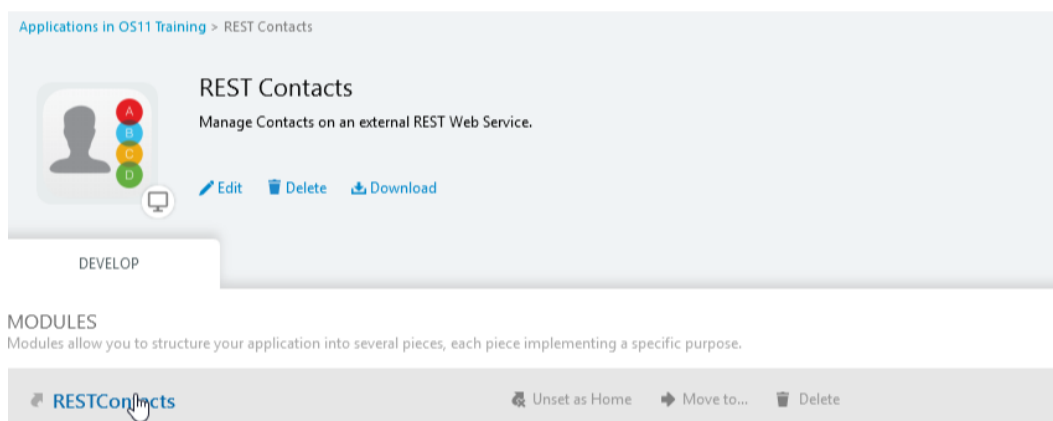
In this section we will create the integration with the external REST Web Service. This integration in specific will allow to delete contacts from the external system.

## 1) Open the Contacts module

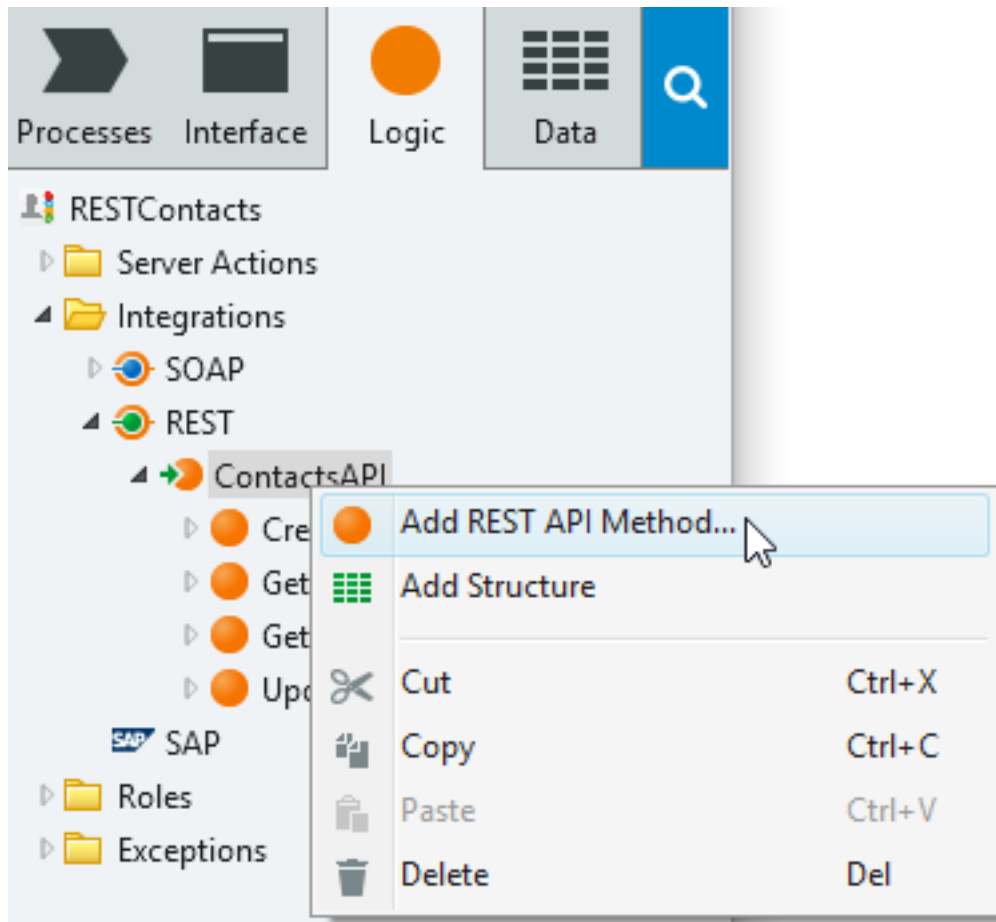
- a) In the Applications list, locate the **REST Contacts** Web application and open it.



## b) Open the **RESTContacts** module



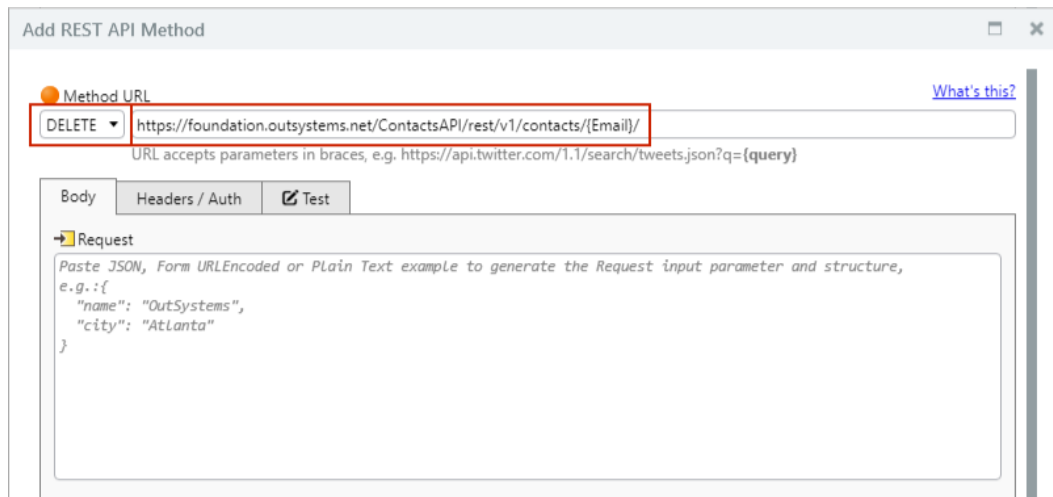
- 2) Consume the DELETE method of the external REST Web Service.
  - a) Switch to the Logic tab and in the Integrations folder, right-click the **ContactsAPI** REST element and select *Add REST API Method...*. This opens a window to specify the REST method that you want to consume.



- b) Set the Method to DELETE and the URL to

```
https://foundation.outsystems.net/ContactsAPI/rest/v1/contacts/{Email}/
```





- c) In the Body tab, leave the Request text area blank. In this method we are passing the information of the contact via the Email URL parameter.
- d) Set the Response text area to

```
{
  "Success": true,
  "ErrorMessage": "string"
}
```

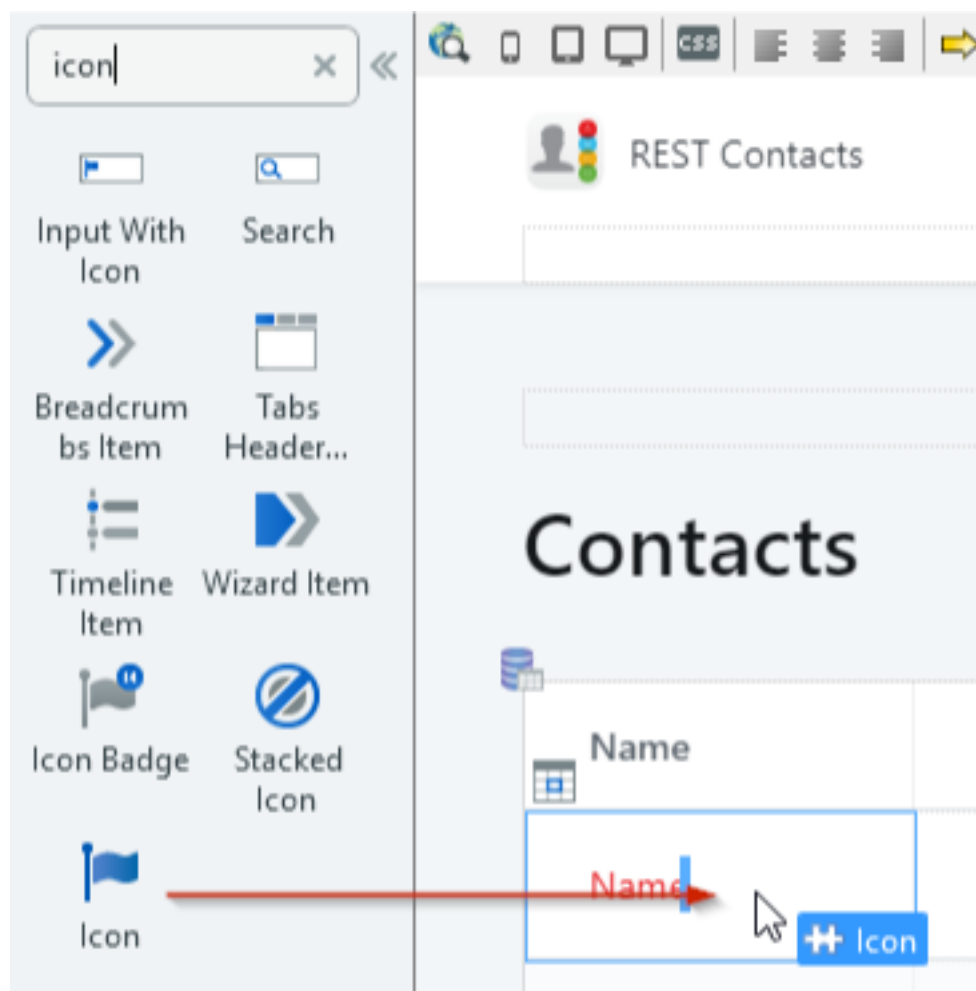
**NOTE:** This value can be obtained from the [Documentation](#) page of the Contacts API.

- e) Click the **OK** button to close the dialog window.

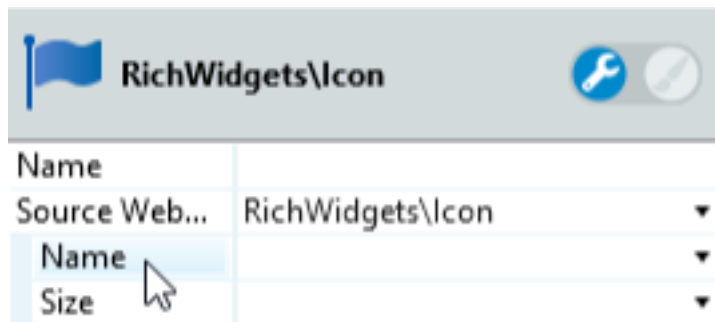
## Add the Delete option to the User Interface

In this section we will modify the existing Contacts screen to allow to end users to delete a contact. In our sample app, the Contacts screen already displays the list of contacts and allows to edit each one of them.

- 1) For each row on the existing Table Records, add a new icon and link it to the Delete Screen Action.
  - a) In the Interface tab, open the **Contacts** Web Screen.
  - b) On the Widgets toolbox, search for Icon, then Drag an Icon and drop it next to the Contact Name.

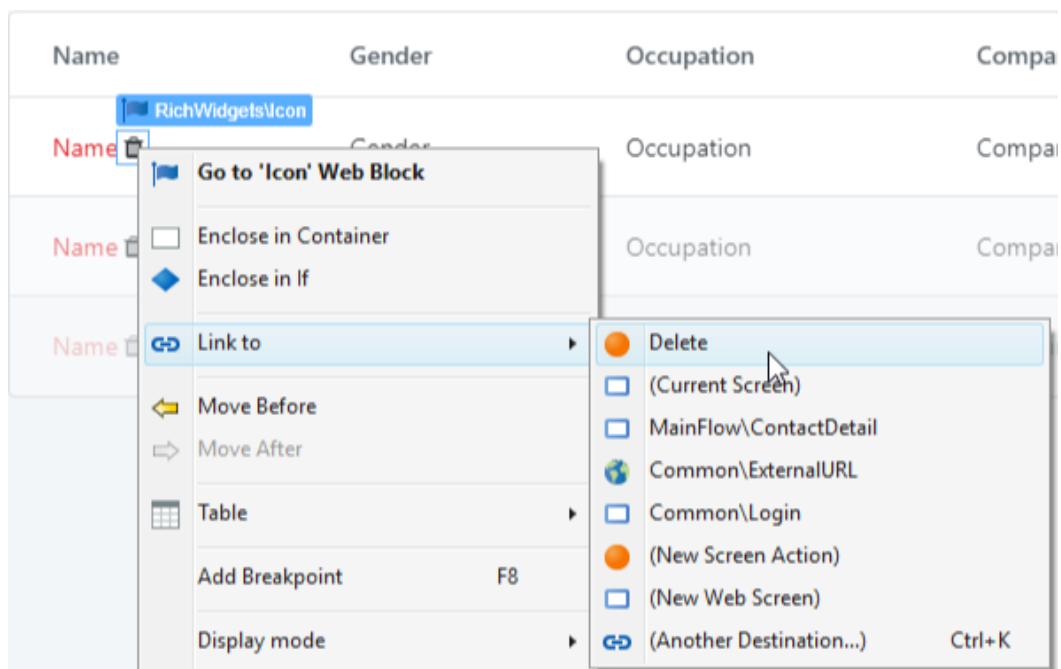


- c) Double-click the Name property of the Icon to open the Icon picker, then select the *trash* icon.



**NOTE:** Alternatively you may set also the Expression to `Entities.IconName.trash`

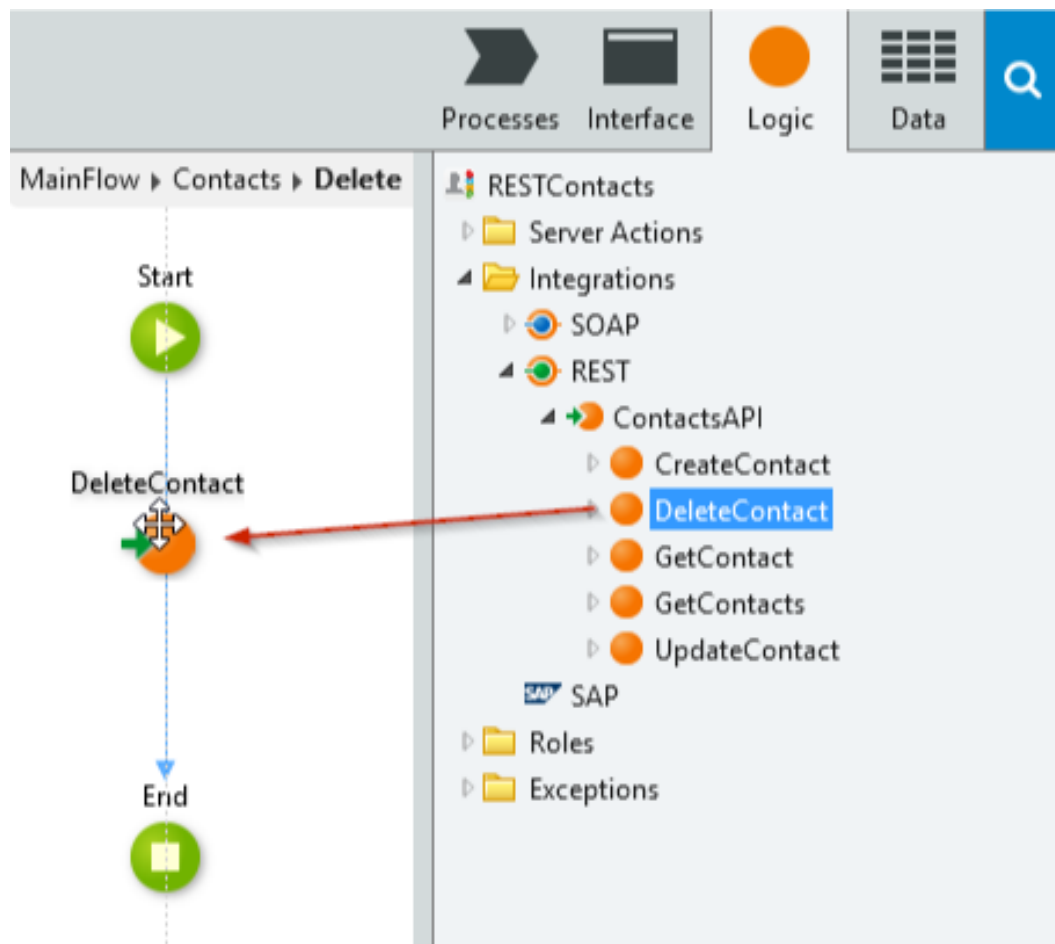
- d) Right-click the Icon on the screen, and select **Link to > Delete**



**NOTE:** Optionally you can set the Confirmation Message property of the Link to ask the end-user for confirmation before actually deleting. In similar operations that may not be rolled back, this is considered a best practice.

- 2) Create the Logic of the Delete screen action to delete the contact in the external system. In the case the operation fails, display a feedback message detailing the error.
  - a) Open the **Delete** Screen Action. It is located under the Contacts screen, or you can also double-click the Link.

- b) From the Logic tab, drag the *DeleteContact* API method created before, and drop just after the start.



- c) Set the Email parameter to

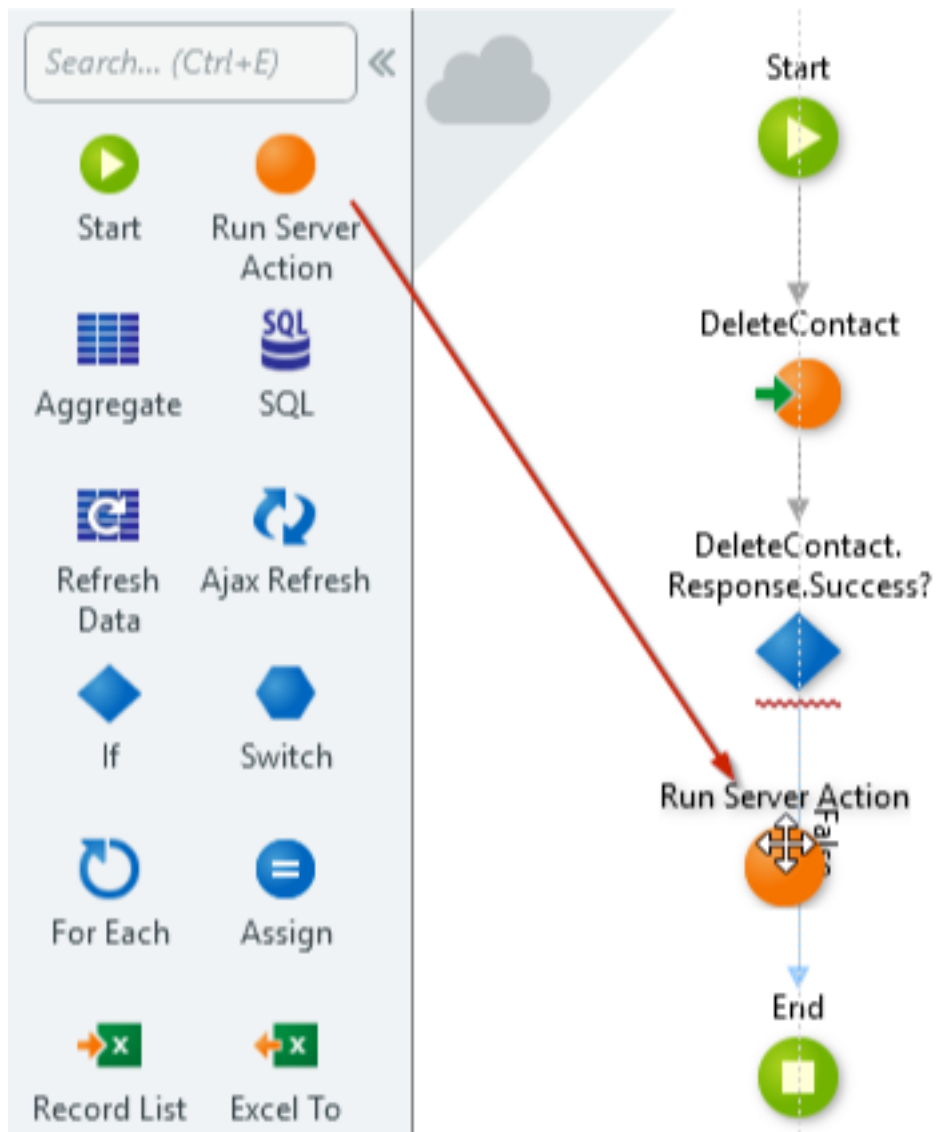
```
TableRecords1.List.Current.Email
```

DeleteContact Run Server Action	
Name	DeleteContact
Action	DeleteContact ▼
Email	TableRecords1.List.Current.Email ▼

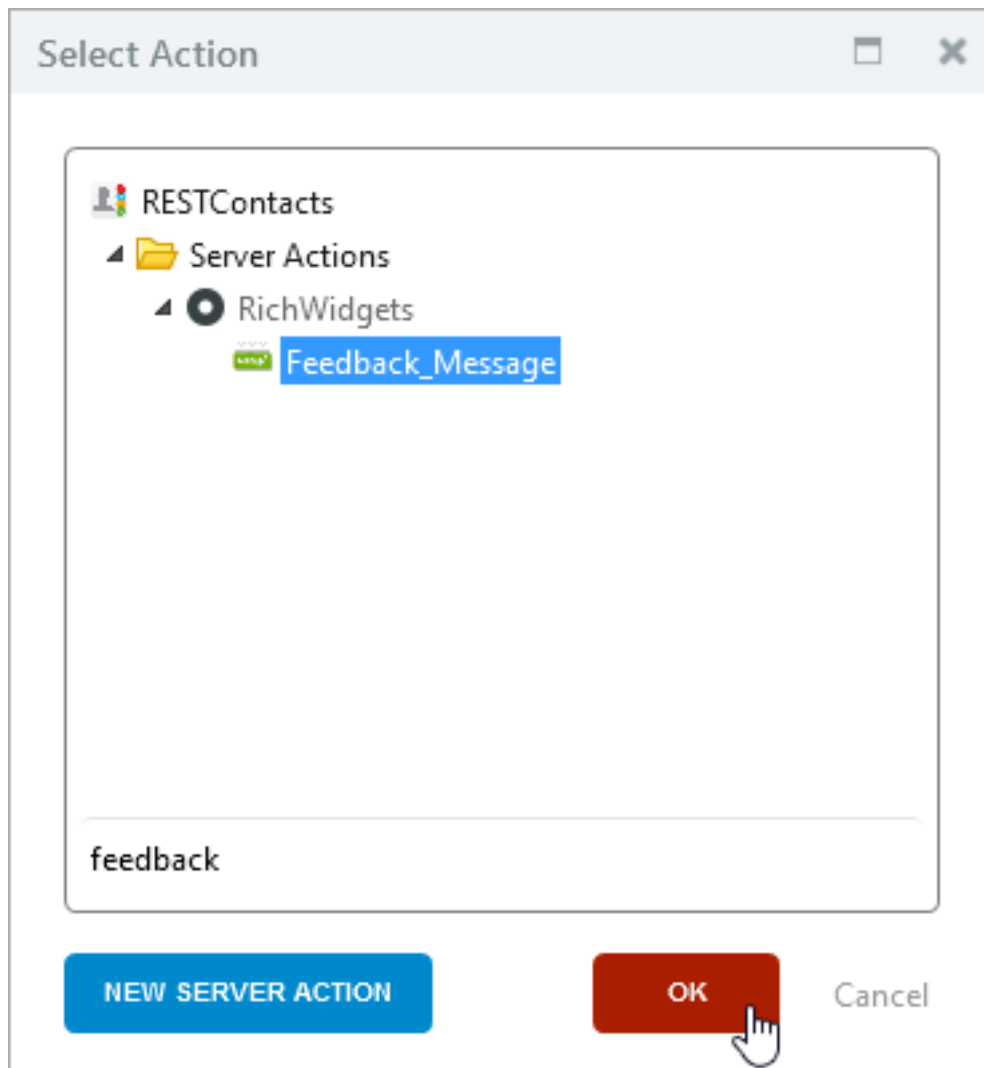
- d) Drag an **If** and drop it between the DeleteContact and the End.
- e) Set the **Condition** of the If to

```
DeleteContact.Response.Success
```

- f) Drag a Run Server Action and drop it on the False branch



- g) In the **Select Action** dialog choose the Feedback\_Message

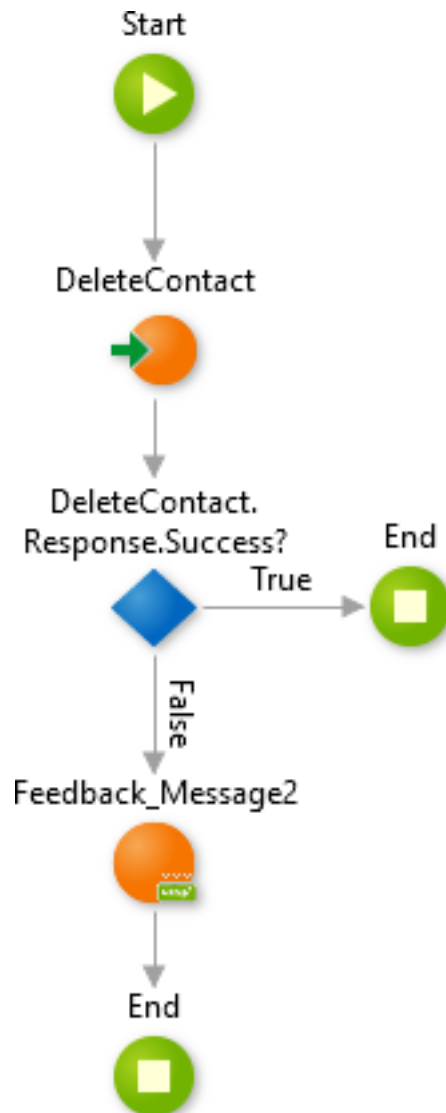


- h) Set the parameters of the Feedback\_Message2 element as follows

Feedback_Message2 Run Server Action	
Name	Feedback_Message2
Action	Feedback_Message ▼
MessageText	DeleteContact.Response.ErrorMessage ▼
MessageType	Entities.MessageType.Error ▼

- i) Drag an **End** and drop it to the right of the If, then connect both.

j) The Delete screen action flow should look like this



3) Publish the application using 1-Click Publish button and verify that the publish completed successfully in the 1-Click Publish Tab.

- a) Click on the **1-Click Publish** button to publish the module to the server.
- b) Verify in the 1-Click Publish tab that the publishing process was successful.

✓ TrueChange™	Debugger	✓ 1-Click Publish
1 Uploading	Storing a new version into 'https://os11training.outsystems.net/ServiceCenter'.	
2 Compiling	Generating and compiling optimized ASP.NET C# code and creating SQL scripts.	
3 Deploying	Updating SQL Server database model and deploying the web application to IIS.	
✓ Done	'RESTContacts' is now available at 'https://os11training.outsystems.net/RESTContacts'.	

- c) Preview the app in browser by clicking on the **Open in Browser** button.

- d) Click on the Trash icon of one of the contacts.
- e) If the operation was successful, the contact should not appear on the list, otherwise, a feedback message should highlight the error.



## End Lab

In this lab, we integrated with a REST Web Service, namely the DELETE method to delete Contacts on the external system.

To accomplish that, we have used the OutSystems visual interface to add an extra method that enables the deletion of a contact. Once the new method was added to the integration, the user interface and underlying logic was changed to allow to delete contacts.

At the end of this exercise you should be able to integrate with a simple REST API to delete data in an OutSystems application.