

Deleting Data from a SOAP Web Service

Table of Contents

Introduction.....	2
Connect to an Environment	2
Get to know the scenario.....	5
Install the Contacts application	5
Business Case Overview	5
Consume a SOAP Web Service Method to Delete a Record.....	7
Add the Delete option to the User Interface.....	11
End of Lab.....	20

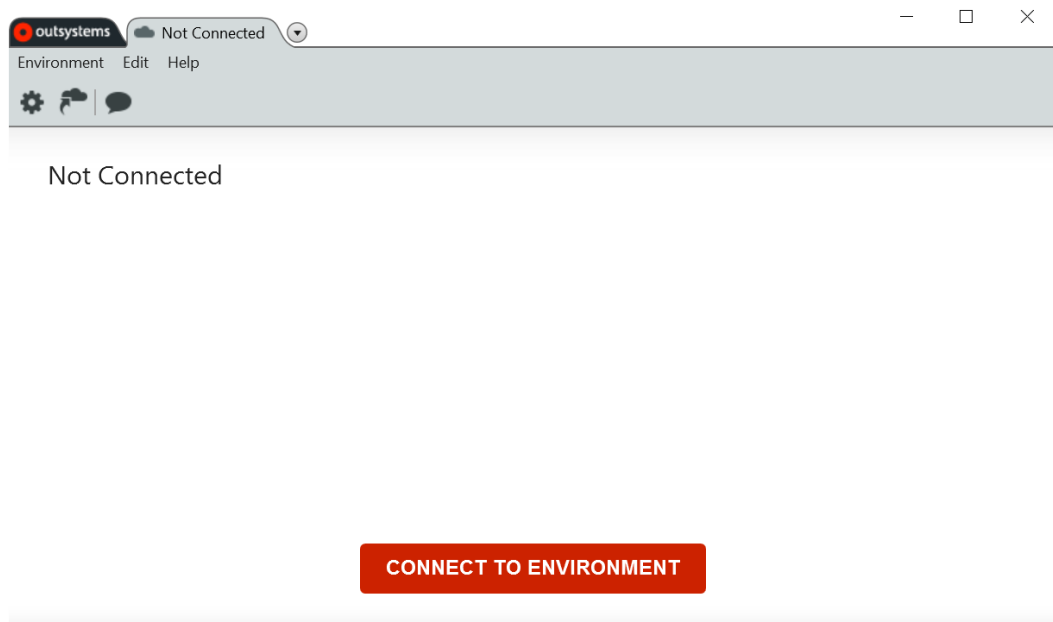
Introduction

In this lab, we are going to integrate with a SOAP Web Service to allow to delete data in an external system. We are going to use a SOAP method to delete a Contact in the external system.

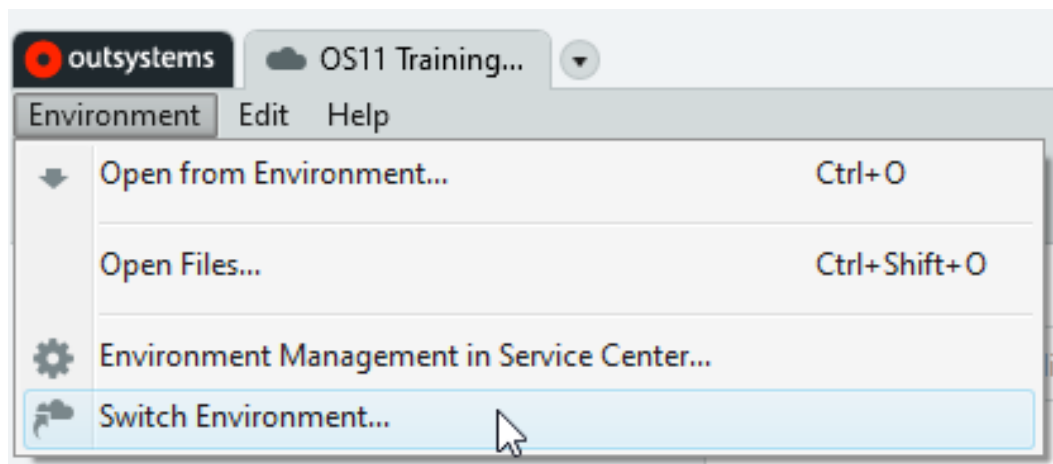
Connect to an Environment

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**.

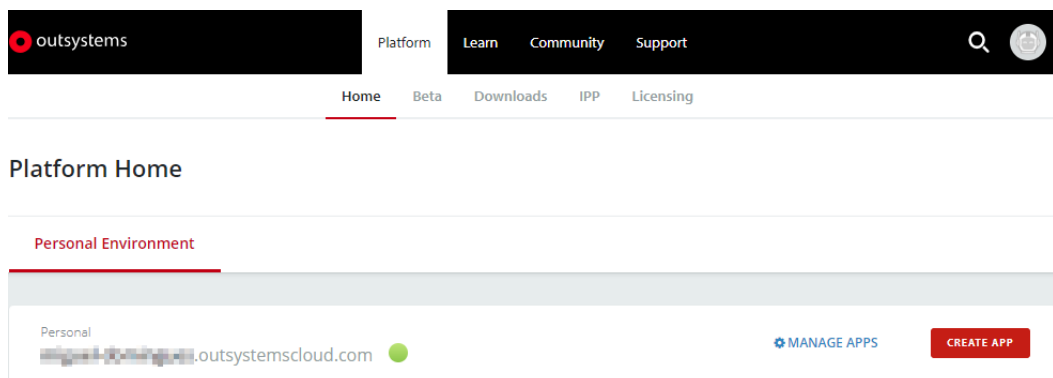


- b) If you are already logged in to an environment, select the **Switch Environment...** option from the Environment menu at the top.

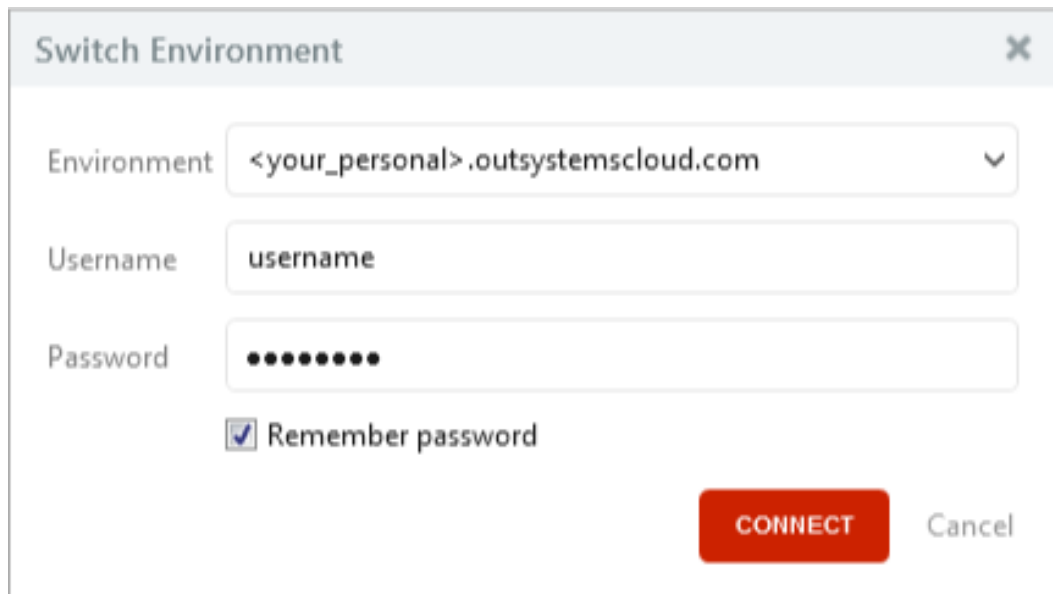


- 2) Connect to your OutSystems personal environment.

- 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

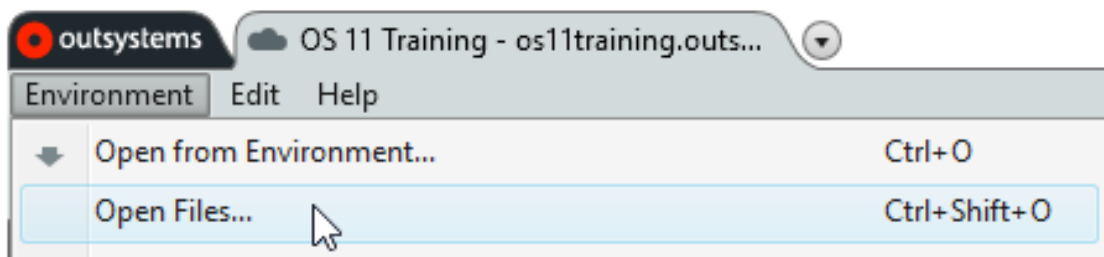
Get to know the scenario

Install the Contacts application

If you have completed the **Writing Data to a SOAP Web Service** 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 **SOAP Contacts - Deleting Data.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 SOAP Contacts - Using Data.oap.
- 3) Click Proceed when asked.
- 4) Wait for the installation to complete and then proceed.

Business Case Overview

The SOAP Contacts application is a simple application, only containing a couple empty Screens, Preparation and some Screen Actions, also empty. These elements are where we'll create the logic to get the Contacts and display the list on the screen. This quick

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

 SOAP Contacts Login

Contacts + Add New Contact			
Name	Gender	Occupation	Company
Christopher Day	Male	Office administrator	Dee's Drive-In
Robert Chadwick	Male	FBI agent	Four Leaf Clover
Shana Demps	Female	Social work policy maker	Present Company
Aron Elliott	Male	Structural and reinforcing iron and metal worker	Father & Son
Edward Noble	Male	Cardiovascular technologist	Clemens Markets
William McDonald	Male	Funeral attendant	MegaSolutions

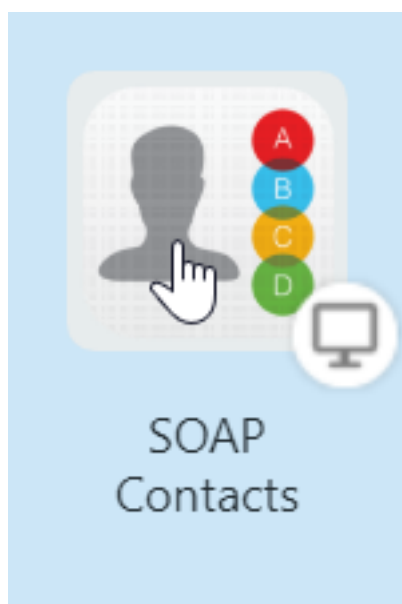
Consume a SOAP Web Service Method to Delete a Record

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

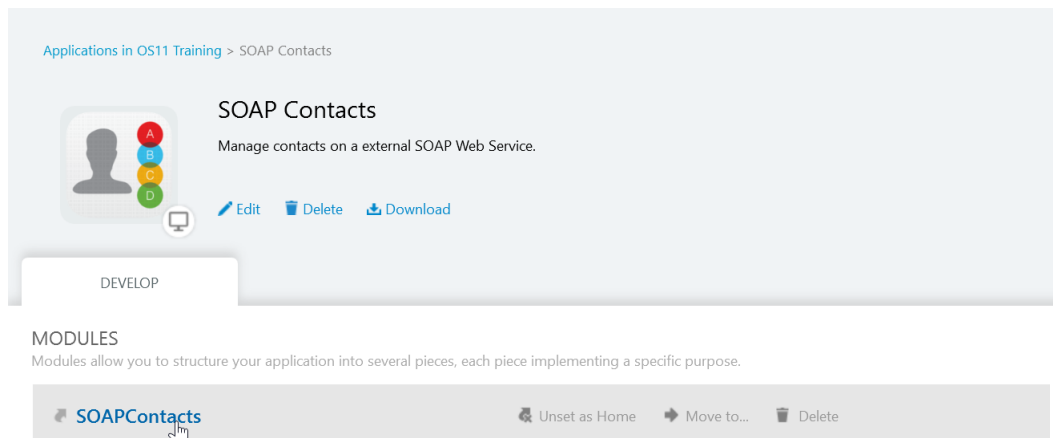
Before you consume any SOAP Web Service it's important to gather all the information you need from the SOAP Web Service documentation. Information such as the expected structures and some examples may be really useful when consuming an external service. In this lab, the documentation for the external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

In this section we will create the integration with the external SOAP 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 **SOAP Contacts** Web application and open it.



b) Open the **SOAPContacts** module.

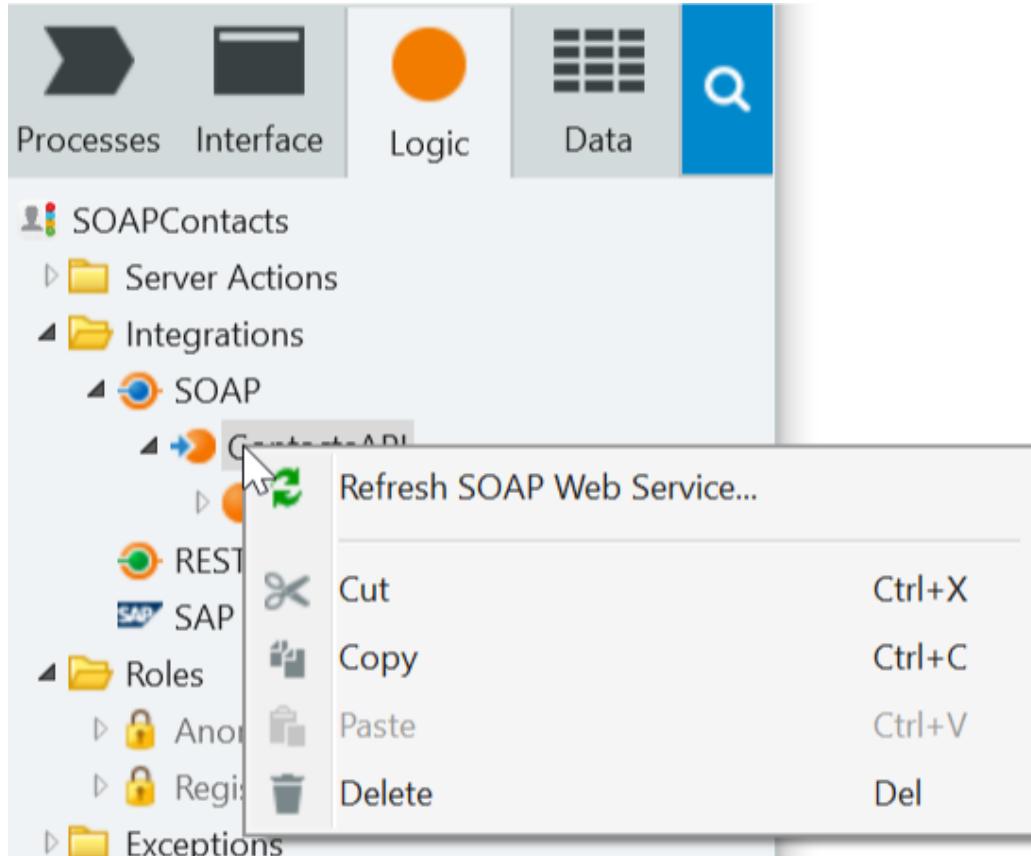


c) If a dialog appears asking to update references, click Yes to open the Manage Dependencies window. There, click on **Refresh All** and then OK.

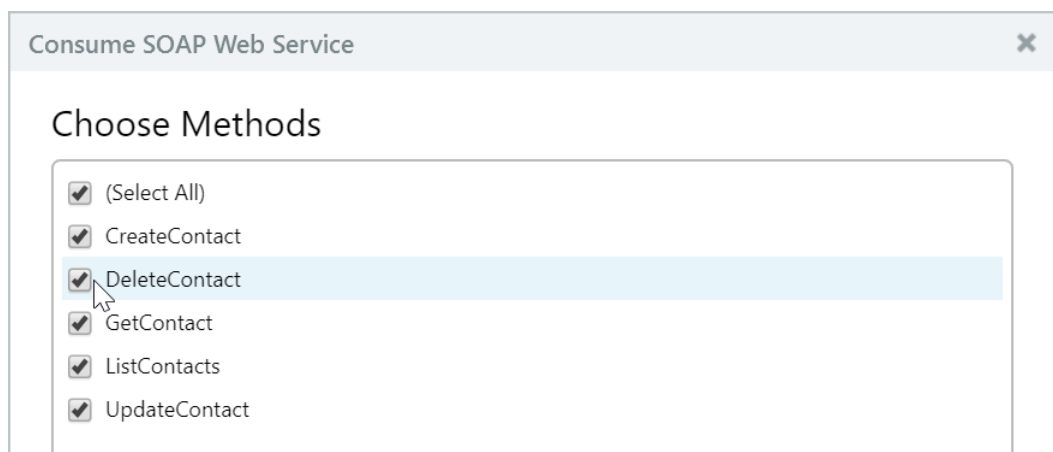


3) Consume the method to delete a Contact in the external SOAP Web Service.

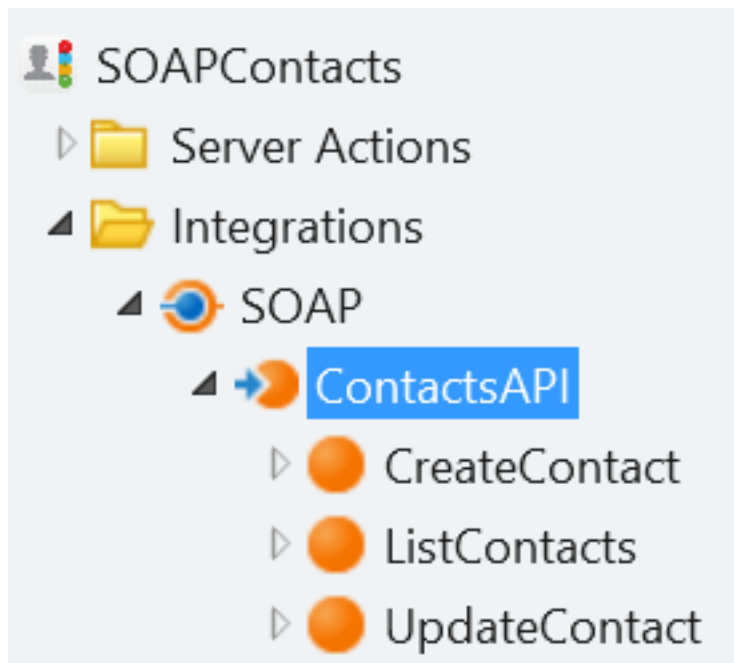
- a) Switch to the Logic tab and, in the Integrations folder, right-click the SOAP element and select **Refresh SOAP Web Service...**



- b) If a dialog appears asking whether to proceed, click **Yes**.
- c) In the next dialog, *ListContacts*, *CreateContact*, *GetContact*, *UpdateContact* should be already selected. Select **DeleteContact** and press *Finish*.



- d) Change the name of the SOAP Integration from *v1* to *ContactsAPI*.

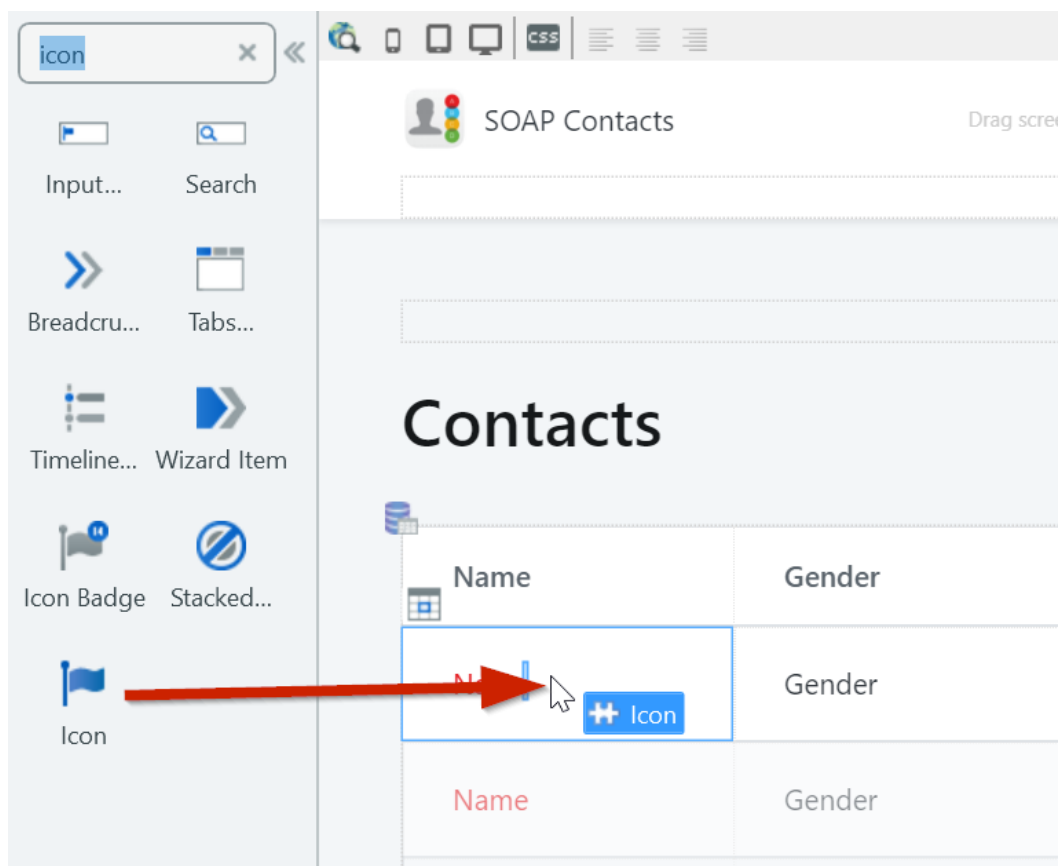


NOTE: This step is not mandatory, but it allows developers to have custom or more familiar names in their integrations.

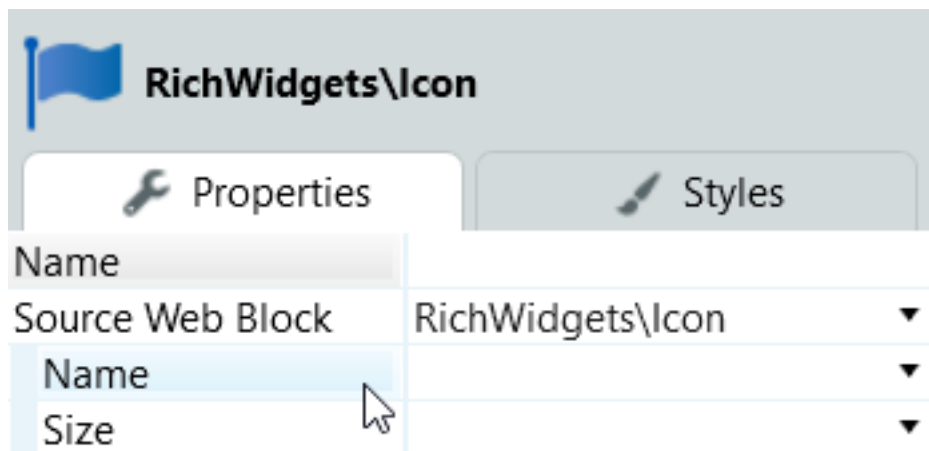
Add the Delete option to the User Interface

In this section we will modify the existing Contacts screen to allow 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, type icon in the SearchBox to filter the widgets. Drag the Icon widget and drop it next to the Contact Name.

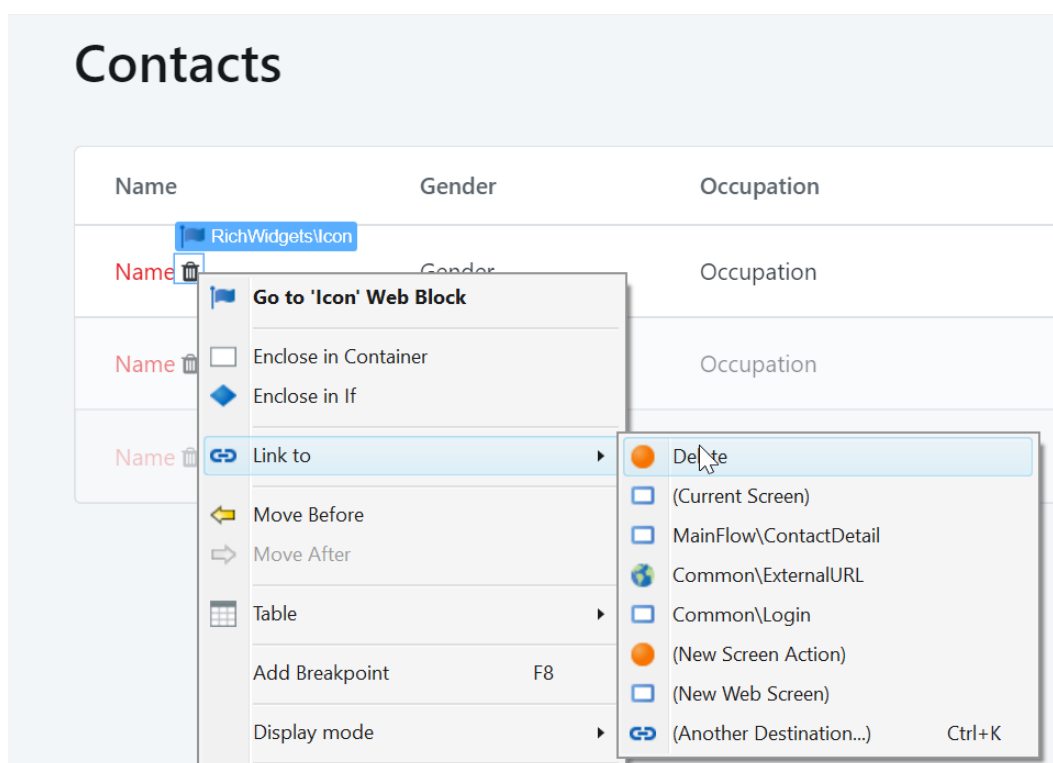


- c) Double-click the Name property of the Icon to open the Icon picker, then type trash, pick the **trash icon** and press OK.



NOTE: Alternatively you may directly set the Name property to:
`Entities.IconName.trash`

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

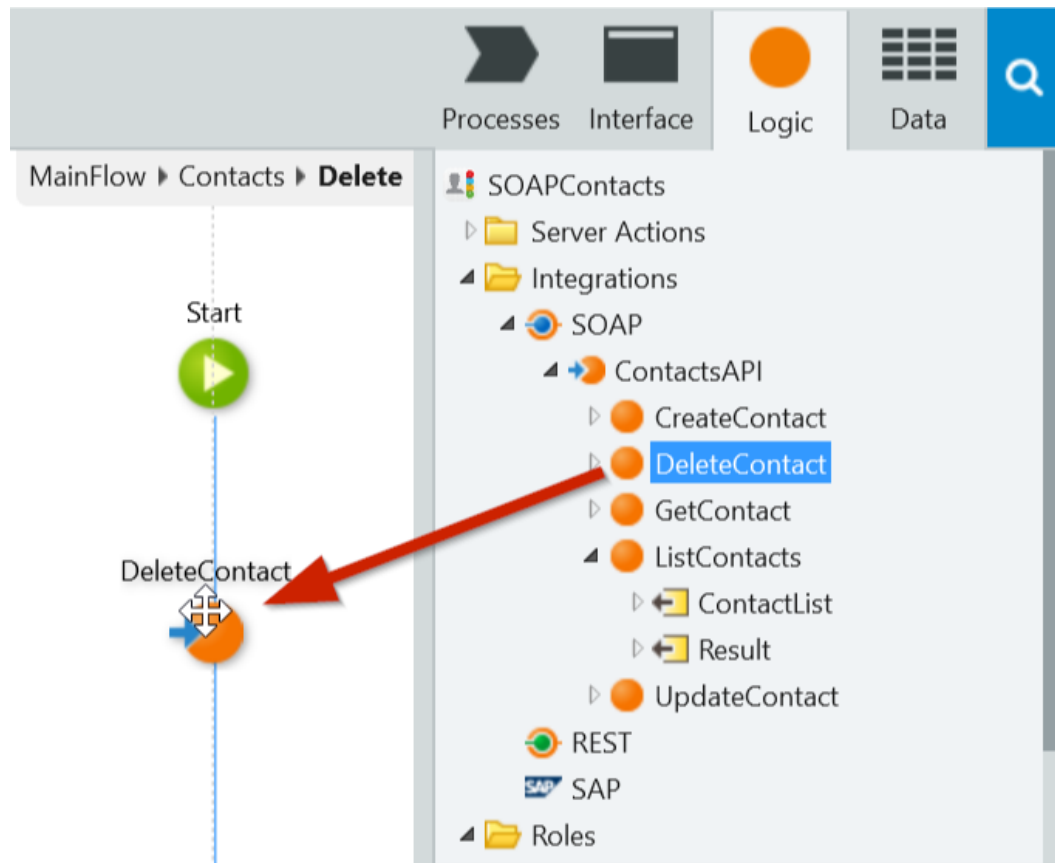


NOTE: Optionally you can set the Confirmation Message property of the Link to ask the end-user for confirmation before actually deleting. In similar operations that may not be rolled back, this is considered a best practice.

- 2) Create the Logic of the Delete screen action to delete a Contact in the external

system. In the case the operation fails, display a feedback message detailing the error.

- Open the **Delete** screen action located under the Contacts screen, or you can also double-click the Link.
- In the Logic tab, drag the *DeleteContact* SOAP method and drop just after the start.



- Set the Email parameter to:

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

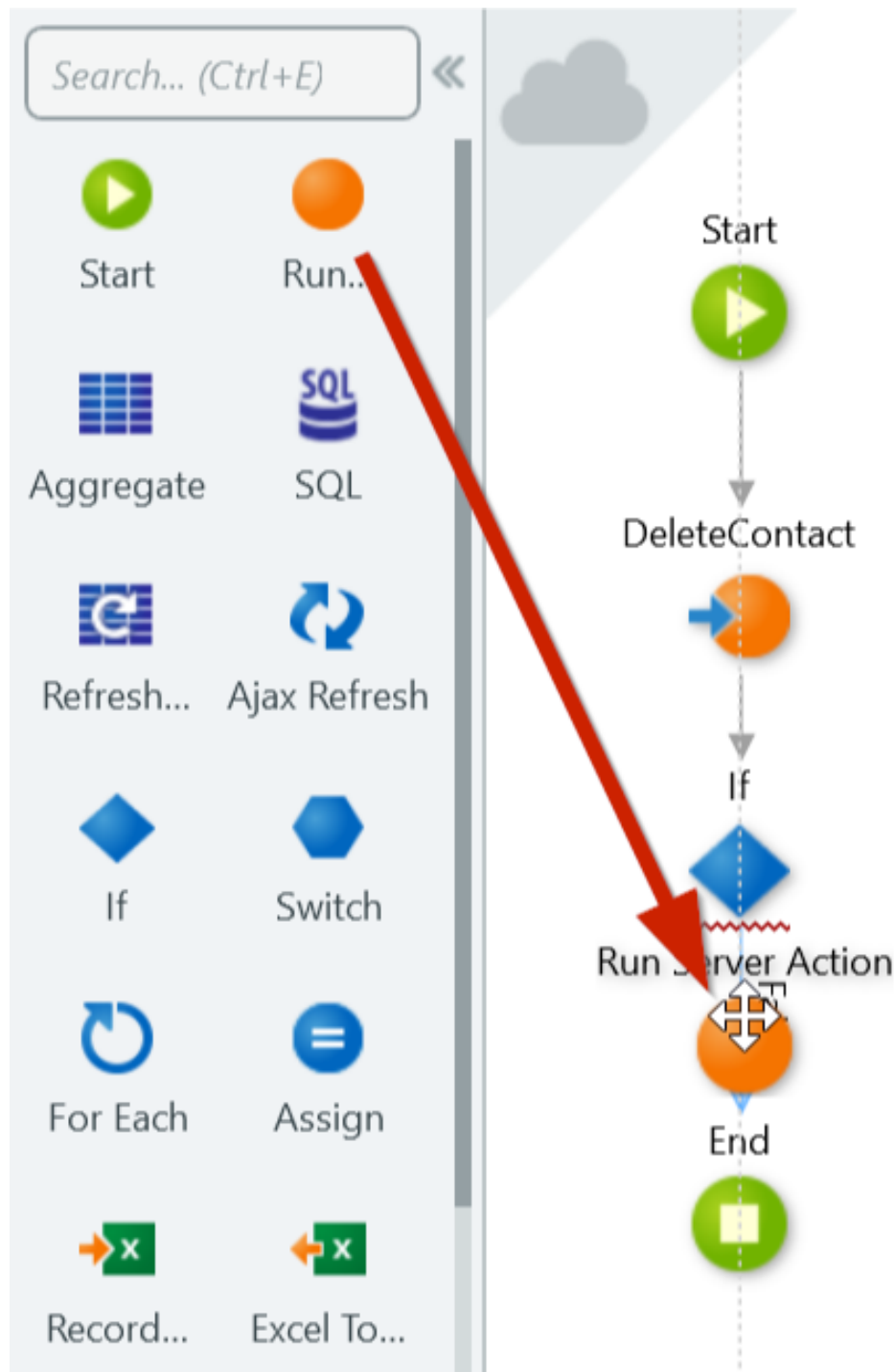
DeleteContact Run Server Action	
Name	DeleteContact
Action	DeleteContact ▼
Email	TableRecords1.List.Current.Email ▼

d) Drag an **If** and drop it between the DeleteContact and the End.

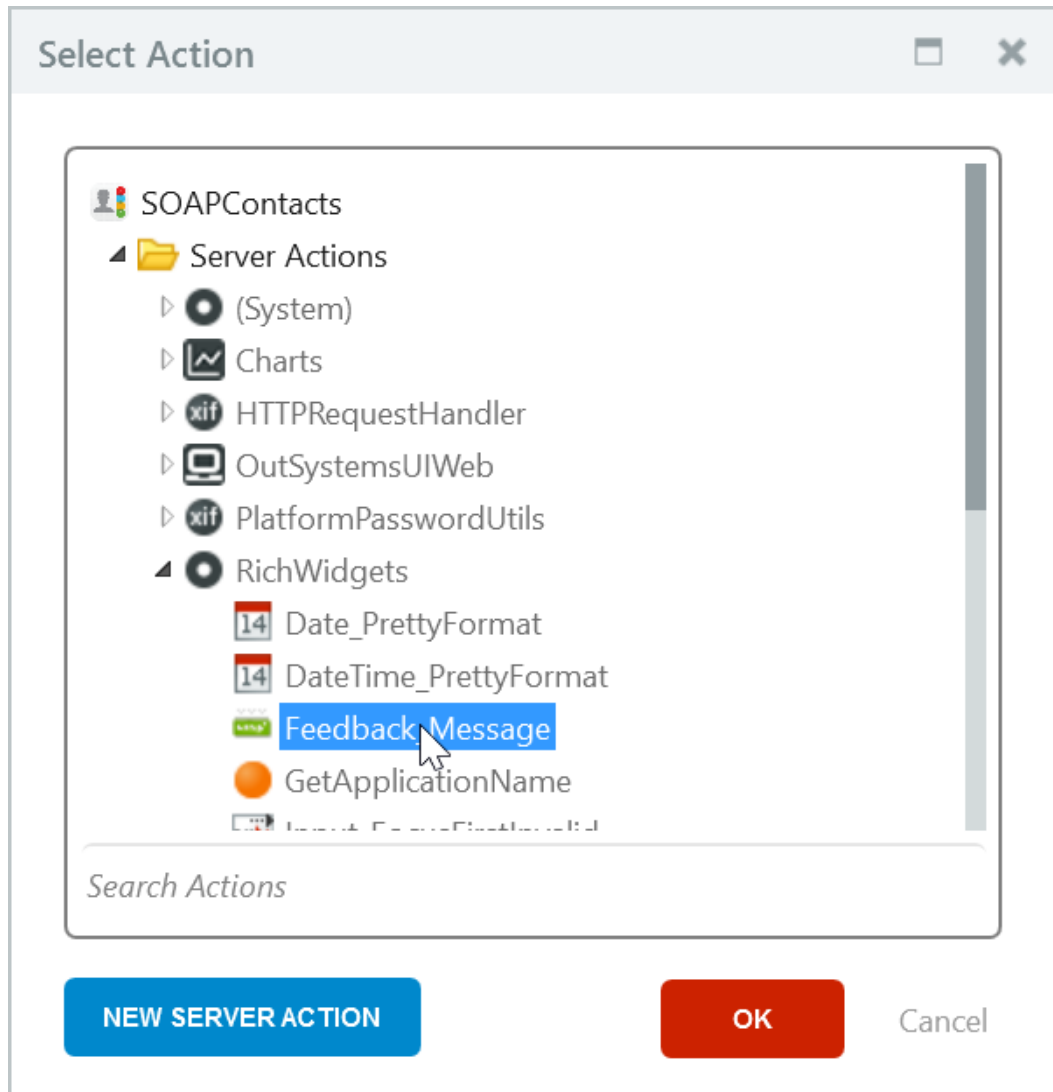
e) Set the **Condition** of the If to

```
DeleteContact.Result.Success
```

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




- g) In the **Select Action** dialog choose the *Feedback_Message* and click OK.



- h) Set the parameters of the Feedback_Message2 element with the following values:

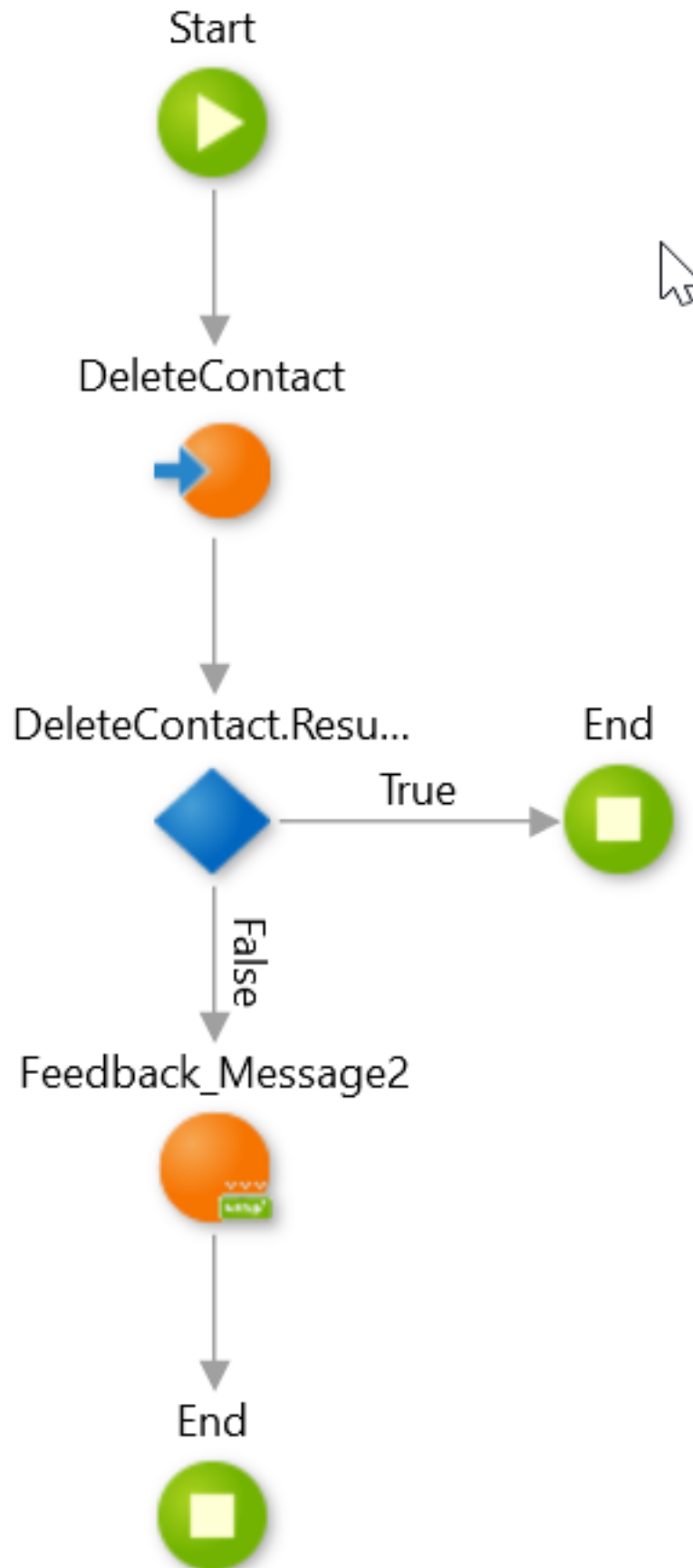
MessageText: DeleteContact.Result.ErrorMessage

MessageType: Entities.MessageType.Error

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

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

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.

1 Warning	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	'SOAPContacts' is now available at 'https://os11training.outsystems.net/SOAPContacts'.	

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 you have created**.

SOAP Contacts			Login
Contacts			+ Add New Contact
Name	Gender	Occupation	Company
Neo 	Male	OutSystems Mascot	OutSystems
Christopher Day 	Male	Office administrator	Dee's Drive-In
Robert Chadwick 	Male	FBI agent	Four Leaf Clover
Shana Demps 	Female	Social work policy maker	Present Company
Aron Elliott 	Male	Structural and reinforcing iron and metal worker	Father & Son

e) If the operation was successful, the contact should not appear on the list, otherwise, a feedback message should highlight the error.

End of Lab

In this lab, we integrated with a SOAP Web Service, namely a method to delete a Contact 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 deleting Contacts.

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