

Creating a Reactive Web App - How-to

Table of Contents

Outline	2
How to	2
Create the Application	2
Create a Client Action	6
Use the Client Action on a Screen	10

Outline

In this exercise, we will create our first reactive web application. This application will manage movies and people involved in those movies, such as cast and crew.

Over the next exercises, we will progressively build the application with new functionality, but for now we'll start by creating the application and two modules, one for the interface and the other for Logic/Entities.

- Reactive Web App Name: OSMDb_<your_initials>
- Interface module: OSMDb_<your_initials> - Reactive web
- Core module: OSMDb_Core_<your_initials> - Blank module

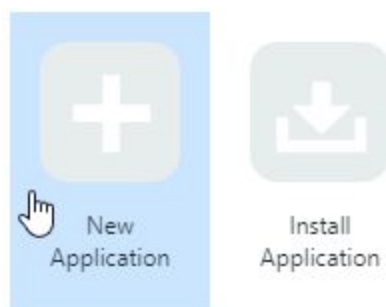
After the application is created, we want to create a simple action that sets a static text to a variable in the Blank module, and then display that message in a Screen in the Reactive Web module.

How to

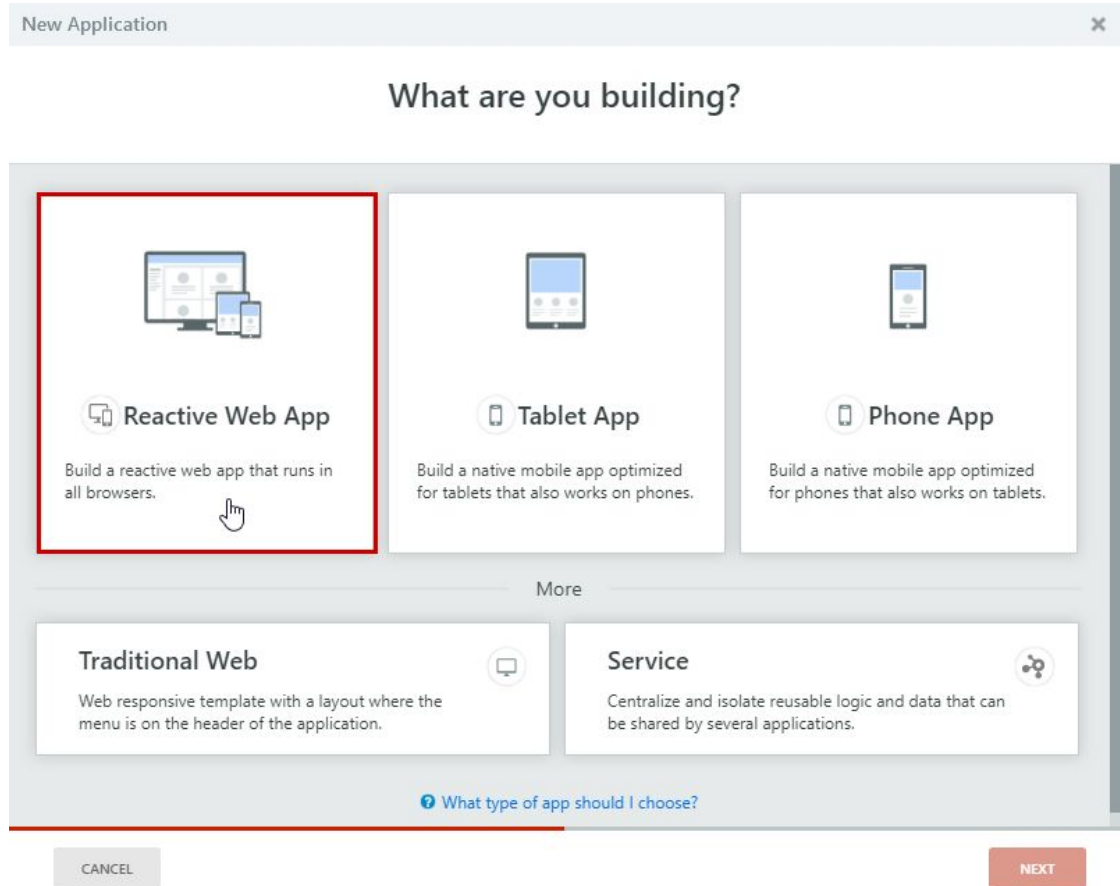
In this document, we will describe, step by step, the exercise 1.3. - *Creating a Reactive Web App*.

Create the Application

1. Create a Reactive Web application, called *OSMDb*.
 - a. In Service Studio, click on the **New Application** icon.




- b. Choose the **Reactive Web App** option.





















- c. Fill in the Application name and description. Since we're using a shared server, add your initials to the application name: *OSMdb_<Your Initials>*.

New Application ×

Fill in your app's basic info




Pick a color to bootstrap your app's interface and icon background

Or use a custom icon

UPLOAD ICON



BACKCREATE APP

- d. Click on **Upload Icon** and select the *OSMdb-icon.png* file to use a custom icon on the app.

Or use a custom icon



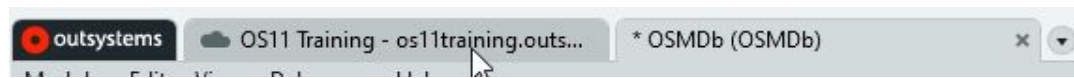
- e. Click on **Create App** to create the application.

2. Create two modules for the application: a **Reactive Web app** module called *OSMDb_<Your Initials>* and a **Blank** module called *OSMDb_Core_<Your Initials>*.
 - a. In the application Screen, there's already a suggestion of a module. Make sure it's called *OSMDb*, of type **Reactive Web App** and select **Create Module**.

Modules

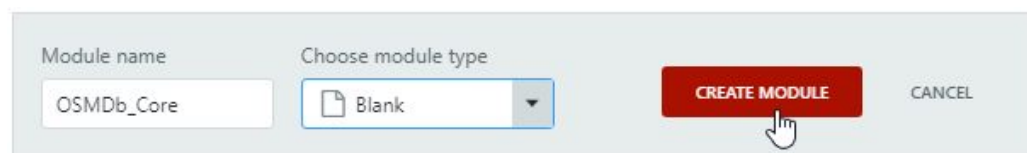
Modules allow you to structure your application into several pieces, each piece implementing a specific purpose.

- b. This will open a new tab with the OSMDb module in Service Studio. Go back to the previous section by selecting the tab with the environment information.



- c. Click on **Add Module** and define the new module to be *OSMDb_Core_<Your Initials>* and of type **Blank**.

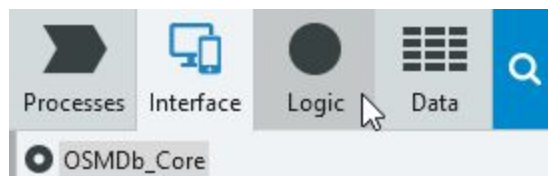
- d. Click on **Create Module**.



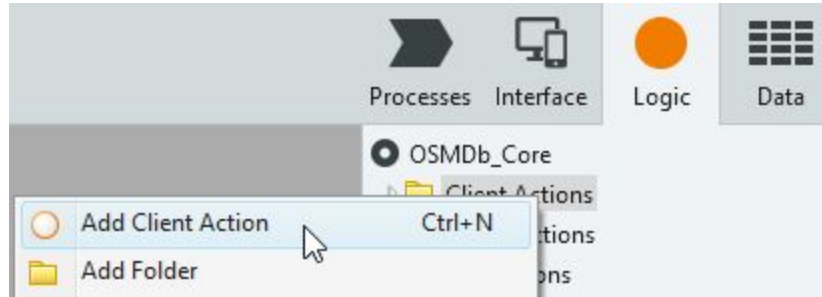
NOTE: From this point onwards, the application and module names will be referred to without the initials, for simplicity.

Create a Client Action

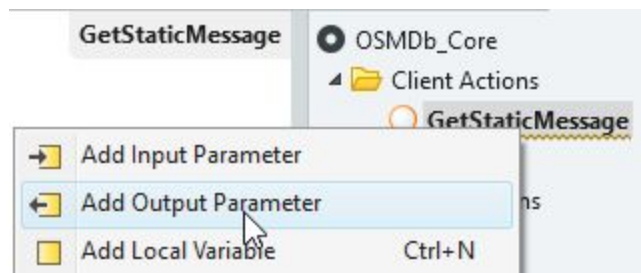
1. In the **OSMDb_Core**, create a Client Action called *GetStaticMessage*, with an Output Parameter named *Message*.
 - a. In the OSMDb_Core module, switch to the Logic tab of Service Studio.



- b. Right-click on the Client Actions folder and select **Add Client Action**.

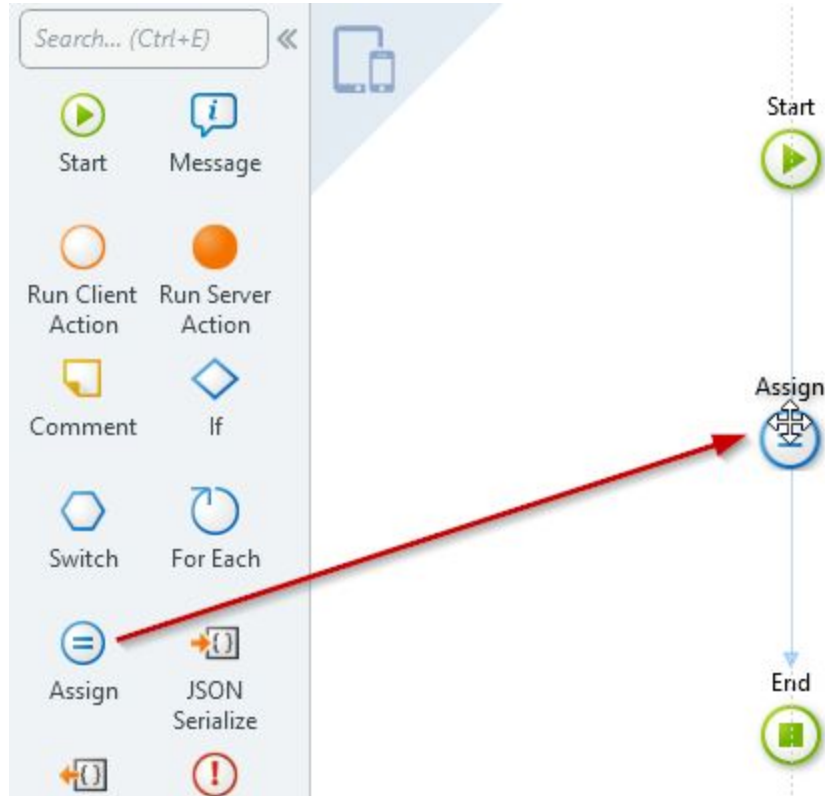


- c. Set its name to *GetStaticMessage*.
- d. Right-click the Action and select the option **Add Output Parameter**. Set its name to *Message*.

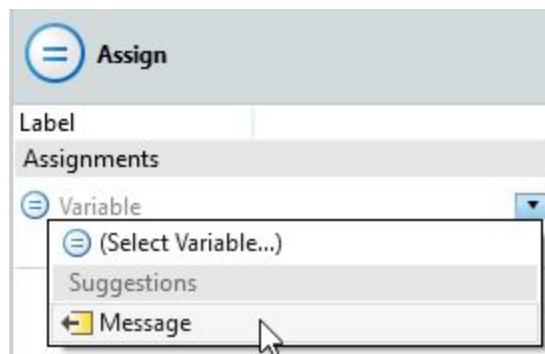


2. Define the logic of the Client Action. The Action should simply assign a static text message to the **Message** Output Parameter. Also, set the Action as **Public**, to be used in other modules, and as a **Function** to be used directly on Screens.

- a. Drag an **Assign** and drop it in the Action flow.



- b. Click on the Assign and in the properties area, under Assignments, select the **Message** Output Parameter.

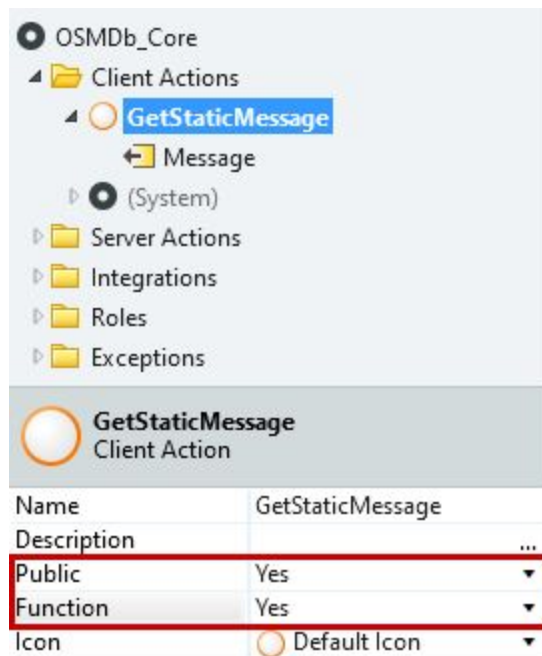


- c. Under **Value**, type a static message. For instance *"Hello from the OSMDb app"*.



NOTE: This is how we create a simple assignment in OutSystems, in this case we have: *Message = "Hello from the OSMDb app!"*

- d. Click on the **GetStaticMessage** Action and in its properties, set the **Public** option to Yes and the **Function** option to Yes.



GetStaticMessage Client Action	
Name	GetStaticMessage
Description	...
Public	Yes
Function	Yes
Icon	Default Icon

- e. Publish the OSMDb_Core module to save the changes in the server.

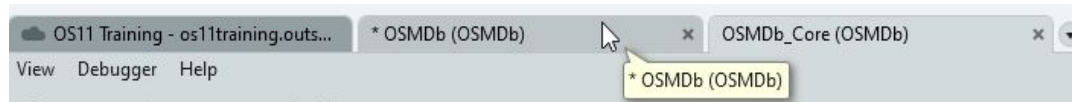


Use the Client Action on a Screen

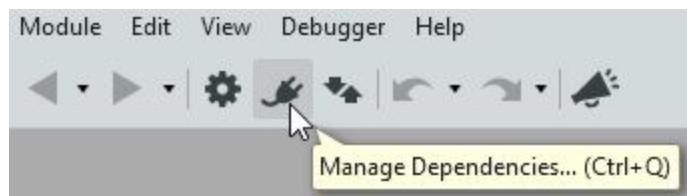
The GetStaticMessage Client Action was created in the OSMDb_Core module, but we want to use it in the OSMDb module, to display the static text on a Screen.

1. Reference the **GetStaticMessage** Client Action from the OSMDb module.

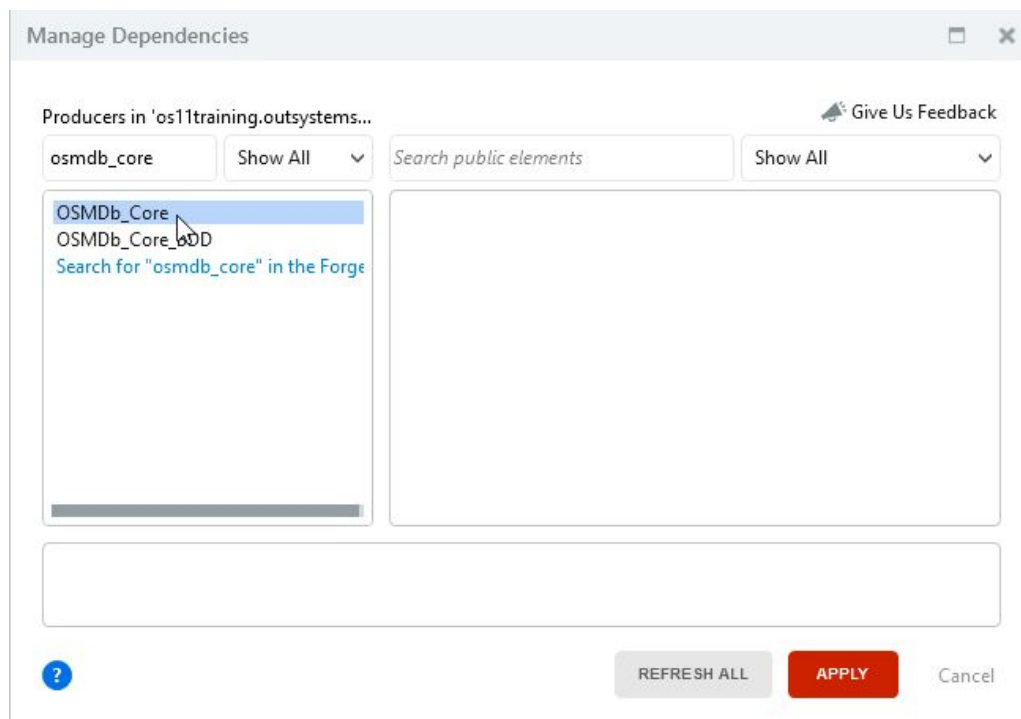
- a. Switch to the **OSMDb** module.



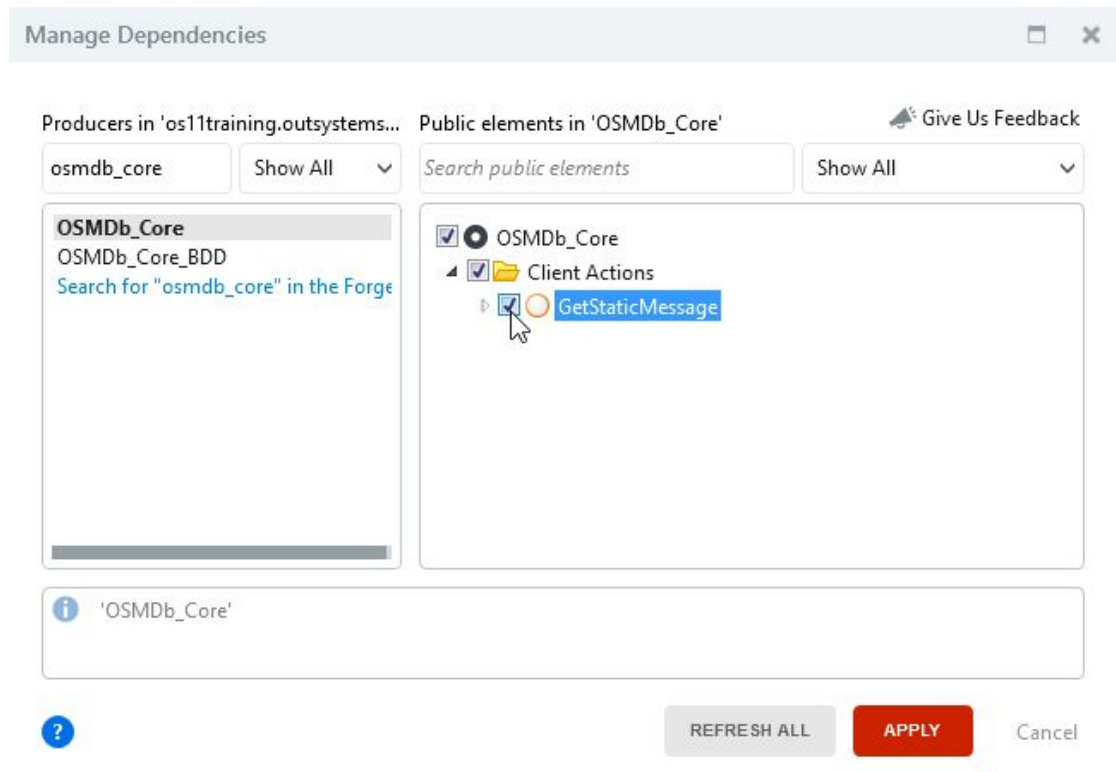
- b. Open the **Manage Dependencies** dialog.



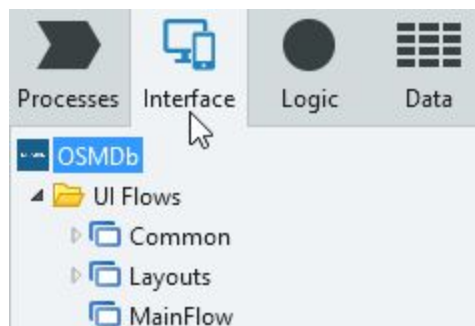
- c. A new dialog appears with all the modules in the environment that have public elements. Search for your **OSMDb_Core** module and click on it to select it.



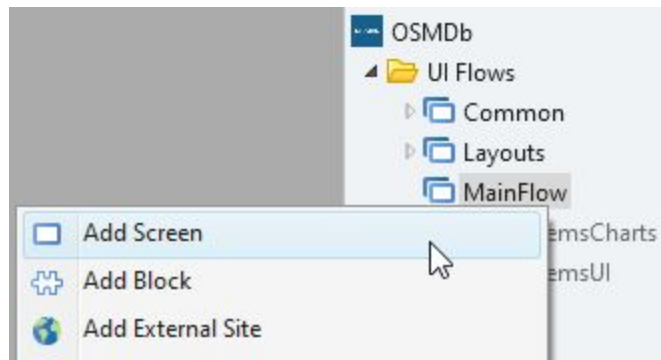
- d. Select the **GetStaticMessage** Action on the right. Click **Apply** to exit. This creates a reference to the Action created in the Core module.



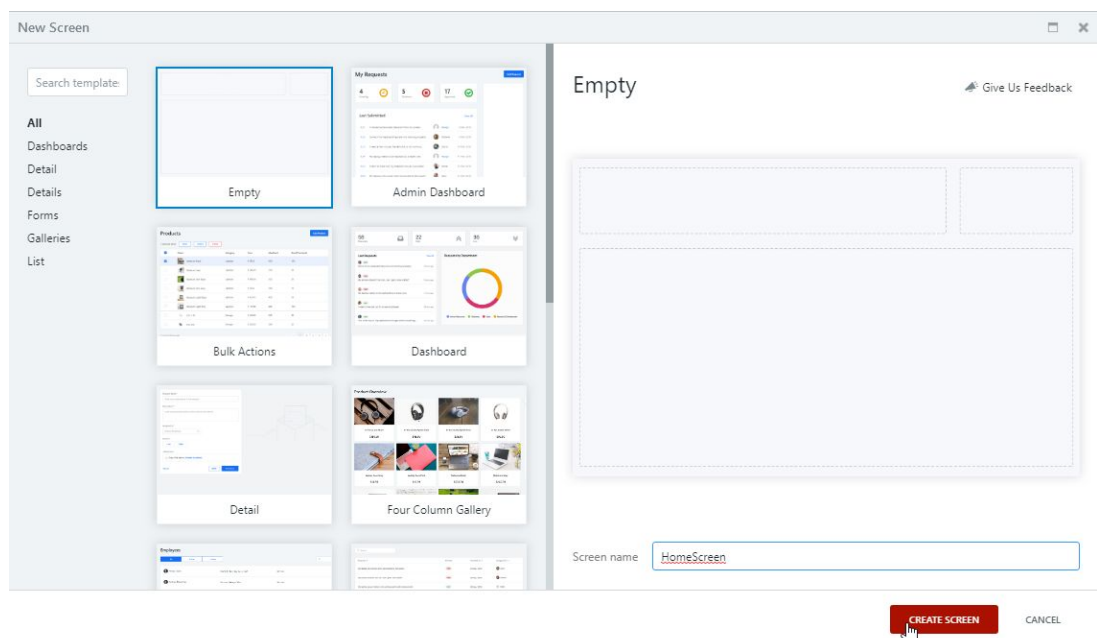
2. Create a new Screen, called *HomeScreen*. This Screen should simply display the static message in the output of the Client Action defined above.
 - a. Switch to the Interface tab in Service Studio.



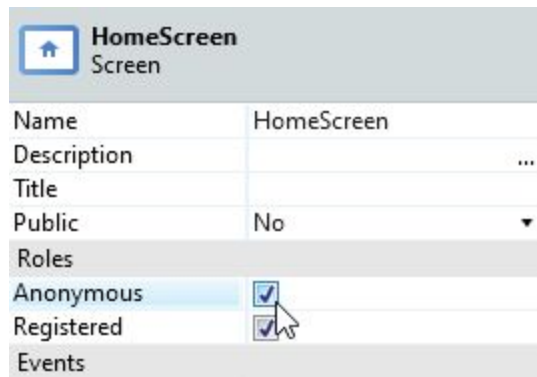
- b. Right-click on the MainFlow and select the option **Add Screen**.



- c. In the next dialog, select the **Empty** template, as we want an empty Screen. Set its name to *HomeScreen* and click **Create Screen** to finish.



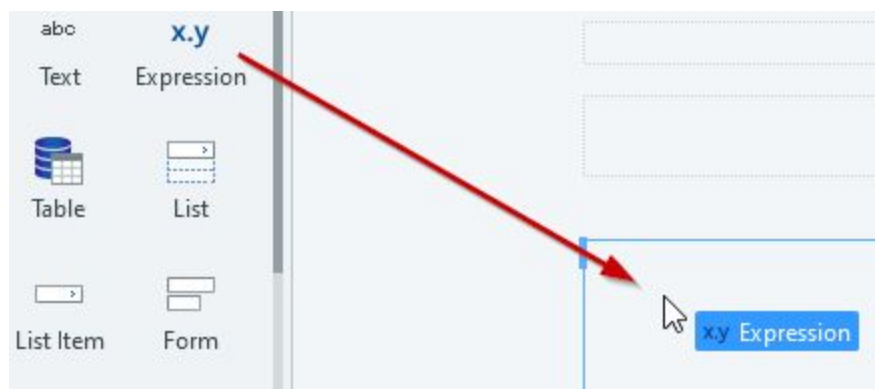
- d. Set the Screen to **Anonymous**, so everyone can access it.



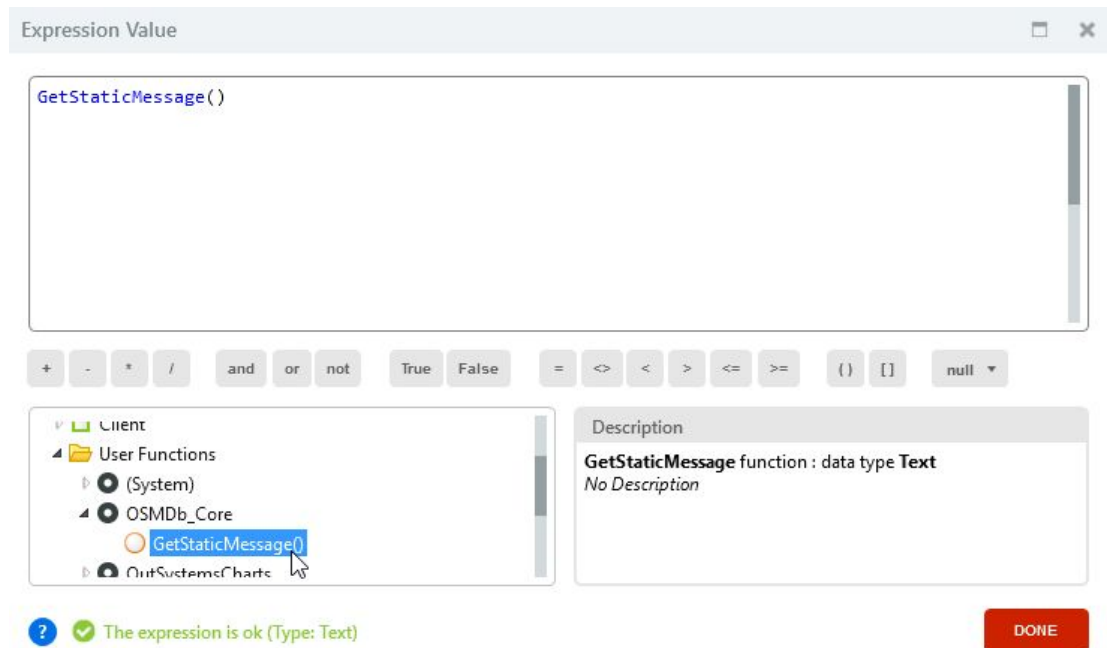
The screenshot shows the configuration form for a screen named 'HomeScreen'. The form includes fields for Name, Description, Title, Public status, Roles, Anonymous access, Registered access, and Events. The 'Anonymous' checkbox is checked, and a mouse cursor is hovering over it.

HomeScreen Screen	
Name	HomeScreen
Description	...
Title	
Public	No
Roles	
Anonymous	<input checked="" type="checkbox"/>
Registered	<input checked="" type="checkbox"/>
Events	

- e. Drag an **Expression** to the Screen.



- f. In the new dialog, expand the User Functions folder and the OSMDb_Core and select the **GetStaticMessage()** Client Action. Click **Done** to exit.



- g. Publish the module by clicking on the 1-click publish button.



- h. Open the application in the browser.



- i. Make sure the static message appears on the Screen.