Practical GXtest

Testing with GeneXus

Table of Contents

Objectives

This training module provides guidance on how to put into practice the main functionalities of GXtest. You will learn to create a *Unit Test* for a given procedure, to validate the database status, and to use mock databases when executing the tests.

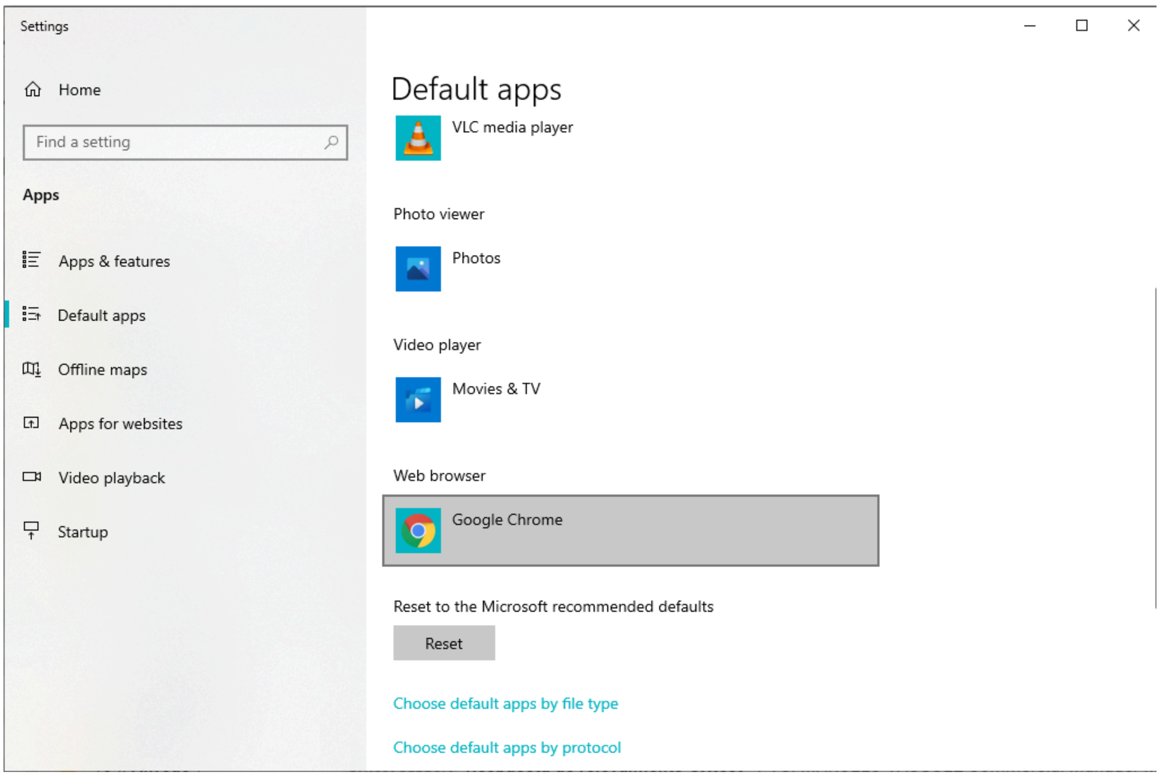
You will be introduced to the *Test Coverage* functionality by carrying out a test coverage analysis for a Unit Test. Then, you will learn to create and execute a *Rest Test* on a procedure portrayed as a REST service with GAM authentication. You will also create *UI tests* using *GXtest Recorder* and import their contents to a *Web UI test* in our KB. Finally, you will learn to build a *Test Suite*-type object to group the previously created tests into a single execution unit.

Environment Set-up

Before you start, you need to set up your environment as follows:

1) Set Google Chrome as your default Windows web browser.

a) Go to "Default apps" and set Google Chrome as web browser.



Default apps

VLC media player

Photo viewer: Photos

Video player: Movies & TV

Web browser: Google Chrome

Reset to the Microsoft recommended defaults

Reset

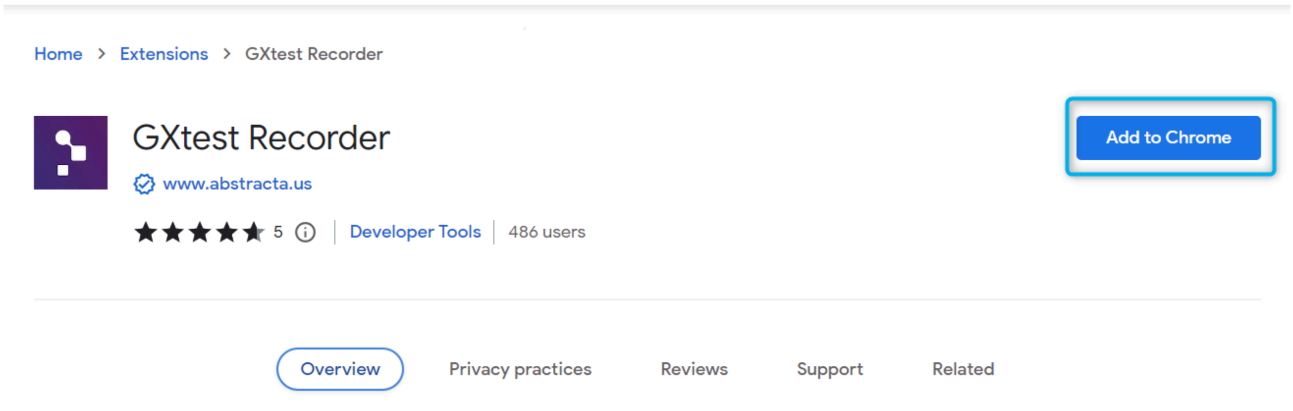
Choose default apps by file type

Choose default apps by protocol

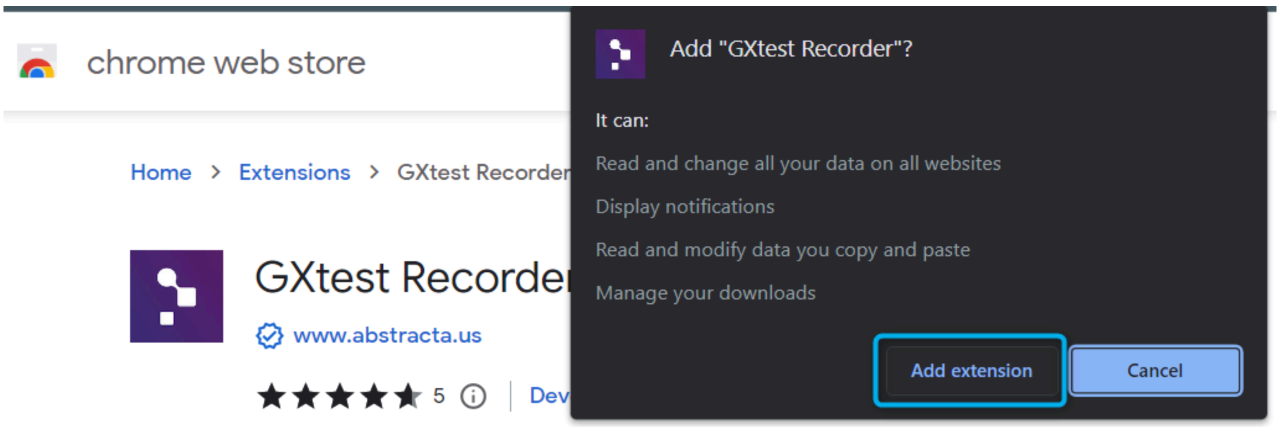
2) Install the GXtest Recorder extension.

a) Go to "Install GXtest Recorder".

b) Select "Add to Chrome".



c) Click on "Add extension".



It can:

Read and change all your data on all websites

Display notifications

Read and modify data you copy and paste

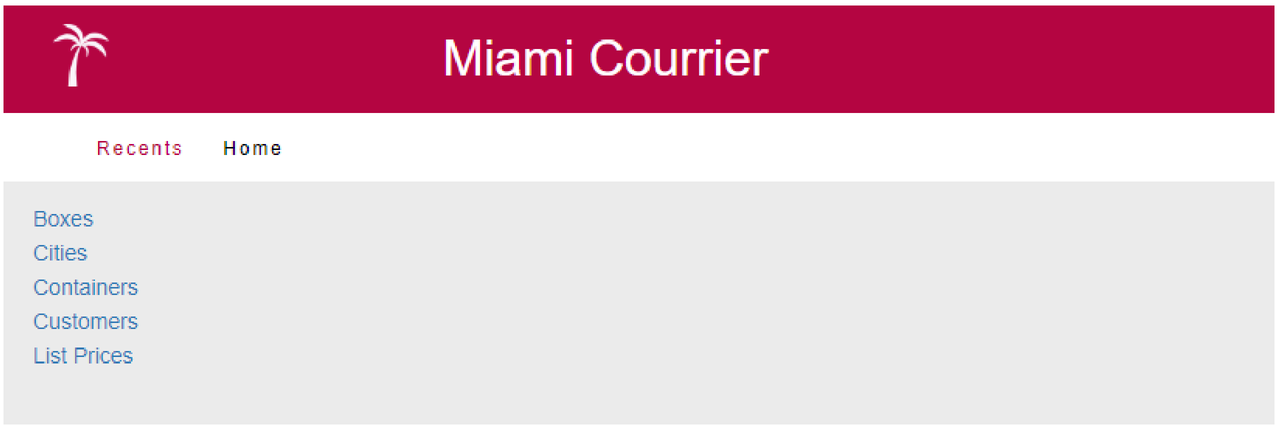
Manage your downloads

d) Close Google Chrome.

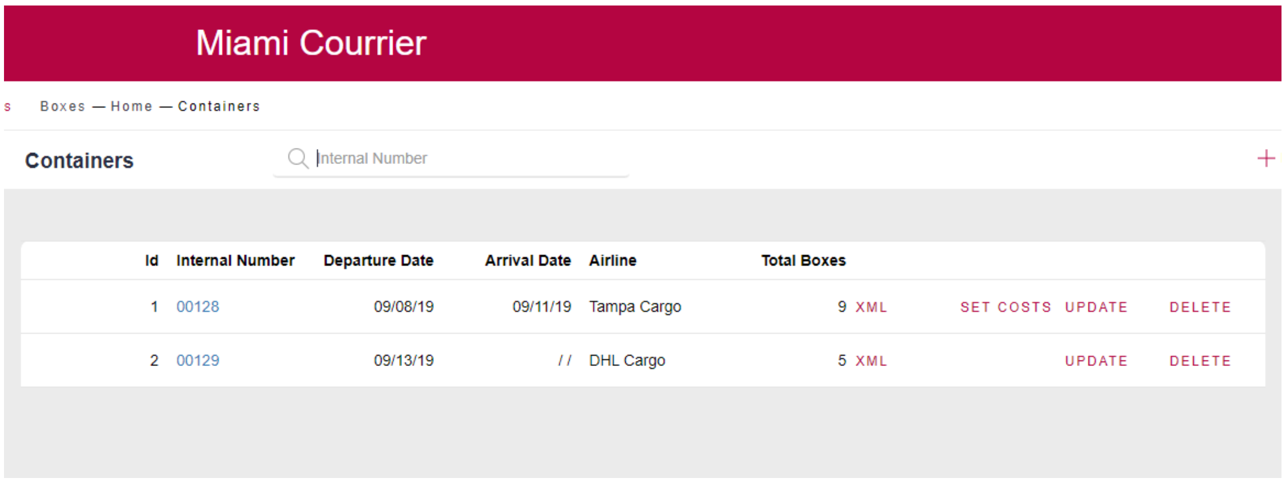
Application

The application you will be working on models a *Business Courier*; it manages shipping containers with their corresponding boxes, clients, and customs agencies. Let's take a look!

- Open the GeneXus IDE.
- Perform the KB checkout, named "**GXtestHandsOn**", from server https://samples.genexusserver.com/v18/dashboard.aspx by going to File > New > Knowledge Base from GeneXus Server and clicking on "Select server KB...", on "Add new server", and by completing the URL with "*https://samples.genexusserver.com/v18/dashboard.aspx*".
- Set As Start Object the "Home" panel in Web folder. Run the application from the GeneXus IDE (F5). The following window should open in your Chrome browser:

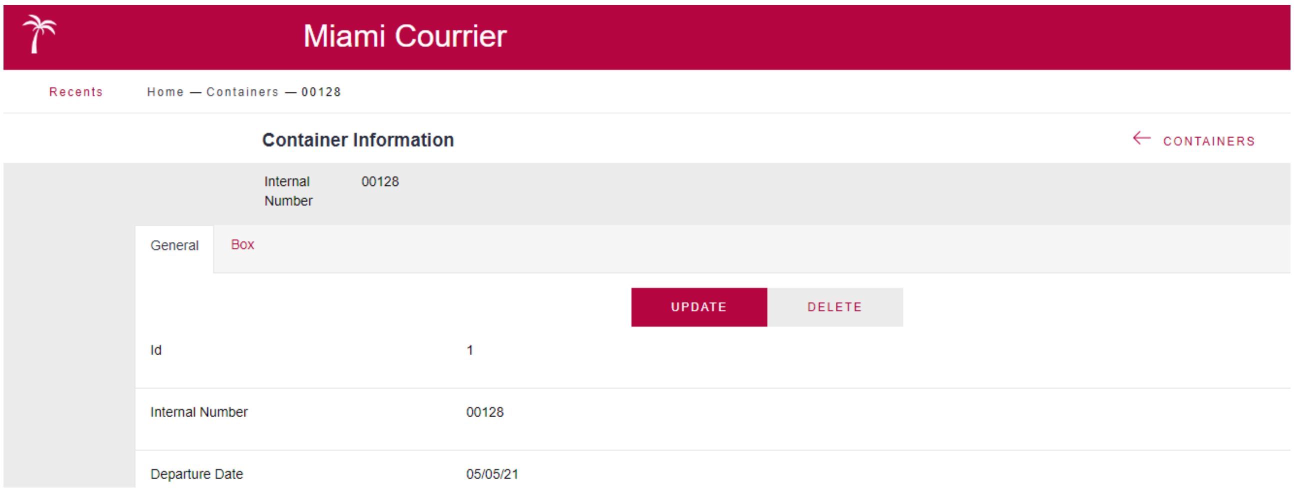


- Click on "Containers". You will see that the application has two containers with their corresponding boxes. Note that container 1 is holding 9 boxes.



Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes				
1	00128	09/08/19	09/11/19	Tampa Cargo	9	XML	SET COSTS	UPDATE	DELETE
2	00129	09/13/19	/ /	DHL Cargo	5	XML		UPDATE	DELETE

- Click on link "00128" of container 1. You will see the Container information:


Miami Courrier

Recents   Home — Containers — 00128

Container Information   ← CONTAINERS

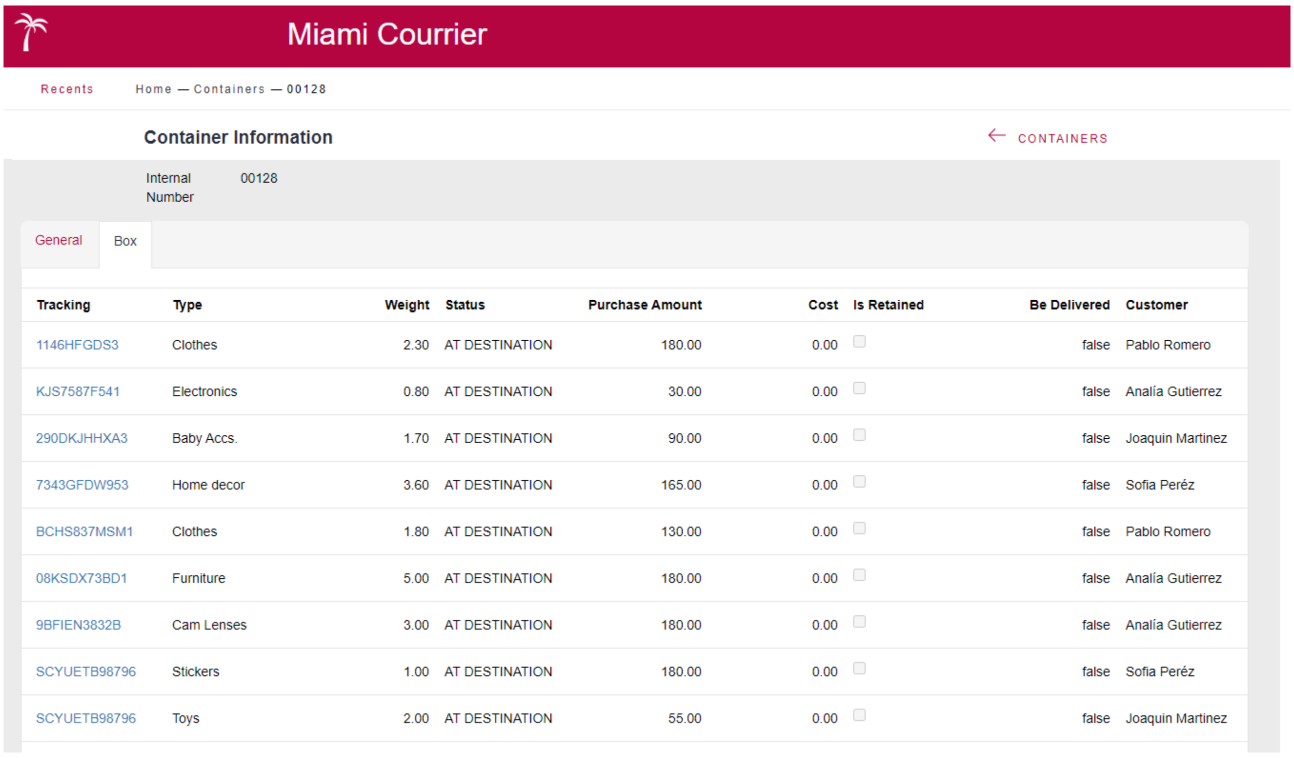
Internal Number   00128

General | Box

UPDATE   DELETE

Id	1
Internal Number	00128
Departure Date	05/05/21

- Click on the “Box” tab. Note that, in the “Cost” column, every box has a default price of 0. When a new box is created, its cost will be 0.


Miami Courrier

Recents   Home — Containers — 00128

Container Information   ← CONTAINERS

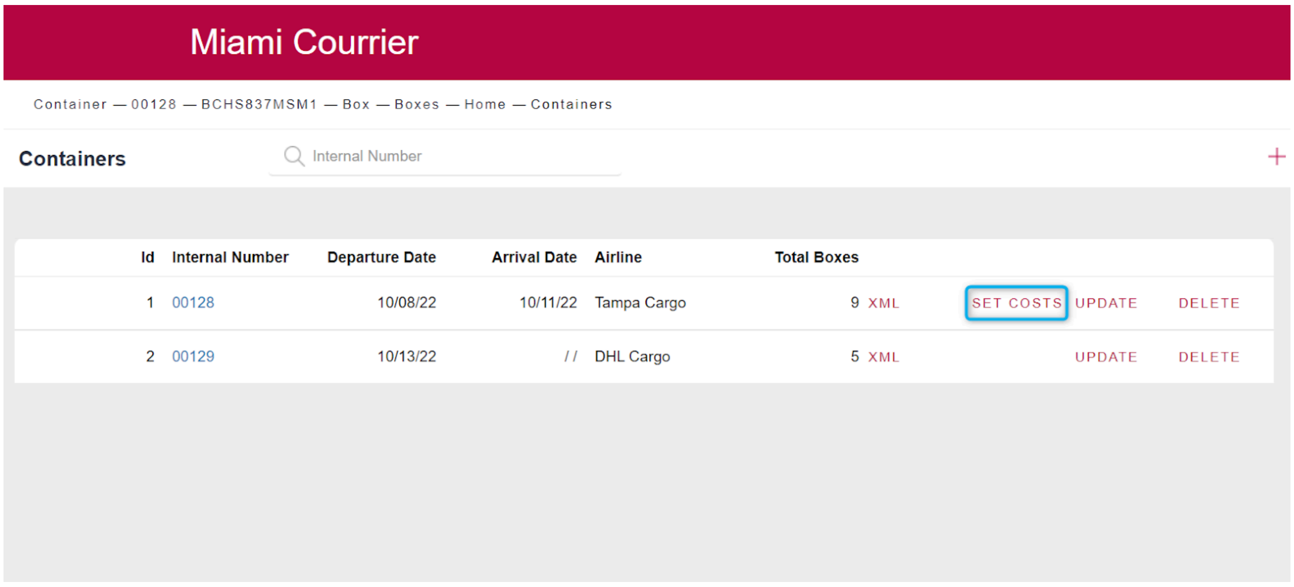
Internal Number   00128

General | Box

Tracking	Type	Weight	Status	Purchase Amount	Cost	Is Retained	Be Delivered	Customer
1146HFGDS3	Clothes	2.30	AT DESTINATION	180.00	0.00	☐	false	Pablo Romero
KJS7587F541	Electronics	0.80	AT DESTINATION	30.00	0.00	☐	false	Analía Gutierrez
290DKJHHXA3	Baby Accs.	1.70	AT DESTINATION	90.00	0.00	☐	false	Joaquin Martinez
7343GFDW953	Home decor	3.60	AT DESTINATION	165.00	0.00	☐	false	Sofia Peréz
BCHS837MSM1	Clothes	1.80	AT DESTINATION	130.00	0.00	☐	false	Pablo Romero
08KSDX73BD1	Furniture	5.00	AT DESTINATION	180.00	0.00	☐	false	Analía Gutierrez
9BFIEN3832B	Cam Lenses	3.00	AT DESTINATION	180.00	0.00	☐	false	Analía Gutierrez
SCYUETB98796	Stickers	1.00	AT DESTINATION	180.00	0.00	☐	false	Sofia Peréz
SCYUETB98796	Toys	2.00	AT DESTINATION	55.00	0.00	☐	false	Joaquin Martinez

Testing

Next, we will validate that the functionality **Set Cost to Boxes in Container** is working correctly. This functionality is implemented at an interface level on the "Containers" screen.
When clicking on the "SET COSTS" link, you can set the cost of the boxes associated to that container. Do not click on "SET COSTS", since this would alter the status of the application database.


Miami Courier
Container — 00128 — BCHS837MSM1 — Box — Boxes — Home — Containers
Containers
Internal Number
+

Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes				
1	00128	10/08/22	10/11/22	Tampa Cargo	9	XML	SET COSTS	UPDATE	DELETE
2	00129	10/13/22	/ /	DHL Cargo	5	XML		UPDATE	DELETE

You will execute five different *Unit Test* scenarios:
1) Unit Test pass: with correct values to execute the test successfully.
2) Unit Test fail: with incorrect values to find out what happens when a test fails.
3) Database validation: adding a database status validation to the Unit Test.
4) Execution with mock data: for the created Unit Test in scenario 1, using mock data recording.
5) Editing mock files for dynamic data: editing contents of mock recorded files in the Unit Test of the procedure **Set Container Arrival Date** to execute it any number of times using *Date*-type data.

You will also learn to create and execute a REST service test for the same procedure as in scenario 1, using GAM authentication.

Then, you will verify this functionality at the interface level, executing a UI test.

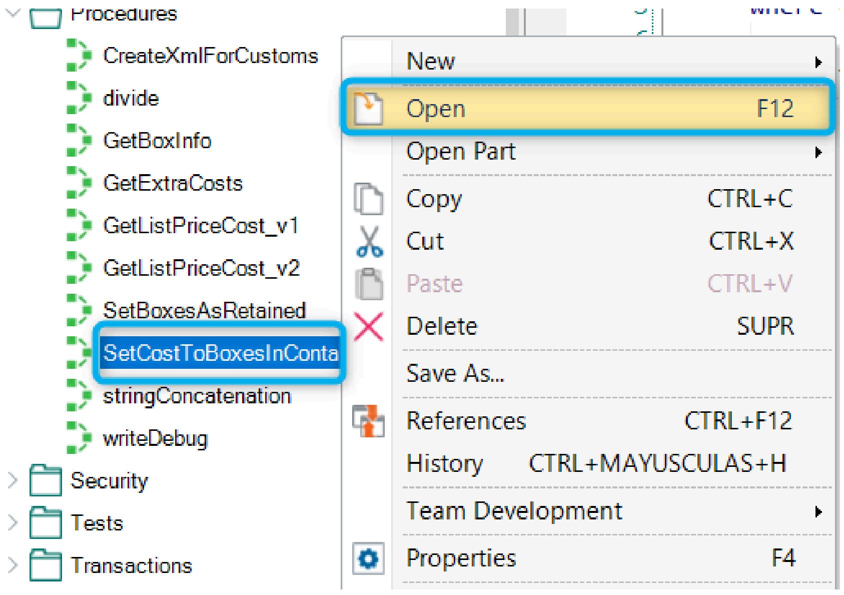
A test coverage analysis for a *Unit Test* created for the functionality **Get Extra Costs** for container boxes is also included, with the goal of covering 100% of the code of the objects consumed by the test.

Finally, you will learn to create a *Test Suite* object to group the previously created tests and to execute them as a unit in a specific order.

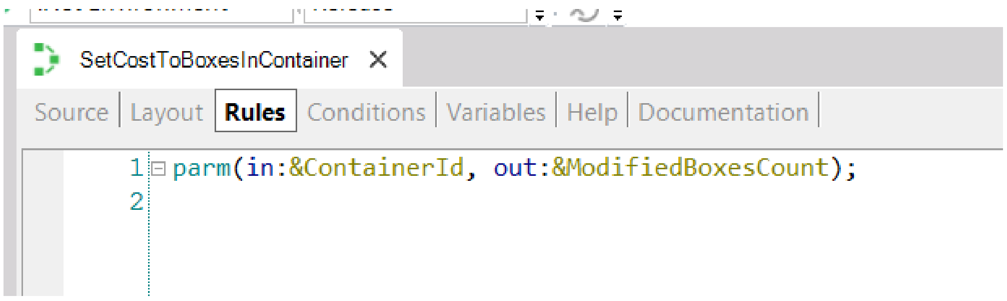
Unit Tests

In this section, we will guide you through executing different Unit Test scenarios.
The functionality **Set Cost to Boxes in Container** is implemented through the procedure ***"SetCostToBoxesInContainer"*** in our *KB*.
Let's take a look!

1) Open the procedure by right-clicking on it and selecting "Open" or simply by double-clicking it.

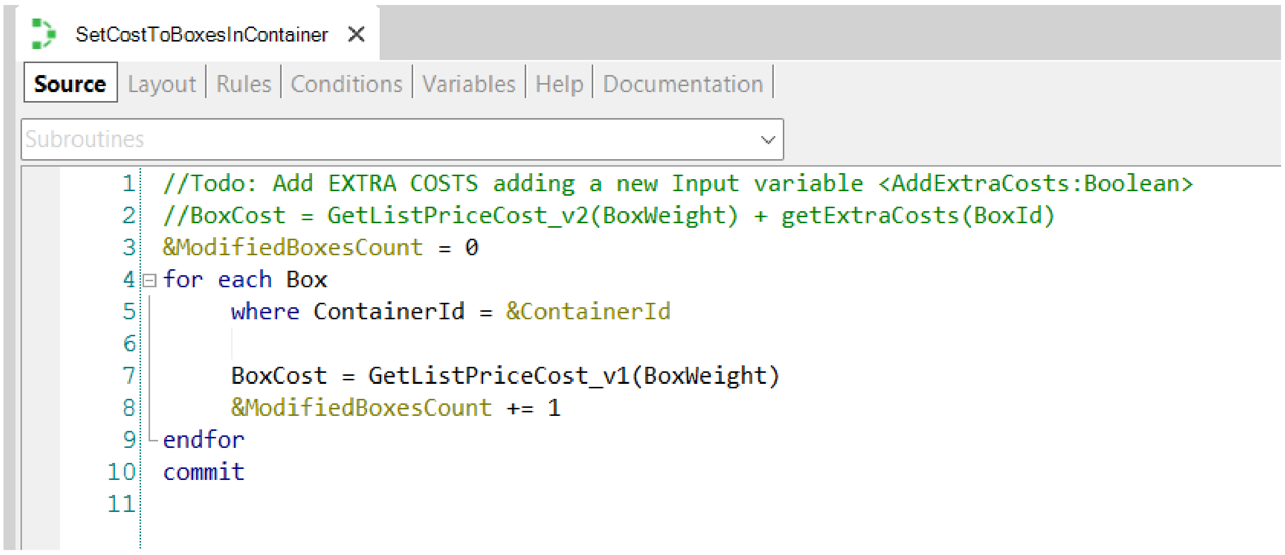


2) Click on the *Rules* tab. You will see that the procedure receives *&ContainerId* as input (this is the container whose boxes you will set the cost to) and returns *&ModifiedBoxesCount* as output (number of modified boxes):



```
parm(in:&ContainerId, out:&ModifiedBoxesCount);
```

3) Click on the *Source* tab. As you can see, the code of the procedure assigns a cost to each box in the container whose ID matches the input parameter *&ContainerId*:


SetCostToBoxesInContainer ✕

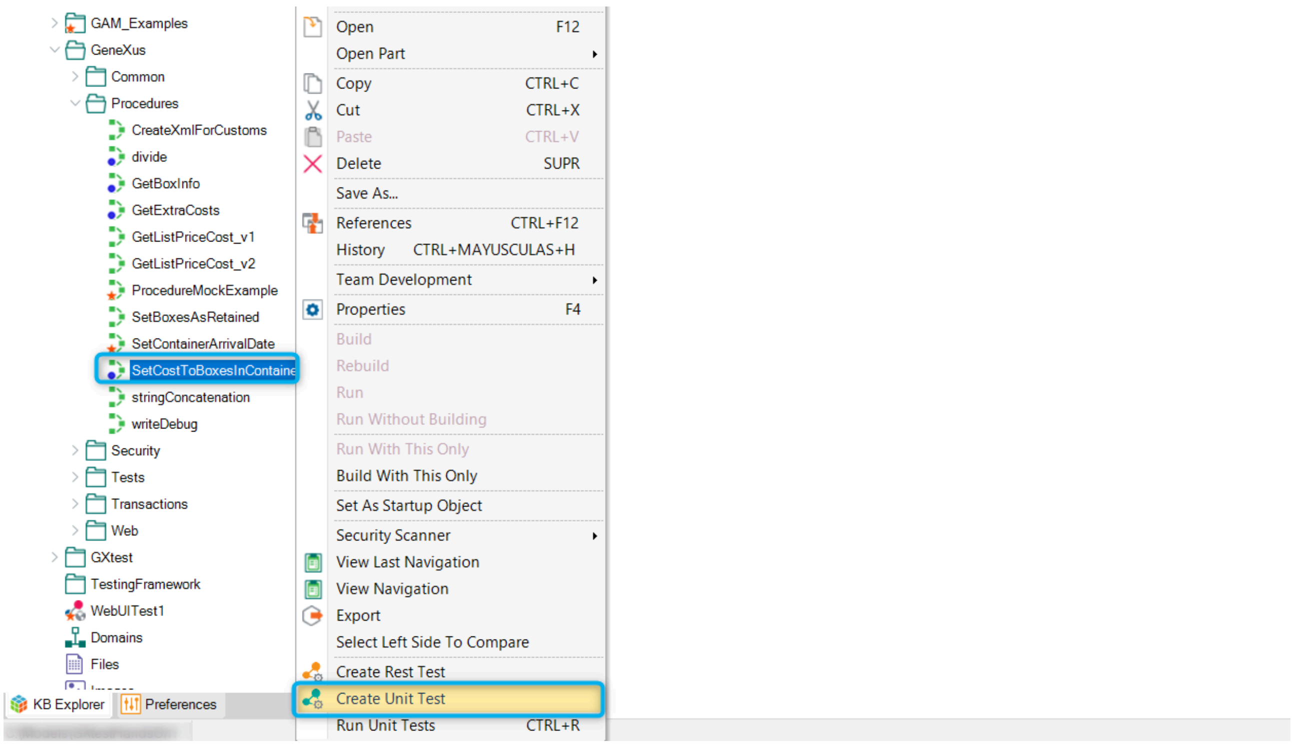
Source | Layout | Rules | Conditions | Variables | Help | Documentation

Subroutines

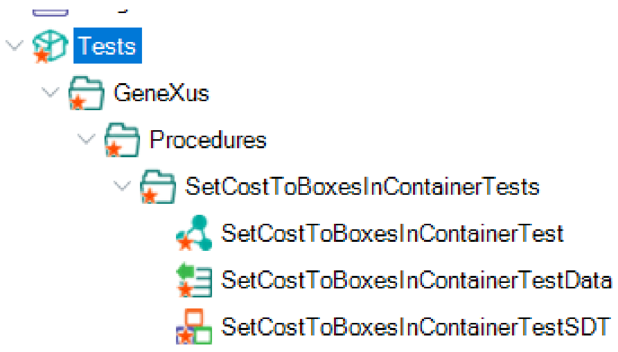
```
//Todo: Add EXTRA COSTS adding a new Input variable <AddExtraCosts:Boolean>
//BoxCost = GetListPriceCost_v2(BoxWeight) + getExtraCosts(BoxId)
&ModifiedBoxesCount = 0
for each Box
      where ContainerId = &ContainerId

      BoxCost = GetListPriceCost_v1(BoxWeight)
      &ModifiedBoxesCount += 1
endfor
commit

```

Creating a Unit Test from a Procedure

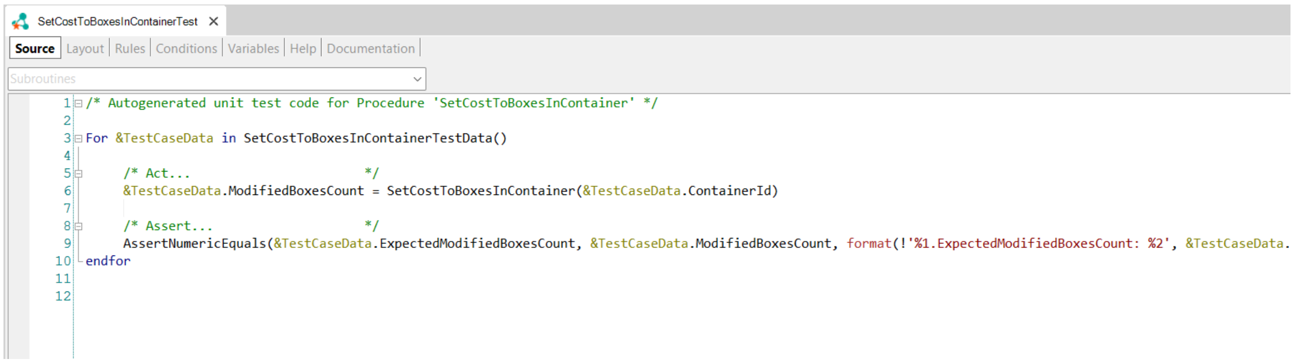
1) Select the procedure "SetCostToBoxesInContainer" in the *KB Explorer* screen, right-click, and choose the option "Create Unit Test".



2) Wait until it generates the test objects “SetCostToBoxesInContainerTest”, “SetCostToBoxesInContainerTestData”, and “SetCostToBoxesInContainerTestSDT”, which will appear in the “Tests” module inside the folder with the procedure named “SetCostToBoxesInContainerTests”.


Tests
GeneXus
Procedures
SetCostToBoxesInContainerTests
SetCostToBoxesInContainerTest
SetCostToBoxesInContainerTestData
SetCostToBoxesInContainerTestSDT

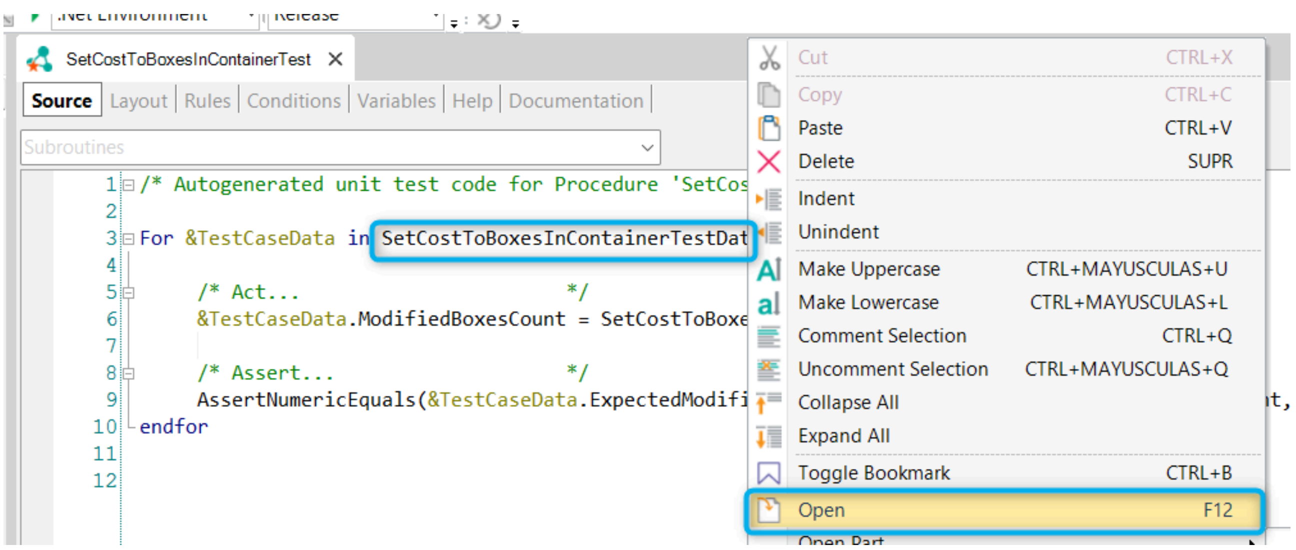
3) As you can see, GXtest automatically generates a GeneXus code template by iterating on the Data Provider “SetCostToBoxesInContainerTestData”, which contains the dataset that the test will execute.



```
/* Autogenerated unit test code for Procedure 'SetCostToBoxesInContainer' */

For &TestCaseData in SetCostToBoxesInContainerTestData()

     /* Act...                          */
     &TestCaseData.ModifiedBoxesCount = SetCostToBoxesInContainer(&TestCaseData.ContainerId)

     /* Assert...                       */
     AssertNumericEquals(&TestCaseData.ExpectedModifiedBoxesCount, &TestCaseData.ModifiedBoxesCount, format(!'%1.ExpectedModifiedBoxesCount: %2', &TestCaseData.
endfor
```

Unit Test Pass

1) Right-click on the object “SetCostToBoxesInContainerTestData” and select “Open”.



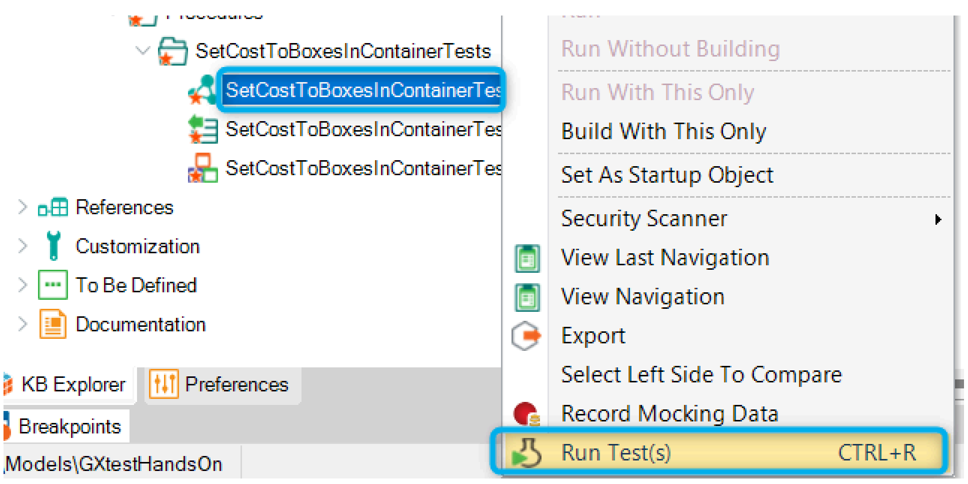
```
/* Autogenerated unit test code for Procedure 'SetCos
For &TestCaseData in SetCostToBoxesInContainerTestDat

    /* Act...                    */
    &TestCaseData.ModifiedBoxesCount = SetCostToBoxe

    /* Assert...                 */
    AssertNumericEquals(&TestCaseData.ExpectedModifi
endfor
```

2) Input the values ***ContainerId* = 1, *ExpectedModifiedBoxesCount* = 9** and ***MsgModifiedBoxesCount* = '9'** in the object SetCostToBoxesInContainerTestData.

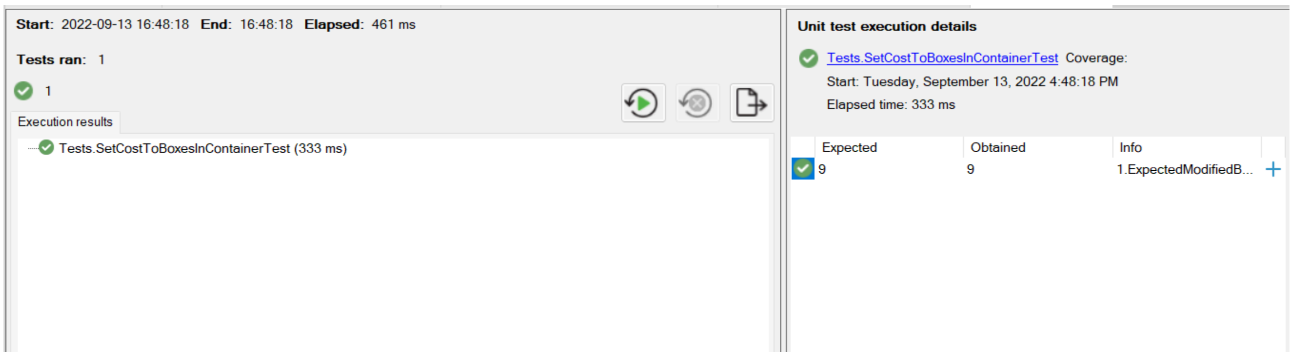
The Data Provider code is:

```
SetCostToBoxesInContainerTestSDT
{
        TestCaseId = '1'
        ContainerId = 1
        ExpectedModifiedBoxesCount = 9
        MsgModifiedBoxesCount = '9'
}
```

3) Save changes, right-click on the Unit Test, and select the option “Run Test(s)”.



4) When the test execution is finished, the results will be displayed in the Test Results panel. As you can see, the test has passed, since the expected number of modified boxes is the same as the number obtained.



Start: 2022-09-13 16:48:18 **End:** 16:48:18 **Elapsed:** 461 ms

Tests ran: 1

✓ 1

Execution results

✓ Tests.SetCostToBoxesInContainerTest (333 ms)

Unit test execution details

✓ Tests.SetCostToBoxesInContainerTest Coverage:

Start: Tuesday, September 13, 2022 4:48:18 PM

Elapsed time: 333 ms

	Expected	Obtained	Info
✓	9	9	1.ExpectedModifiedB...

Unit Test Fail

To execute a test which is expected to fail, you must modify the Data Provider values so that the expected number is different to the number obtained which returns the result "9".

Identify the fields to be edited in the object "SetCostToBoxesInContainerTestData", change the numbers, and execute the test by selecting the option "Run Test(s)".

The test will fail, displaying the differences between the obtained result and the expected result.

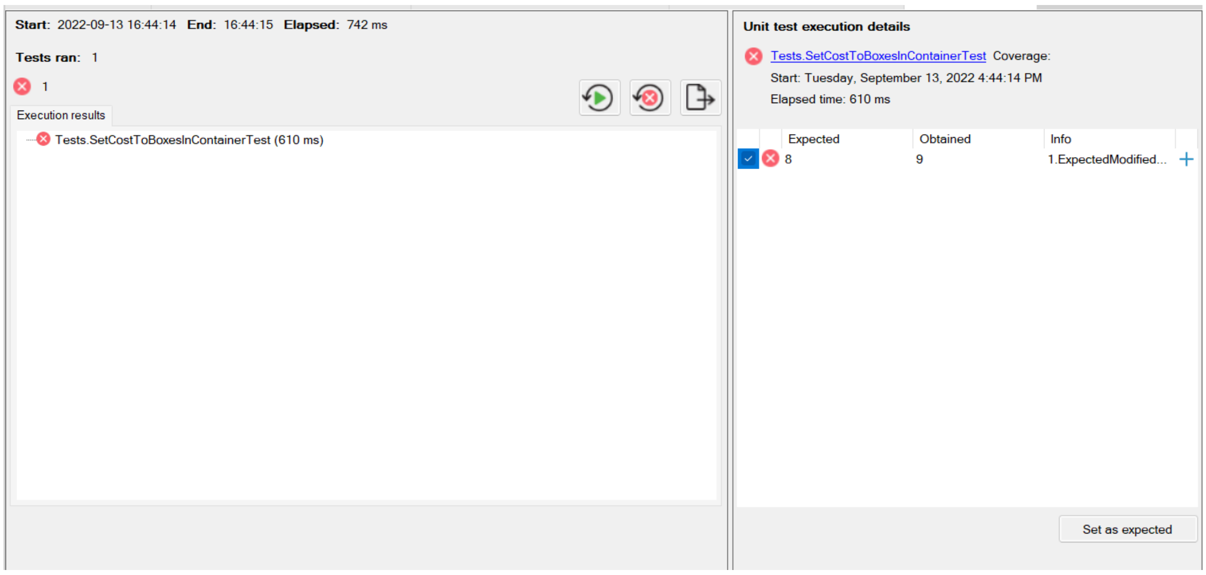
SOLUTION:

1) Open the object "SetCostToBoxesInContainerTestData".
2) Modify values **ExpectedModifiedBoxesCount = 8** and **MsgModifiedBoxesCount = '8'**.

The Data Provider code is:

```
SetCostToBoxesInContainerTestSDT
{
        TestCaseId = '1'
        ContainerId = 1
        ExpectedModifiedBoxesCount = 8
        MsgModifiedBoxesCount = '8'
}
```

3) Save changes and execute the test by selecting the option "Run Test(s)".
4) Observe how, when the test is finished, the results will be displayed in the Test Results screen, indicating the errors and the difference between the expected result and the obtained result.


Start: 2022-09-13 16:44:14 **End**: 16:44:15 **Elapsed**: 742 ms

Tests ran: 1

Execution results

Tests.SetCostToBoxesInContainerTest (610 ms)

Unit test execution details

Tests.SetCostToBoxesInContainerTest Coverage:

Start: Tuesday, September 13, 2022 4:44:14 PM

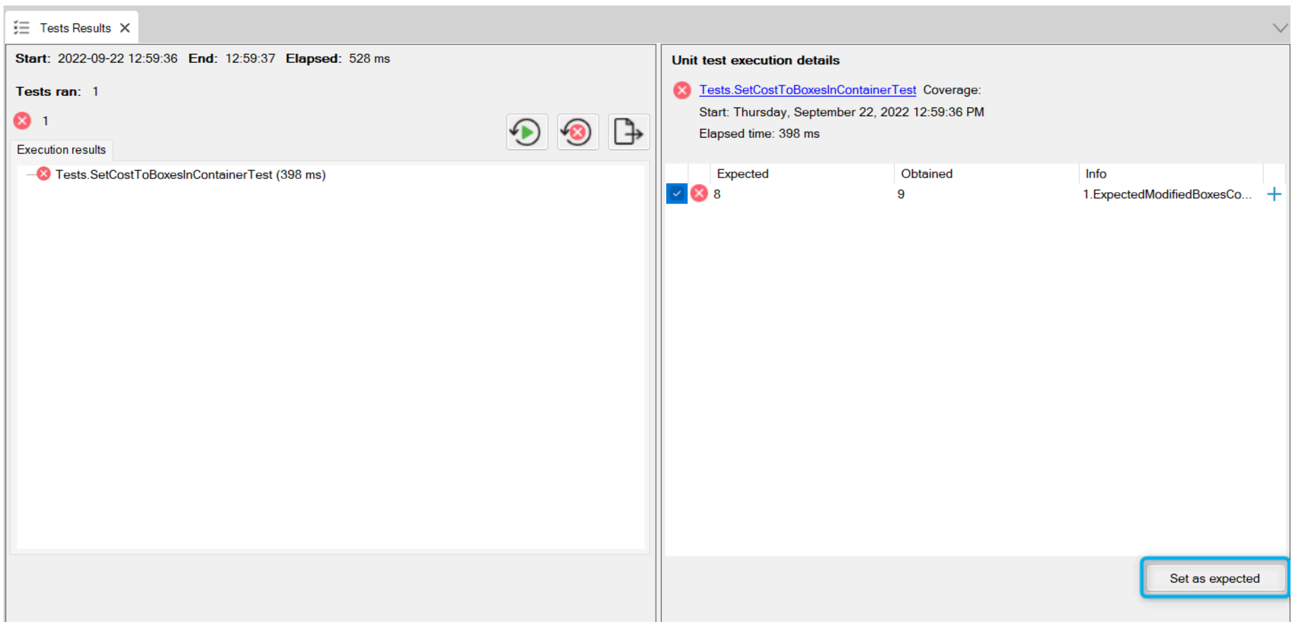
Elapsed time: 610 ms

Expected	Obtained	Info
8	9	1.ExpectedModified...

Set As Expected

There is a functionality for setting the expected results with the obtained results of the execution, always you are sure that the obtained values are correct. If you click on the “**Set as expected**” button, the expected results in the Data Provider will substitute. Let's try with the last example just to understand the functionality!

1) Click on “Set as expected” in the Test Result screen.



Start: 2022-09-22 12:59:36 **End**: 12:59:37 **Elapsed**: 528 ms

Tests ran: 1

Execution results

Tests.SetCostToBoxesInContainerTest (398 ms)

Unit test execution details

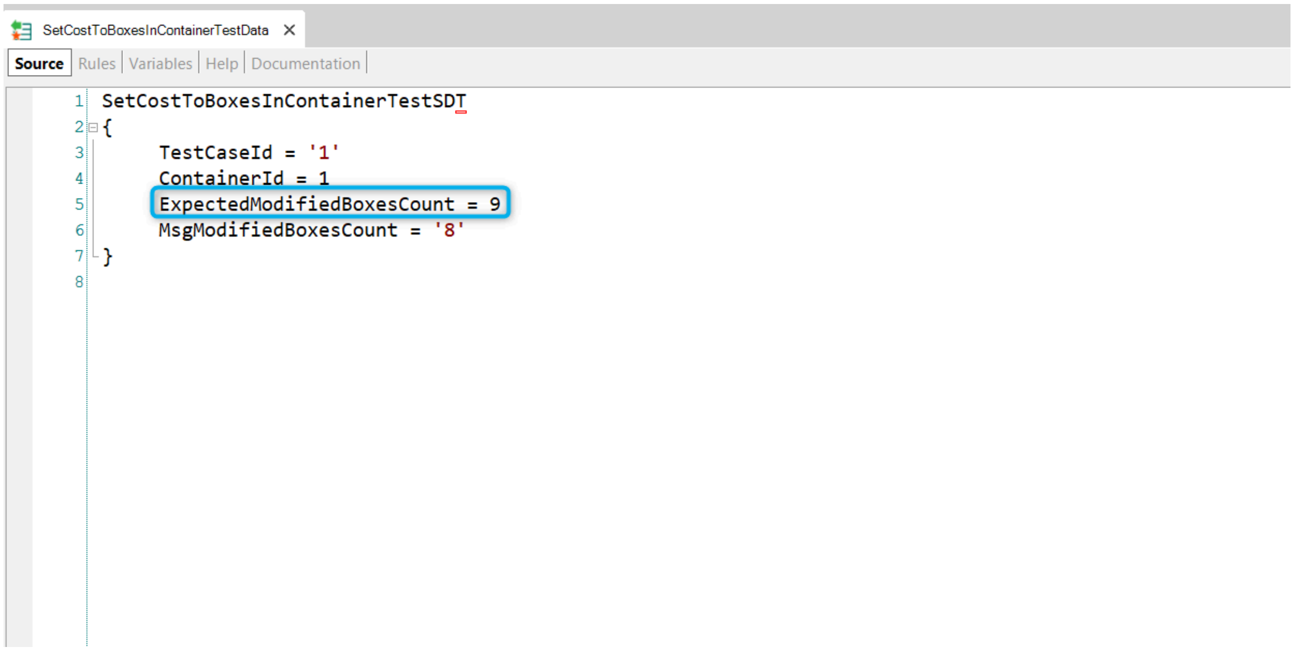
Tests.SetCostToBoxesInContainerTest Coverage:

Start: Thursday, September 22, 2022 12:59:36 PM

Elapsed time: 398 ms

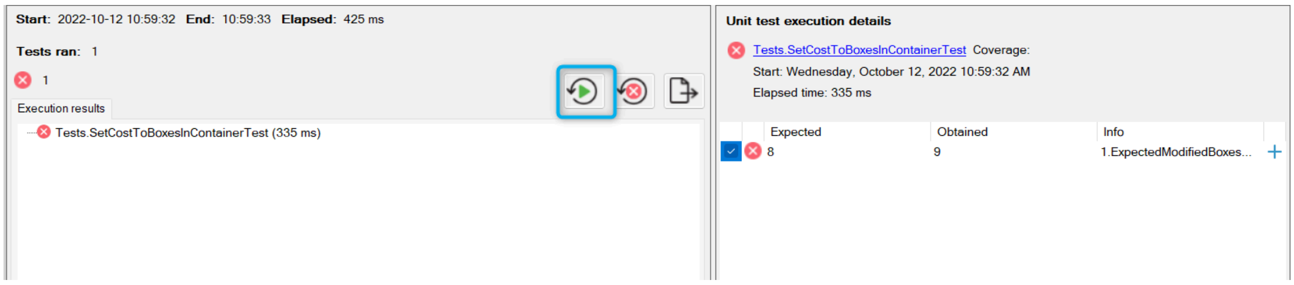
Expected	Obtained	Info
8	9	1.ExpectedModifiedBoxesCo...

2) Open the object “SetCostToBoxesInContainerTestData” and see how the number in “ExpectedModifiedBoxesCount” has changed to 9.



```
SetCostToBoxesInContainerTestSDT
{
	TestCaseId = '1'
	ContainerId = 1
	ExpectedModifiedBoxesCount = 9
	MsgModifiedBoxesCount = '8'
}
```

3) Run the test "SetCostToBoxesInContainerTest" again by clicking on "Run Again".



Start: 2022-10-12 10:59:32 **End**: 10:59:33 **Elapsed**: 425 ms

Tests ran: 1

✗ 1

Execution results

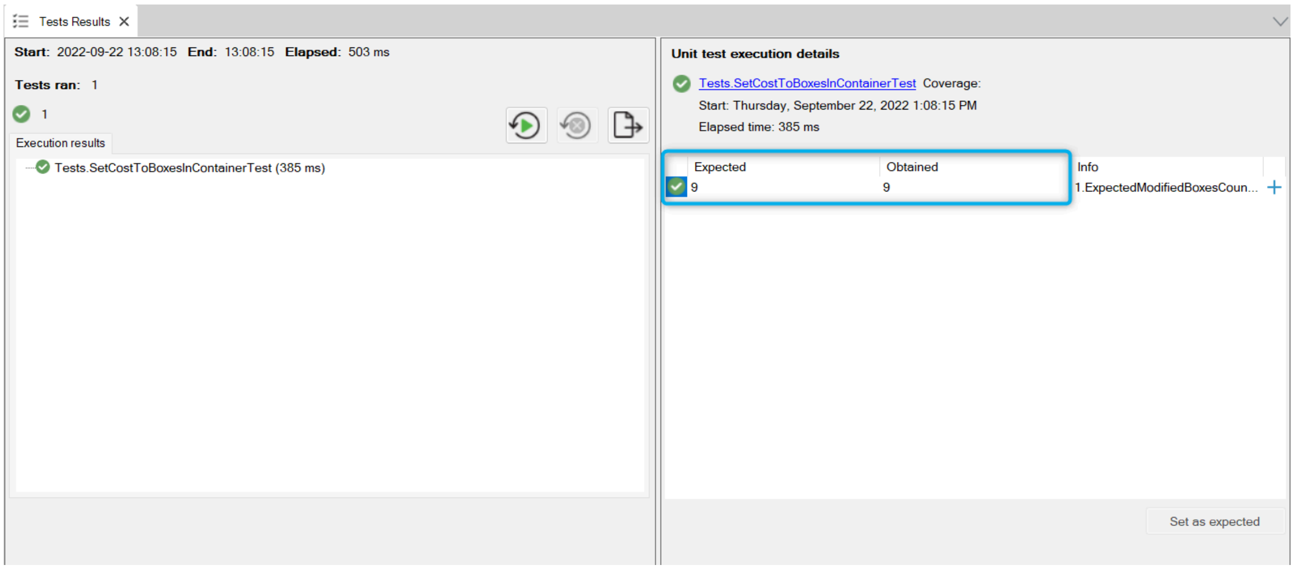
✗ Tests.SetCostToBoxesInContainerTest (335 ms)

Unit test execution details

✗ Tests.SetCostToBoxesInContainerTest Coverage:
Start: Wednesday, October 12, 2022 10:59:32 AM
Elapsed time: 335 ms

	Expected	Obtained	Info
✓ ✗	8	9	1.ExpectedModifiedBoxes...

4) As you can see, the test has now passed, obtaining the same result which has been set as expected.



Start: 2022-09-22 13:08:15 **End**: 13:08:15 **Elapsed**: 503 ms

Tests ran: 1

✓ 1

Execution results

✓ Tests.SetCostToBoxesInContainerTest (385 ms)

Unit test execution details

✓ Tests.SetCostToBoxesInContainerTest Coverage:
Start: Thursday, September 22, 2022 1:08:15 PM
Elapsed time: 385 ms

	Expected	Obtained	Info
✓	9	9	1.ExpectedModifiedBoxesCoun...

Note that this feature is used in the first test execution (typically with the Expected result set as zero or null) to set the expected results of the test cases. The previous example is just to show how to work it.

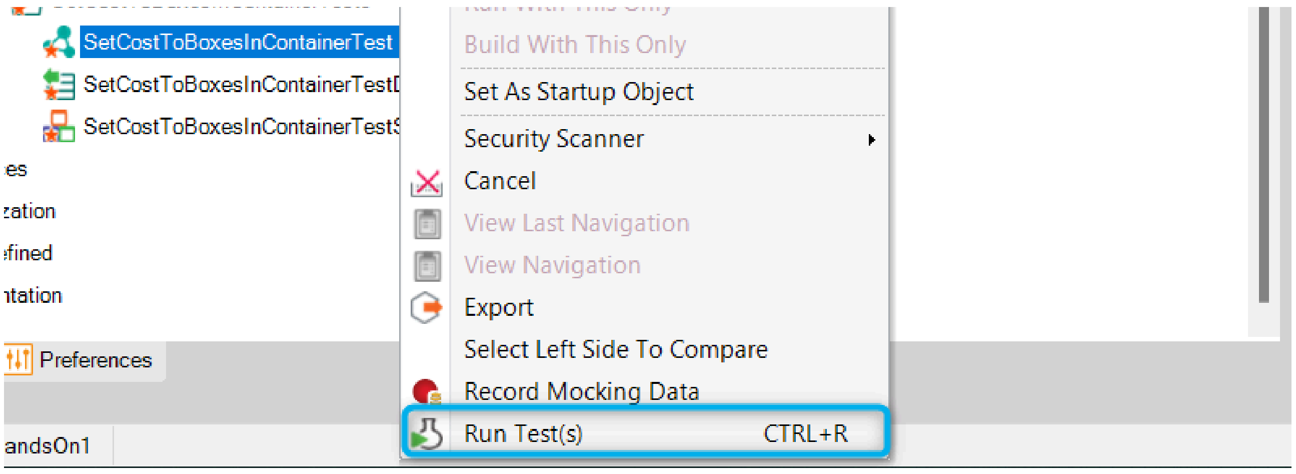
Validation of Database Status

You will now validate that, when the procedure is executed, the database actually changes its status. To do this, you will add an assertion to check the database status. In this case, the validation will check if the cost of every box of the container has a value higher than 0.

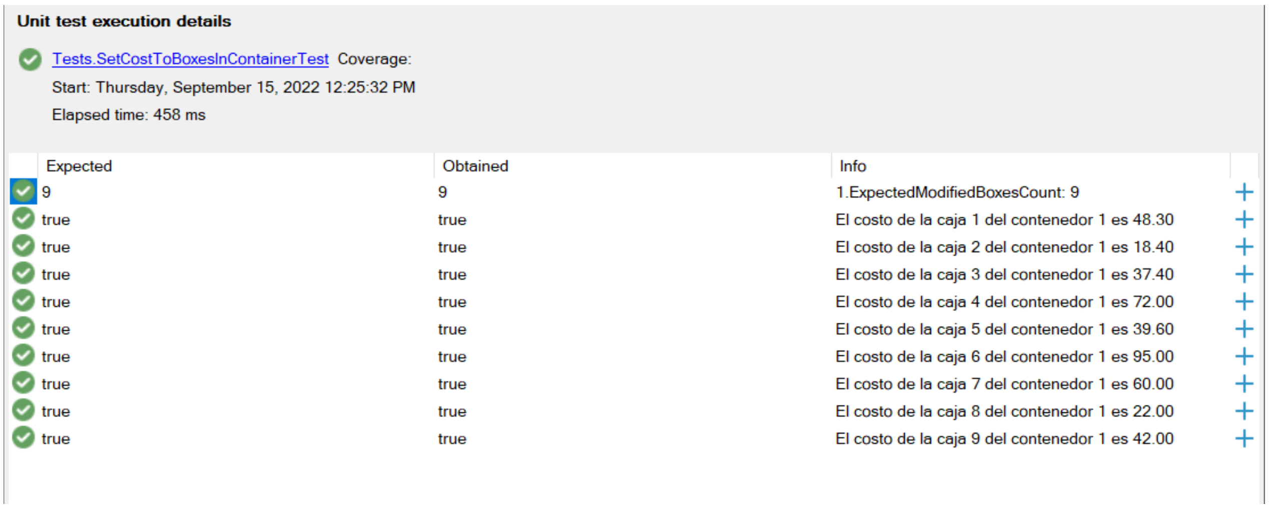
1) Open the object “SetCostToBoxesInContainerTest”.
2) Add the following assertion under the existing one:

```
/* Assertion of database status. */

for each Box

    where ContainerId = &TestCaseData.ContainerId

AssertBoolEquals(true, BoxCost > 0, Format("El costo de la caja %1 del contenedor %2 es %3",BoxId,
ContainerId, BoxCost))

endfor
```

3) Save changes and run the test by right-clicking on the Unit Test object and selecting the option “Run Test(s)”.



On the results screen, you will see the validation of the procedure output parameters in the first line and, in the remaining lines, the validations of the database status of the cost of each box.


Unit test execution details

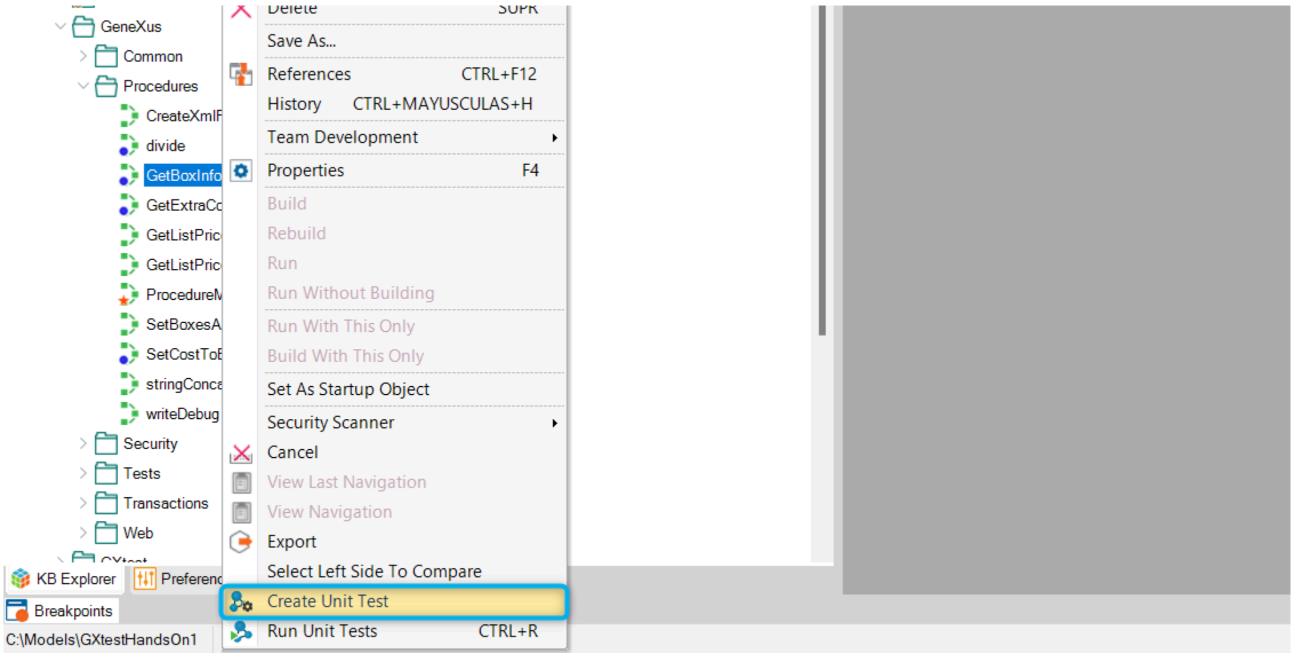
Tests.SetCostToBoxesInContainerTest Coverage:
Start: Thursday, September 15, 2022 12:25:32 PM
Elapsed time: 458 ms

Expected	Obtained	Info
9	9	1.ExpectedModifiedBoxesCount: 9
true	true	El costo de la caja 1 del contenedor 1 es 48.30
true	true	El costo de la caja 2 del contenedor 1 es 18.40
true	true	El costo de la caja 3 del contenedor 1 es 37.40
true	true	El costo de la caja 4 del contenedor 1 es 72.00
true	true	El costo de la caja 5 del contenedor 1 es 39.60
true	true	El costo de la caja 6 del contenedor 1 es 95.00
true	true	El costo de la caja 7 del contenedor 1 es 60.00
true	true	El costo de la caja 8 del contenedor 1 es 22.00
true	true	El costo de la caja 9 del contenedor 1 es 42.00

Execution with Database Mocking

To understand **Database Mocking** functionality, suppose that you want to test the GetBoxInfo procedure, given the BoxId received by parameter it returns the information about that box. So, we will create a test case with mock to check the information of BoxId = 1 of Container = 1.

1) Create a Unit Test for the procedure "**GetBoxInfo**" in our KB by right-clicking it and selecting the option "Create Unit Test".

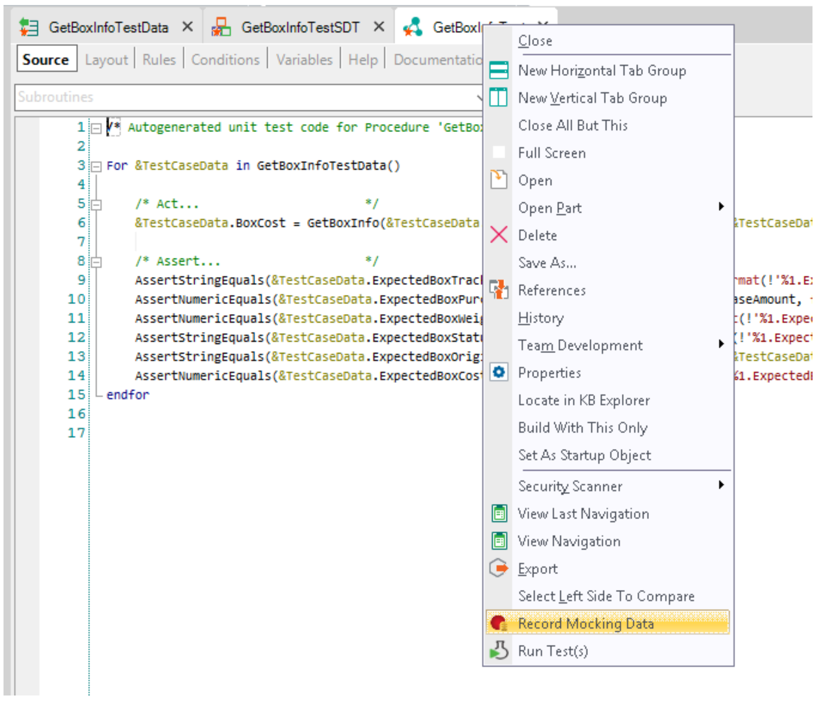


2) Open the object "GetBoxInfoTestData" and edit the value "***ExpectedBoxCost = 48.3***". If you want all the assertions to pass, you will also need to complete the rest of the assertions with the data of "BoxId=1", which can be seen in the application.

The Data Provider code is:

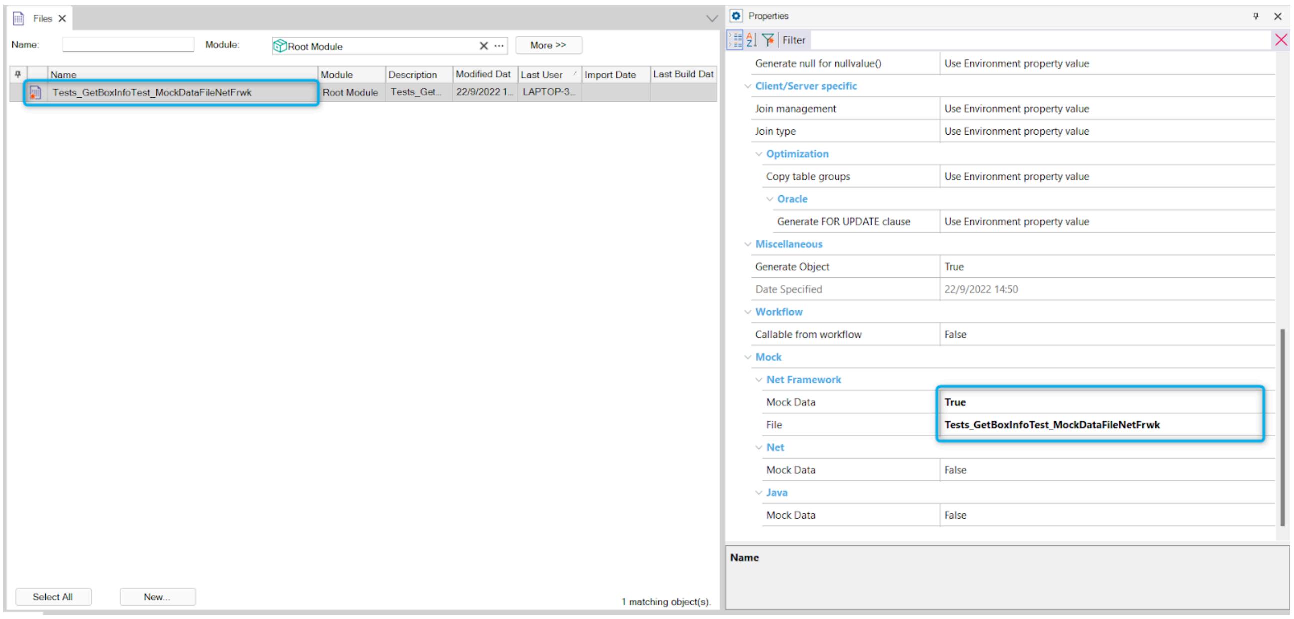
```
GetBoxInfoTestSDT
{
        TestCaseId = '1'
        BoxId = 1
        ExpectedBoxTracking = '1146HFGDS3'
        MsgBoxTracking = '1146HFGDS3'
        ExpectedBoxPurchaseAmount = 180
        MsgBoxPurchaseAmount = '180'
        ExpectedBoxWeight = 2.3
        MsgBoxWeight = '2.3'
        ExpectedBoxStatus = Status.ATDESTINATION
        MsgBoxStatus = 'Status.ATDESTINATION'
        ExpectedBoxOriginArrivalDate = #2018-10-22#
        MsgBoxOriginArrivalDate = '#2018-10-22#'
        ExpectedBoxCost = 48.3
        MsgBoxCost = '48.3'
}
```

3) Save changes, right-click on the Unit Test, and select the option “Record Mocking Data”.



```
/* Autogenerated unit test code for Procedure 'GetBo

For &TestCaseData in GetBoxInfoTestData()

    /* Act...                      */
    &TestCaseData.BoxCost = GetBoxInfo(&TestCaseData

    /* Assert...                   */
    AssertStringEquals(&TestCaseData.ExpectedBoxTrac
    AssertNumericEquals(&TestCaseData.ExpectedBoxPur
    AssertNumericEquals(&TestCaseData.ExpectedBoxWei
    AssertStringEquals(&TestCaseData.ExpectedBoxStat
    AssertStringEquals(&TestCaseData.ExpectedBoxOrig
    AssertNumericEquals(&TestCaseData.ExpectedBoxCos
endfor
```

As you can see, a file is created on the “Files” screen with the name “Tests_GetBoxInfoTest_MockDataFileNetFrwk”. The property “**Mock Data**” in the Unit Test will change to “True”, and the created file will be assigned automatically in File.



Files

Name: Module: Root Module More >>

Name	Module	Description	Modified Dat	Last User	Import Date	Last Build Dat
Tests_GetBoxInfoTest_MockDataFileNetFrwk	Root Module	Tests_Get...	22/9/2022 1...	LAPTOP-3...		

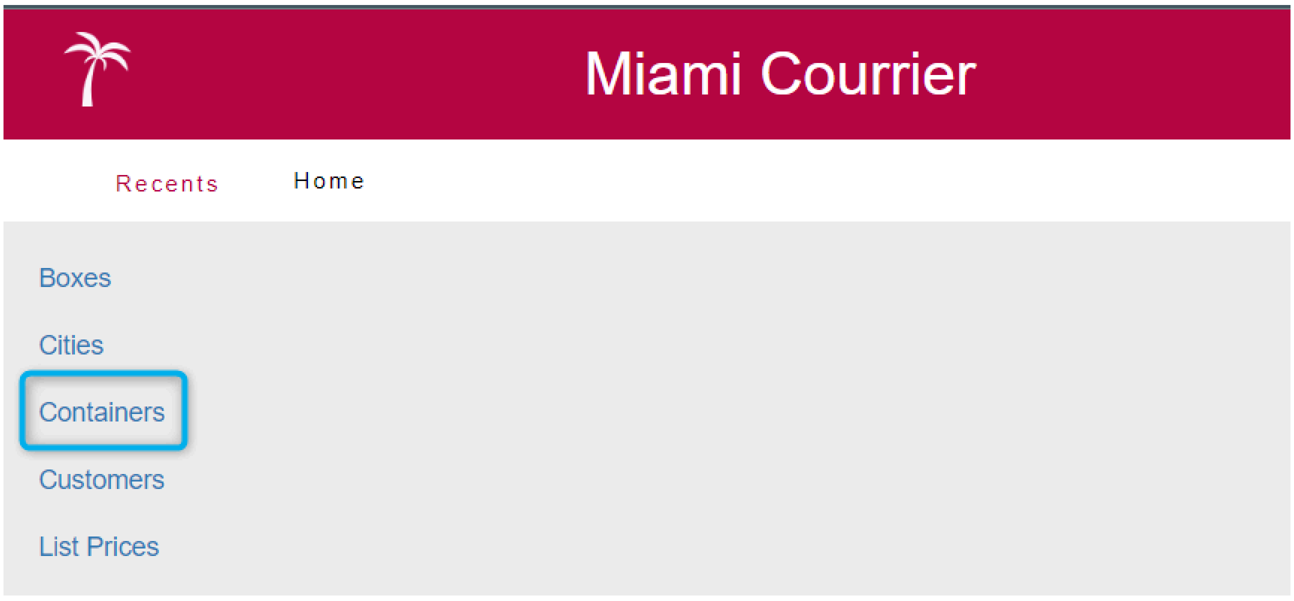
Select All New... 1 matching object(s).

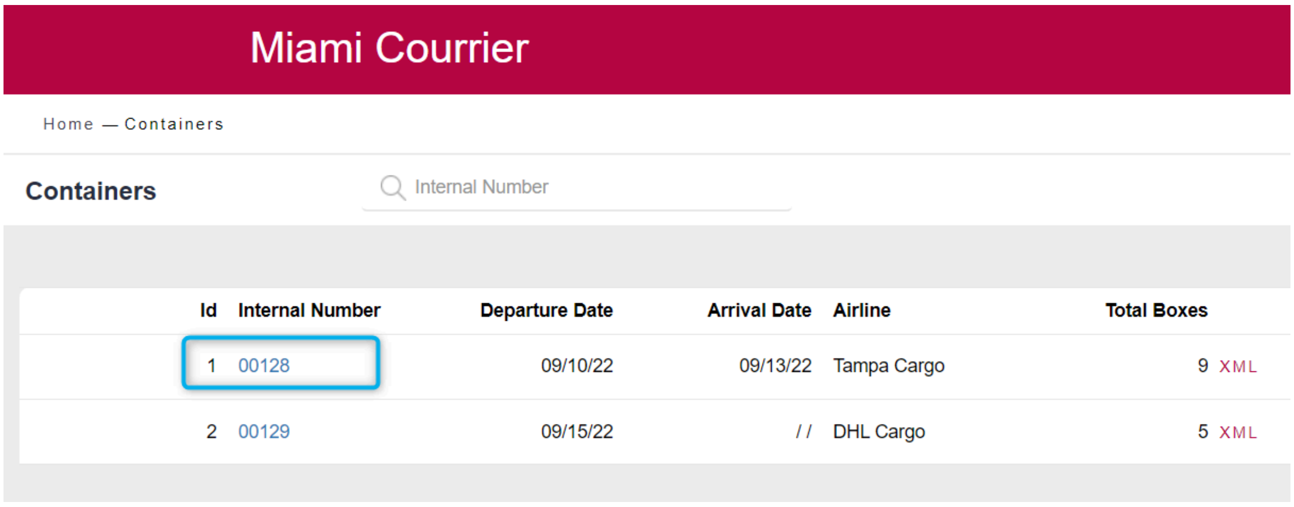
Properties

Generate null for nullvalue()	Use Environment property value
Client/Server specific	
Join management	Use Environment property value
Join type	Use Environment property value
Optimization	
Copy table groups	Use Environment property value
Oracle	
Generate FOR UPDATE clause	Use Environment property value
Miscellaneous	
Generate Object	True
Date Specified	22/9/2022 14:50
Workflow	
Callable from workflow	False
Mock	
Net Framework	
Mock Data	**True**
File	**Tests_GetBoxInfoTest_MockDataFileNetFrwk**
Net	
Mock Data	False
Java	
Mock Data	False

Name

4) Click on “Containers” and, after that, click on the container link “00128” of ID 1 to access the container info.





Miami Courrier

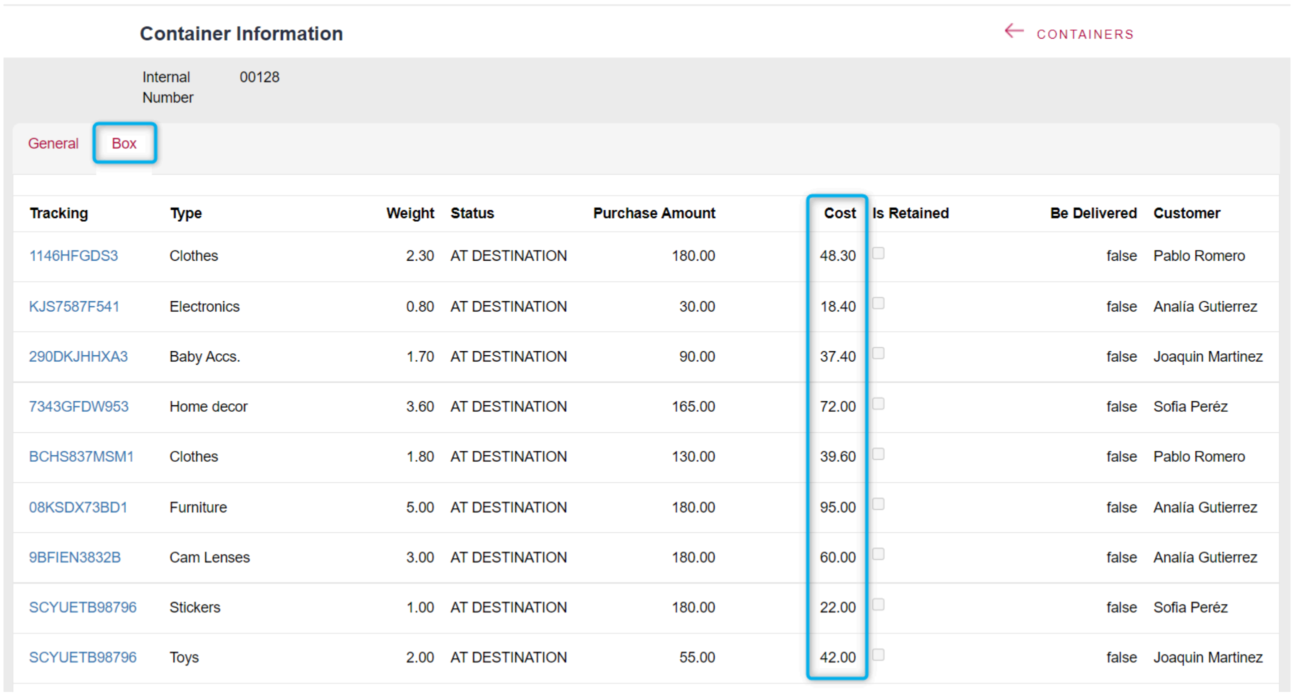
Home — Containers

Containers

Internal Number

Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes	
1	00128	09/10/22	09/13/22	Tampa Cargo	9	XML
2	00129	09/15/22	/ /	DHL Cargo	5	XML

5) Click on the “Box” tab. You will see that, in the “Cost” column, every box states the numbers set by the procedure “SetCostToBoxesInContainer”.



Container Information

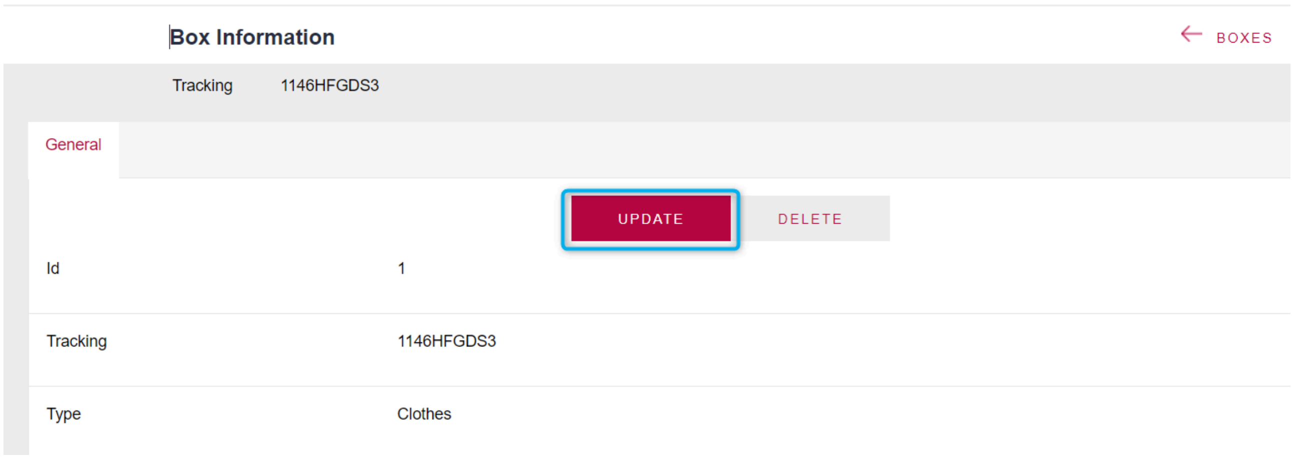
← CONTAINERS

Internal Number 00128

General | Box

Tracking	Type	Weight	Status	Purchase Amount	Cost	Is Retained	Be Delivered	Customer
1146HFGDS3	Clothes	2.30	AT DESTINATION	180.00	48.30	☐	false	Pablo Romero
KJS7587F541	Electronics	0.80	AT DESTINATION	30.00	18.40	☐	false	Analía Gutierrez
290DKJHHXA3	Baby Accs.	1.70	AT DESTINATION	90.00	37.40	☐	false	Joaquin Martinez
7343GFDW953	Home decor	3.60	AT DESTINATION	165.00	72.00	☐	false	Sofia Peréz
BCHS837MSM1	Clothes	1.80	AT DESTINATION	130.00	39.60	☐	false	Pablo Romero
08KSDX73BD1	Furniture	5.00	AT DESTINATION	180.00	95.00	☐	false	Analía Gutierrez
9BFIEN3832B	Cam Lenses	3.00	AT DESTINATION	180.00	60.00	☐	false	Analía Gutierrez
SCYUETB98796	Stickers	1.00	AT DESTINATION	180.00	22.00	☐	false	Sofia Peréz
SCYUETB98796	Toys	2.00	AT DESTINATION	55.00	42.00	☐	false	Joaquin Martinez

6) Select Box 1 and click on “Update”.


Box Information

BOXES

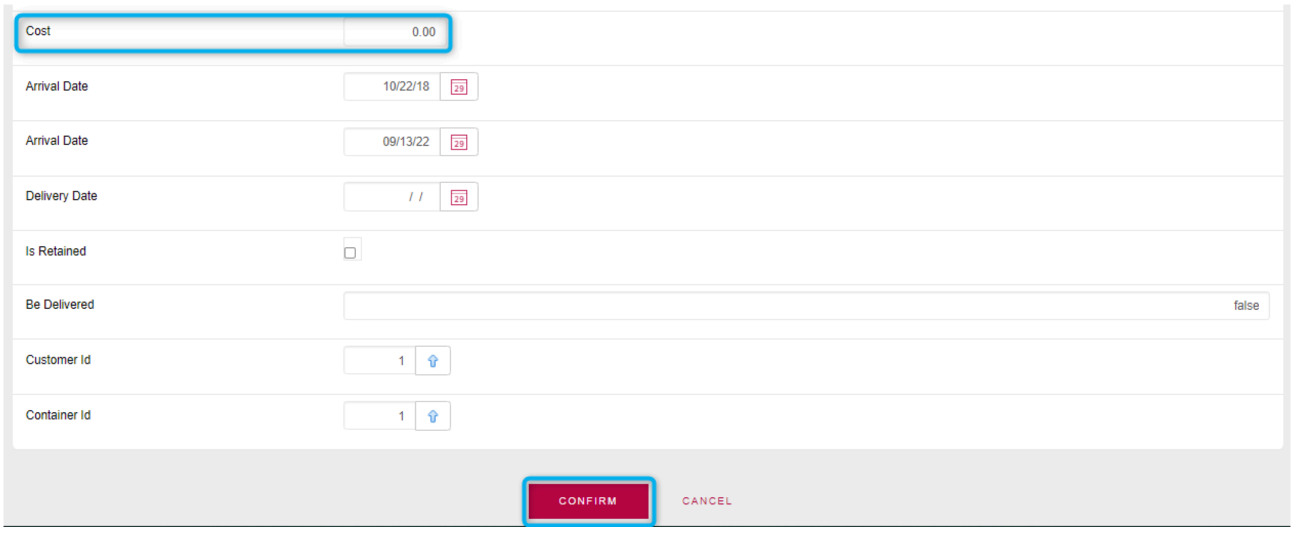
Tracking 1146HFGDS3

General

UPDATE DELETE

Id	1
Tracking	1146HFGDS3
Type	Clothes

7) In the “Cost” cell, change the value 48.30 to 0.00 and click on “Confirm”.

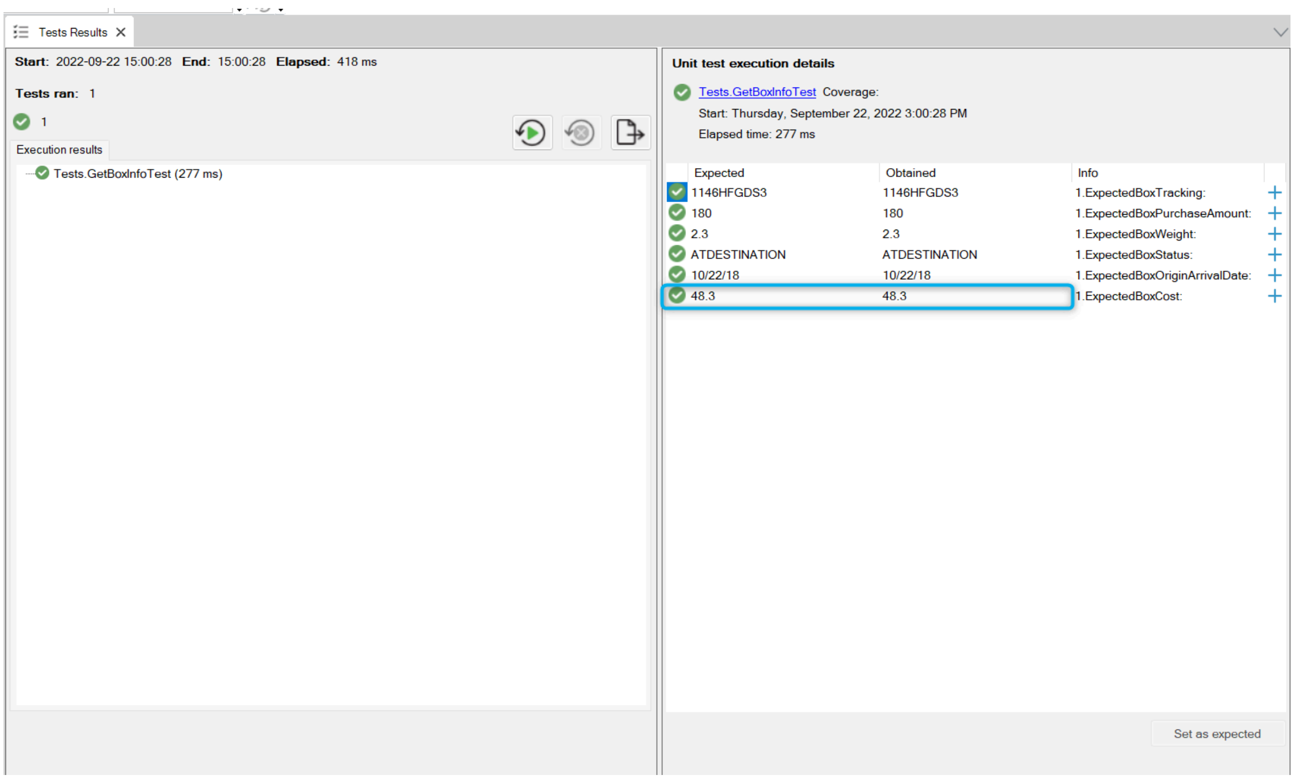


Cost	0.00
Arrival Date	10/22/18
Arrival Date	09/13/22
Delivery Date	/ /
Is Retained	☐
Be Delivered	false
Customer Id	1
Container Id	1

CONFIRM CANCEL

8) Run the test again by clicking on “Run Again”. This time, it will run the test with the recorded mocking data.

When the test execution is finished, the results will be displayed in the Test Results panel. You will see that the cost “48.3” is obtained from the mock file and not from the database that you edited before.


Tests Results

Start: 2022-09-22 15:00:28 **End:** 15:00:28 **Elapsed:** 418 ms

Tests ran: 1

✓ 1

Execution results

✓ Tests.GetBoxInfoTest (277 ms)

Unit test execution details

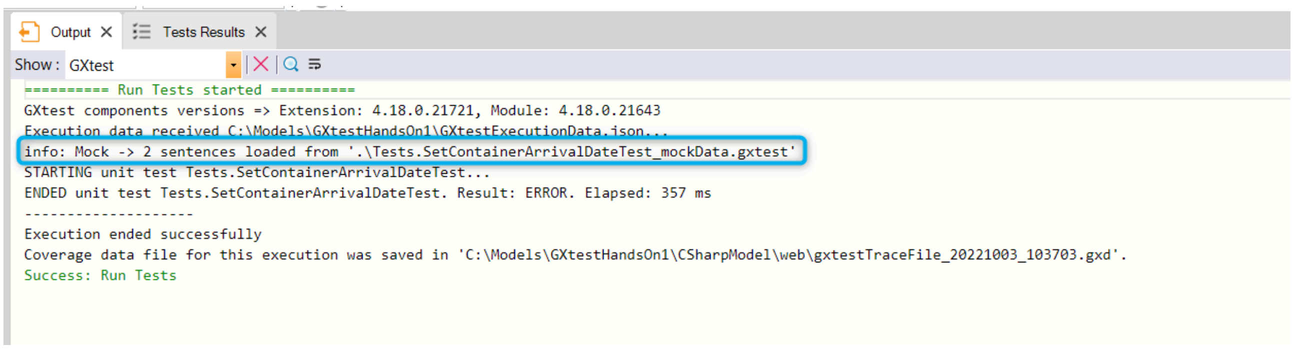
✓ Tests.GetBoxInfoTest Coverage:

Start: Thursday, September 22, 2022 3:00:28 PM

Elapsed time: 277 ms

Expected	Obtained	Info
✓ 1146HFGDS3	1146HFGDS3	1.ExpectedBoxTracking:
✓ 180	180	1.ExpectedBoxPurchaseAmount:
✓ 2.3	2.3	1.ExpectedBoxWeight:
✓ ATDESTINATION	ATDESTINATION	1.ExpectedBoxStatus:
✓ 10/22/18	10/22/18	1.ExpectedBoxOriginArrivalDate:
✓ 48.3	48.3	1.ExpectedBoxCost:

Set as expected


Output | Tests Results

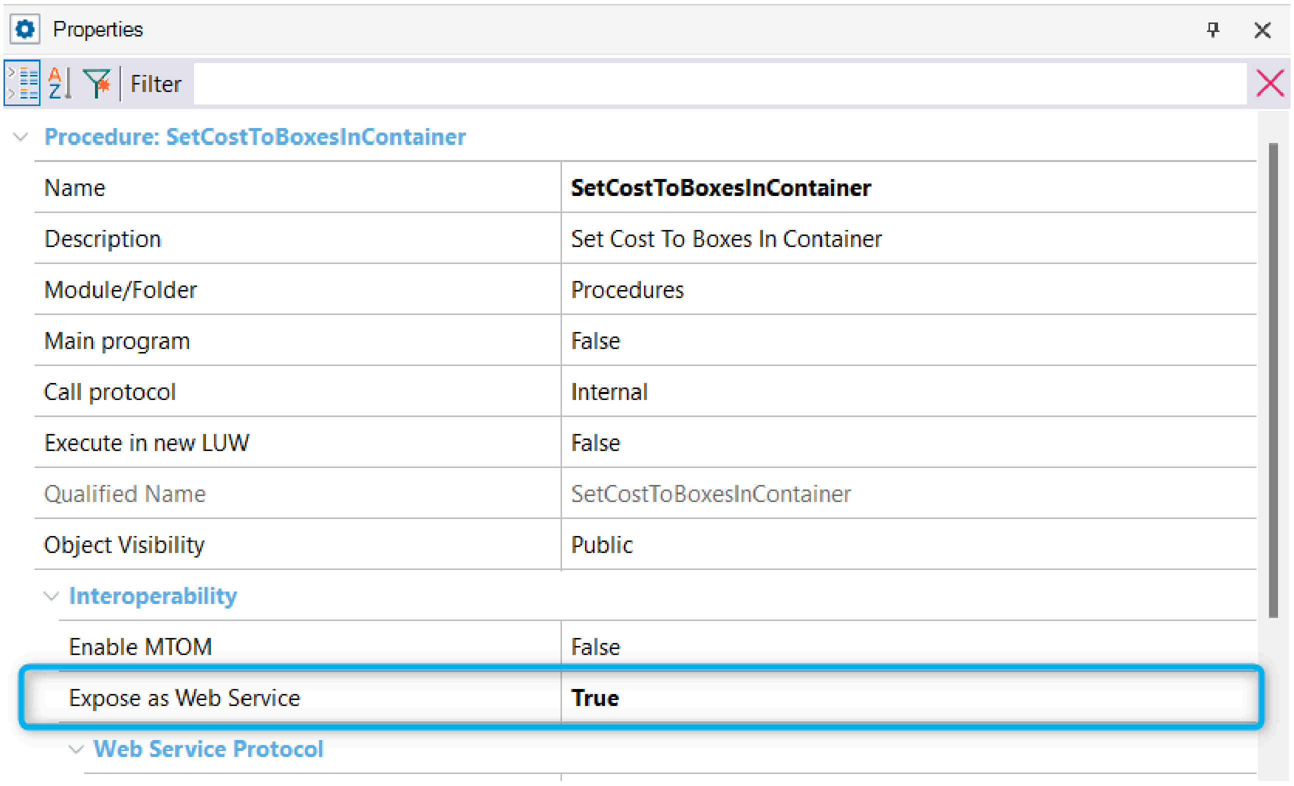
Show : GXtest

```
========== Run Tests started ==========
GXtest components versions => Extension: 4.18.0.21721, Module: 4.18.0.21643
Execution data received C:\Models\GXtestHandsOn1\GXtestExecutionData.json...
info: Mock -> 2 sentences loaded from '.\Tests.SetContainerArrivalDateTest_mockData.gxtest'
STARTING unit test Tests.SetContainerArrivalDateTest...
ENDED unit test Tests.SetContainerArrivalDateTest. Result: ERROR. Elapsed: 357 ms
--------------------
Execution ended successfully
Coverage data file for this execution was saved in 'C:\Models\GXtestHandsOn1\CSharpModel\web\gxtestTraceFile_20221003_103703.gxd'.
Success: Run Tests
```

Rest Tests

It is possible to test procedures portrayed as REST services by consuming them through the HTTP protocol. In this section, you will get a step-by-step how-to on creating a REST services test for the procedure "**SetCostToBoxesInContainer**" with GAM authentication enabled.

1) Open the properties of procedure "SetCostToBoxesInContainer" and change the property "Expose as Web Service" to "True".

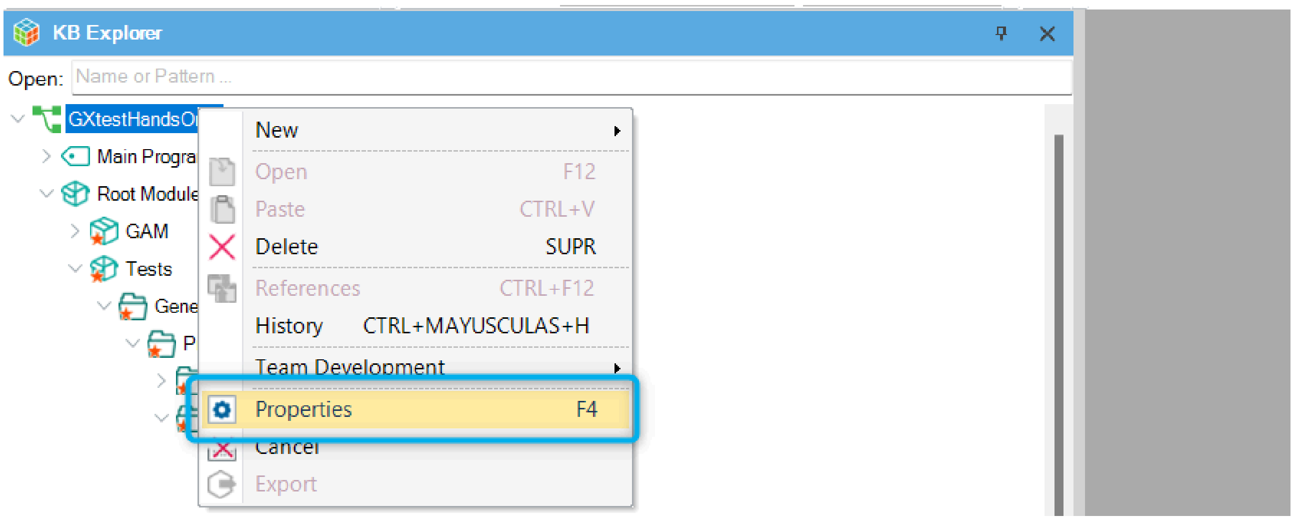


Properties

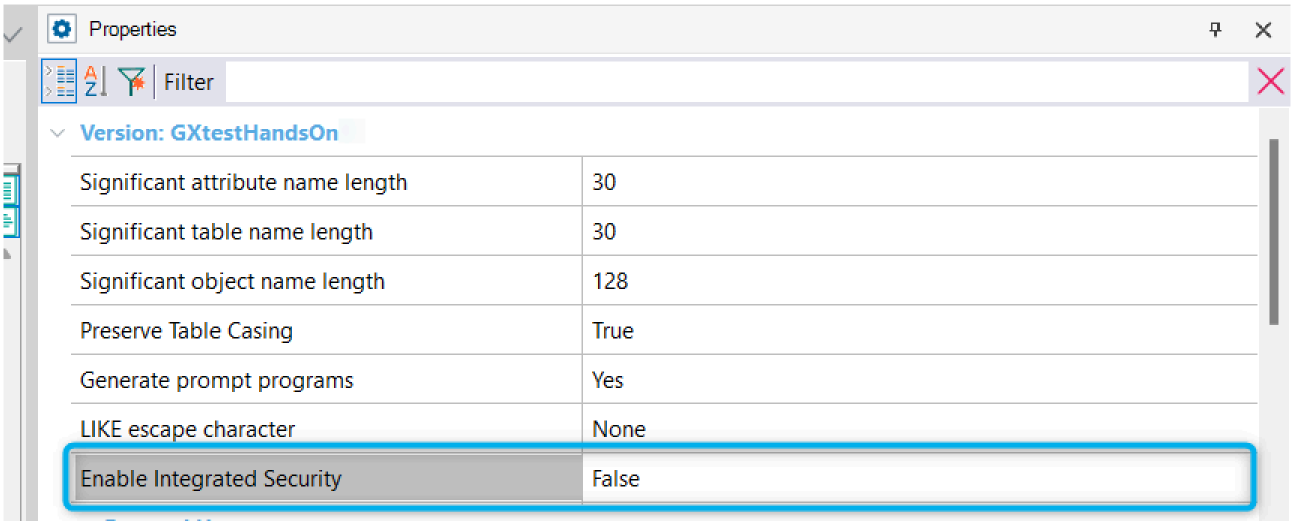
Procedure: SetCostToBoxesInContainer

Name	**SetCostToBoxesInContainer**
Description	Set Cost To Boxes In Container
Module/Folder	Procedures
Main program	False
Call protocol	Internal
Execute in new LUW	False
Qualified Name	SetCostToBoxesInContainer
Object Visibility	Public
Interoperability	
Enable MTOM	False
Expose as Web Service	**True**
Web Service Protocol	

2) Open the KB properties by right-clicking and selecting the option "Properties".

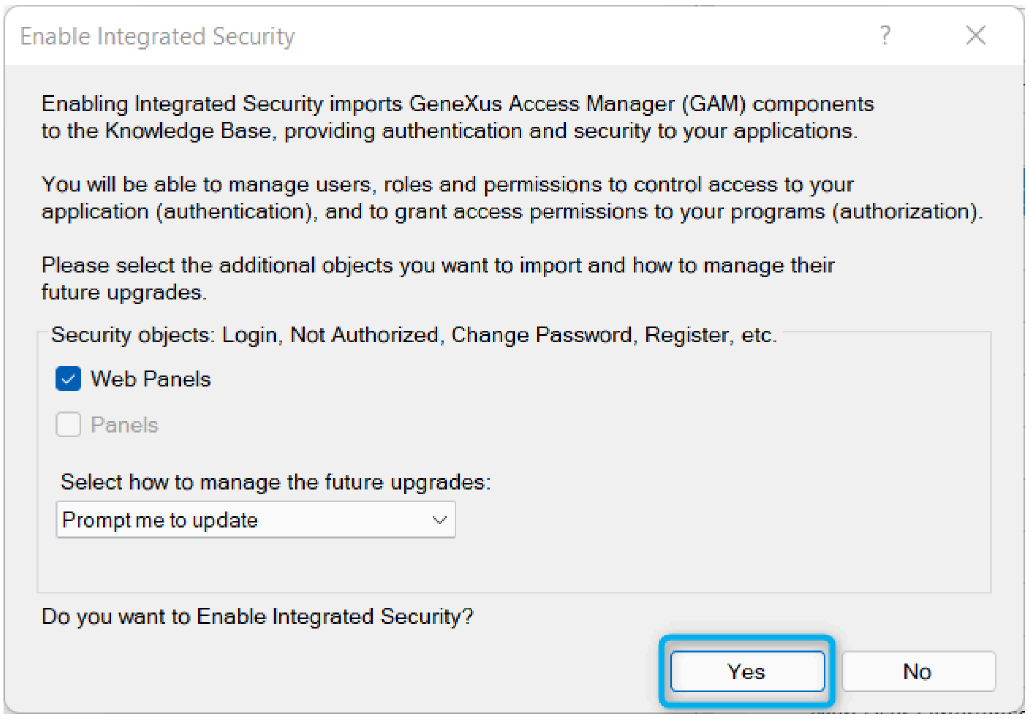


3) Identify the property "Enable Integrated Security" which shows as "False".



Significant attribute name length	30
Significant table name length	30
Significant object name length	128
Preserve Table Casing	True
Generate prompt programs	Yes
LIKE escape character	None
Enable Integrated Security	False

4) Change the property to "True". A pop-up window will request confirmation to import GAM components. Click "Yes".


Enable Integrated Security

Enabling Integrated Security imports GeneXus Access Manager (GAM) components to the Knowledge Base, providing authentication and security to your applications.

You will be able to manage users, roles and permissions to control access to your application (authentication), and to grant access permissions to your programs (authorization).

Please select the additional objects you want to import and how to manage their future upgrades.

Security objects: Login, Not Authorized, Change Password, Register, etc.

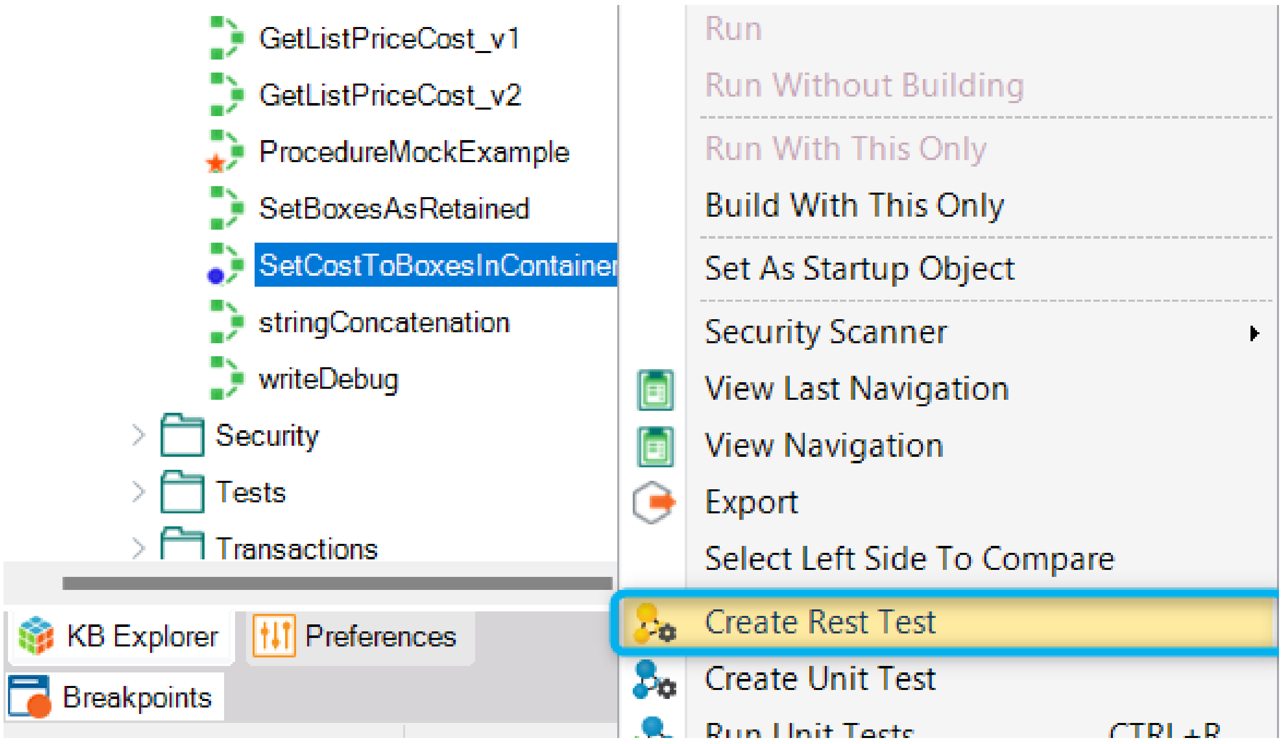
[✓] Web Panels

[ ] Panels

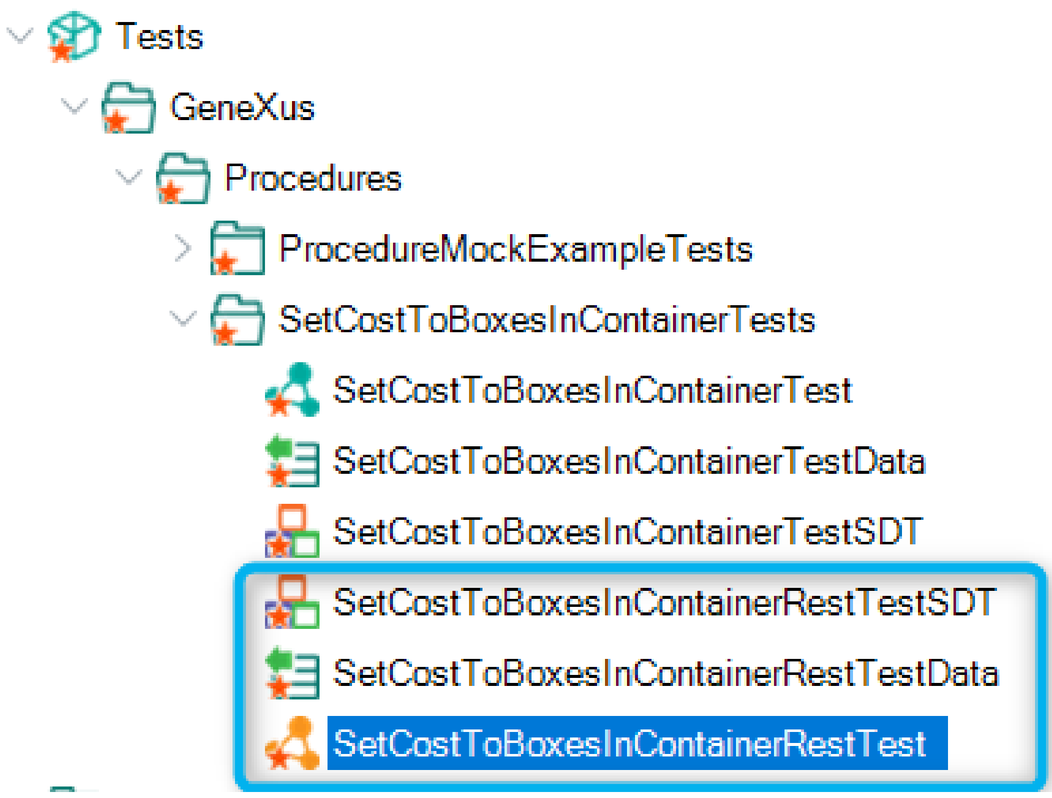
Select how to manage the future upgrades: Prompt me to update

Do you want to Enable Integrated Security?

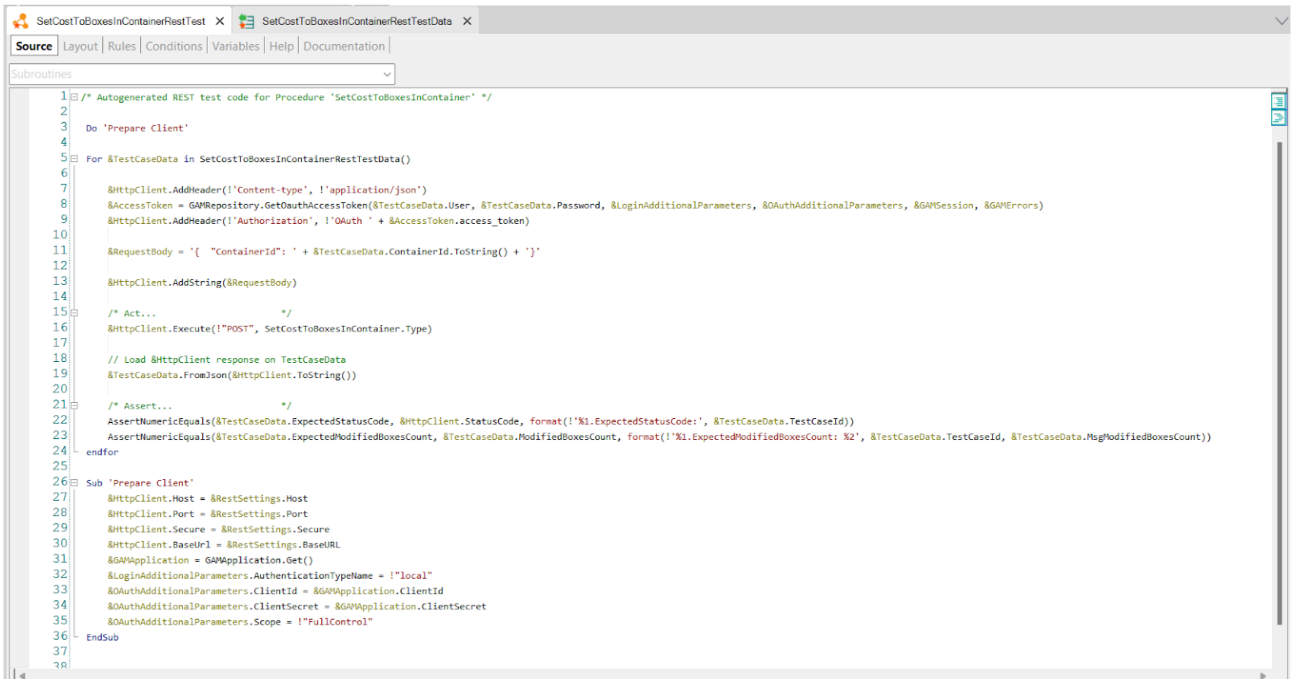
5) Select the procedure "SetCostToBoxesInContainer" in the *KB Explorer* screen, right-click, and choose the option "Create Rest Test".



6) Wait for the test objects “SetCostToBoxesInContainerRestTest”, “SetCostToBoxesInContainerRestTestData”, and “SetCostToBoxesInContainerRestTestSDT” to be created. They will appear in the “Tests” module, inside the folder with the name of the procedure (“SetCostToBoxesInContainerTests”).

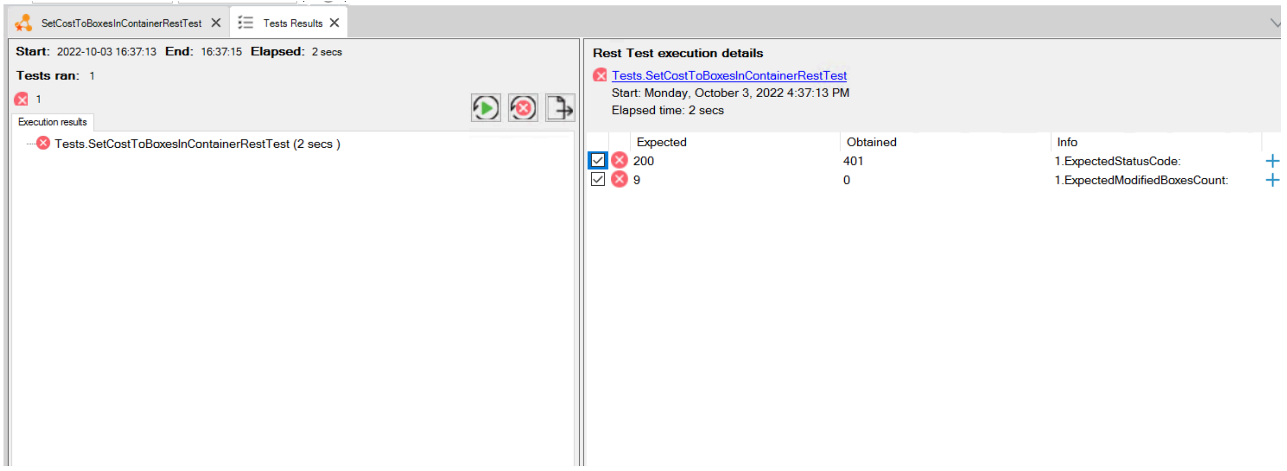

Tests
GeneXus
Procedures
ProcedureMockExampleTests
SetCostToBoxesInContainerTests
SetCostToBoxesInContainerTest
SetCostToBoxesInContainerTestData
SetCostToBoxesInContainerTestSDT
SetCostToBoxesInContainerRestTestSDT
SetCostToBoxesInContainerRestTestData
SetCostToBoxesInContainerRestTest

As you can see, GXtest automatically creates a GeneXus code template iterating over the Data Provider "SetCostToBoxesInContainerRestTestData", which contains the dataset which the test will execute.



```
/* Autogenerated REST test code for Procedure 'SetCostToBoxesInContainer' */

Do 'Prepare Client'

For &TestCaseData in SetCostToBoxesInContainerRestTestData()

    &HttpClient.AddHeader(!'Content-type', !'application/json')
    &AccessToken = GAMRepository.GetOauthAccessToken(&TestCaseData.User, &TestCaseData.Password, &LoginAdditionalParameters, &OAuthAdditionalParameters, &GAMSession, &GAMErrors)
    &HttpClient.AddHeader(!'Authorization', !'OAuth ' + &AccessToken.access_token)

    &RequestBody = '{  "ContainerId": ' + &TestCaseData.ContainerId.ToString() + '}'

    &HttpClient.AddString(&RequestBody)

    /* Act...                    */
    &HttpClient.Execute(!"POST", SetCostToBoxesInContainer.Type)

    // Load &HttpClient response on TestCaseData
    &TestCaseData.FromJson(&HttpClient.ToString())

    /* Assert...                 */
    AssertNumericEquals(&TestCaseData.ExpectedStatusCode, &HttpClient.StatusCode, format(!'%1.ExpectedStatusCode:', &TestCaseData.TestCaseId))
    AssertNumericEquals(&TestCaseData.ExpectedModifiedBoxesCount, &TestCaseData.ModifiedBoxesCount, format(!'%1.ExpectedModifiedBoxesCount: %2', &TestCaseData.TestCaseId, &TestCaseData.MsgModifiedBoxesCount))
endfor

Sub 'Prepare Client'
    &HttpClient.Host = &RestSettings.Host
    &HttpClient.Port = &RestSettings.Port
    &HttpClient.Secure = &RestSettings.Secure
    &HttpClient.BaseUrl = &RestSettings.BaseURL
    &GAMApplication = GAMApplication.Get()
    &LoginAdditionalParameters.AuthenticationTypeName = !"local"
    &OAuthAdditionalParameters.ClientId = &GAMApplication.ClientId
    &OAuthAdditionalParameters.ClientSecret = &GAMApplication.ClientSecret
    &OAuthAdditionalParameters.Scope = !"FullControl"
EndSub
```

7) Run "SetCostToBoxesInContainerRestTest" by right-clicking and selecting "Run Test(s)".



```
Start: 2022-10-03 16:37:13  End: 16:37:15  Elapsed: 2 secs
Tests ran: 1
1
Execution results
Tests.SetCostToBoxesInContainerRestTest (2 secs )

Rest Test execution details
Tests.SetCostToBoxesInContainerRestTest
Start: Monday, October 3, 2022 4:37:13 PM
Elapsed time: 2 secs
```

Expected	Obtained	Info
200	401	1.ExpectedStatusCode:
9	0	1.ExpectedModifiedBoxesCount:

8) Observe the status code obtained for the executed dataset. Since you did not enter the credentials needed for authentication, a 401-error code was obtained and the service could not be consumed.

GAM Authentication

To run a Rest Test with authentication, you must modify the Data Provider values so that the credentials are valid and the code 200 is obtained.

1) Identify the fields to be edited in object "SetCostToBoxesInContainerRestTestData".
2) Enter GAM credentials, save changes, and run the test by selecting the option "Run Test(s)".

The test should pass, displaying the assertion result and the code 200 obtained from the consumed HTTP, verifying that authentication was successful.

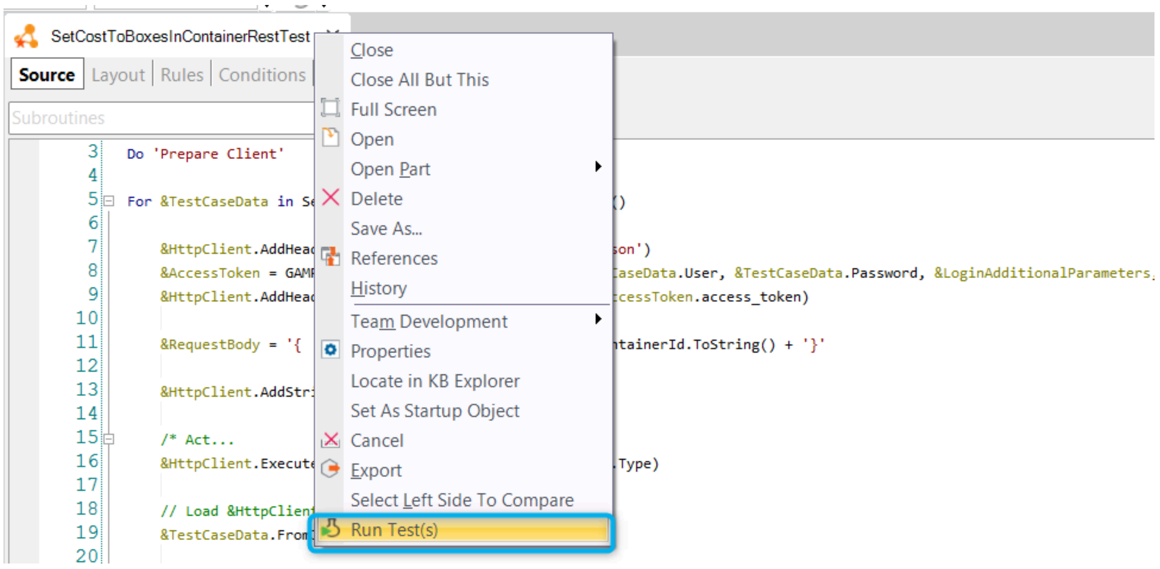
SOLUTION:

1) Enter values ***User = 'admin', Password = 'admin123' , and ExpectedStatusCode = 200*** in the object "SetCostToBoxesInContainerRestTestData".

The Data Provider code is:

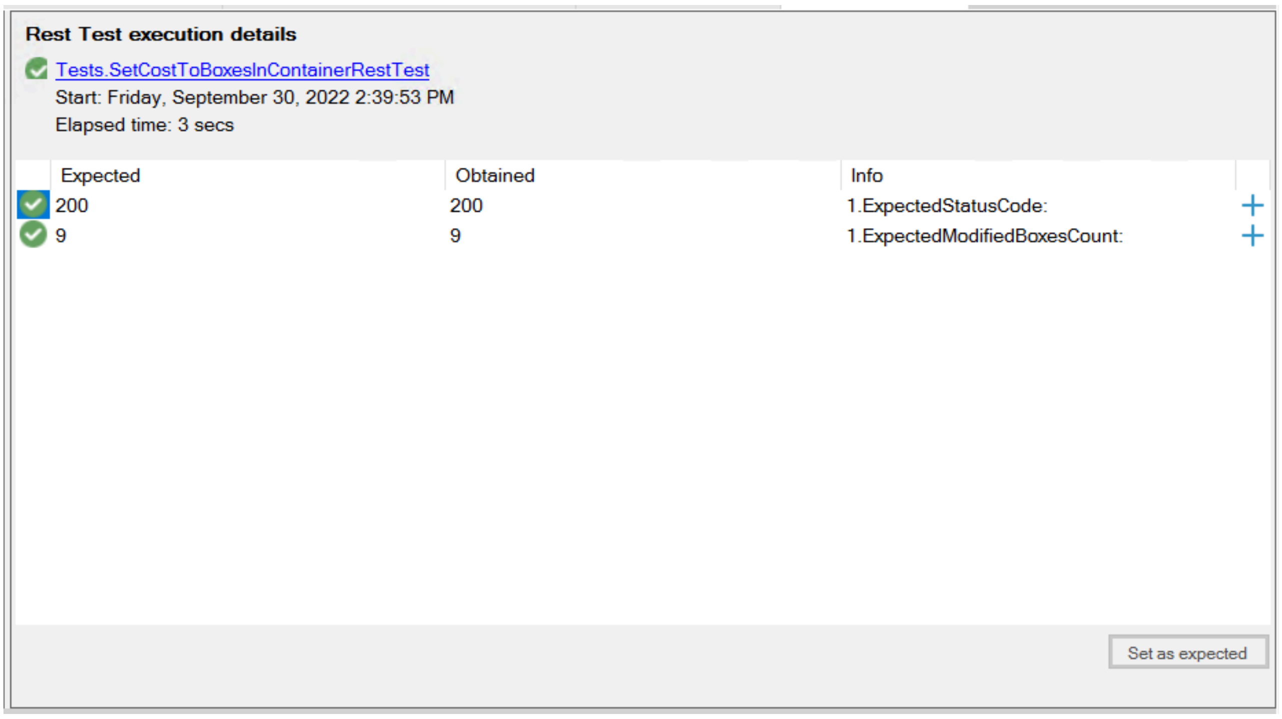
```
SetCostToBoxesInContainerRestTestSDT
{
        TestCaseId = '1'
        ExpectedStatusCode = 200
        User = 'admin'
        Password = 'admin123'
        ContainerId = 1
        ExpectedModifiedBoxesCount = 9
        MsgModifiedBoxesCount = '9'
}
```

2) Save changes, right-click on the Rest Test, and select the option "Run Test(s)".



```
Do 'Prepare Client'

For &TestCaseData in Se                                      ()

    &HttpClient.AddHea                                       son')
    &AccessToken = GAMF                                      aseData.User, &TestCaseData.Password, &LoginAdditionalParameters,
    &HttpClient.AddHea                                       cessToken.access_token)

    &RequestBody = '{                                        tainerId.ToString() + '}'

    &HttpClient.AddStr

    /* Act...
    &HttpClient.Execute                                      .Type)

    // Load &HttpClien
    &TestCaseData.From
```

9) When the test execution is finished, the results will be displayed in the Test Results screen. You will see the test passes since authentication was successful, the expected status code was obtained, and the number of modified boxes was also the expected one.



Rest Test execution details

✓ Tests.SetCostToBoxesInContainerRestTest
Start: Friday, September 30, 2022 2:39:53 PM
Elapsed time: 3 secs

	Expected	Obtained	Info
✓	200	200	1.ExpectedStatusCode:
✓	9	9	1.ExpectedModifiedBoxesCount:

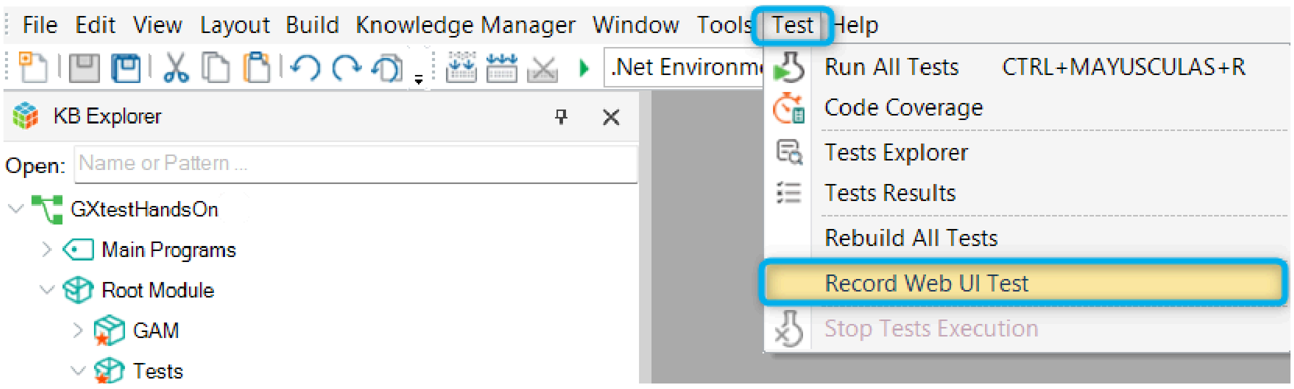
UI Tests

In this section, you will be guided through running a user interface test (from here on, "UI Test") in the web application in which you have been working.

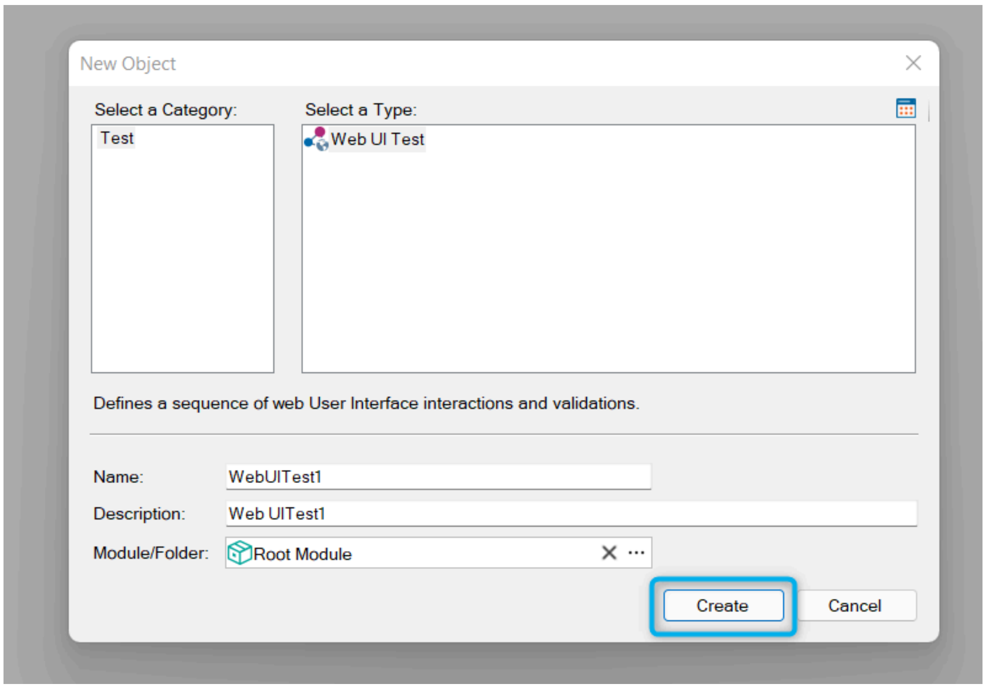
You will record the following sequence: creating a box belonging to container 1, setting the cost of container 1, validating that the cost was assigned, and eliminating the box.

Sequence Recording with GXtest Recorder

1) Go to the "Test" tab and select the option "Record Web UI Test".



2) A window will open to confirm the creation of the new object, "Web UI Test". Click on "Create".

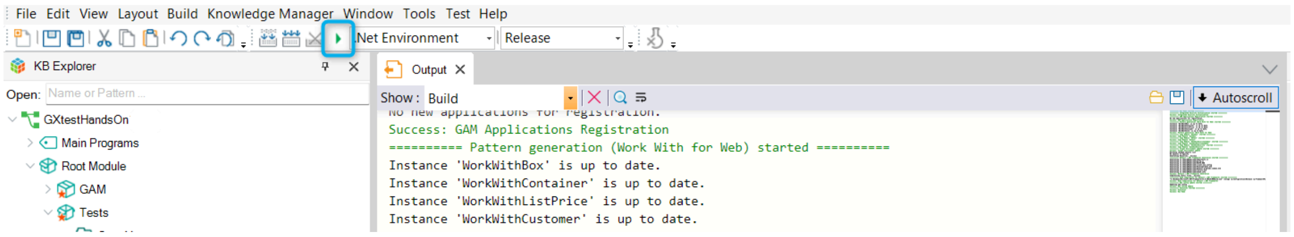


Defines a sequence of web User Interface interactions and validations.

Name:	WebUITest1
Description:	Web UITest1
Module/Folder:	Root Module

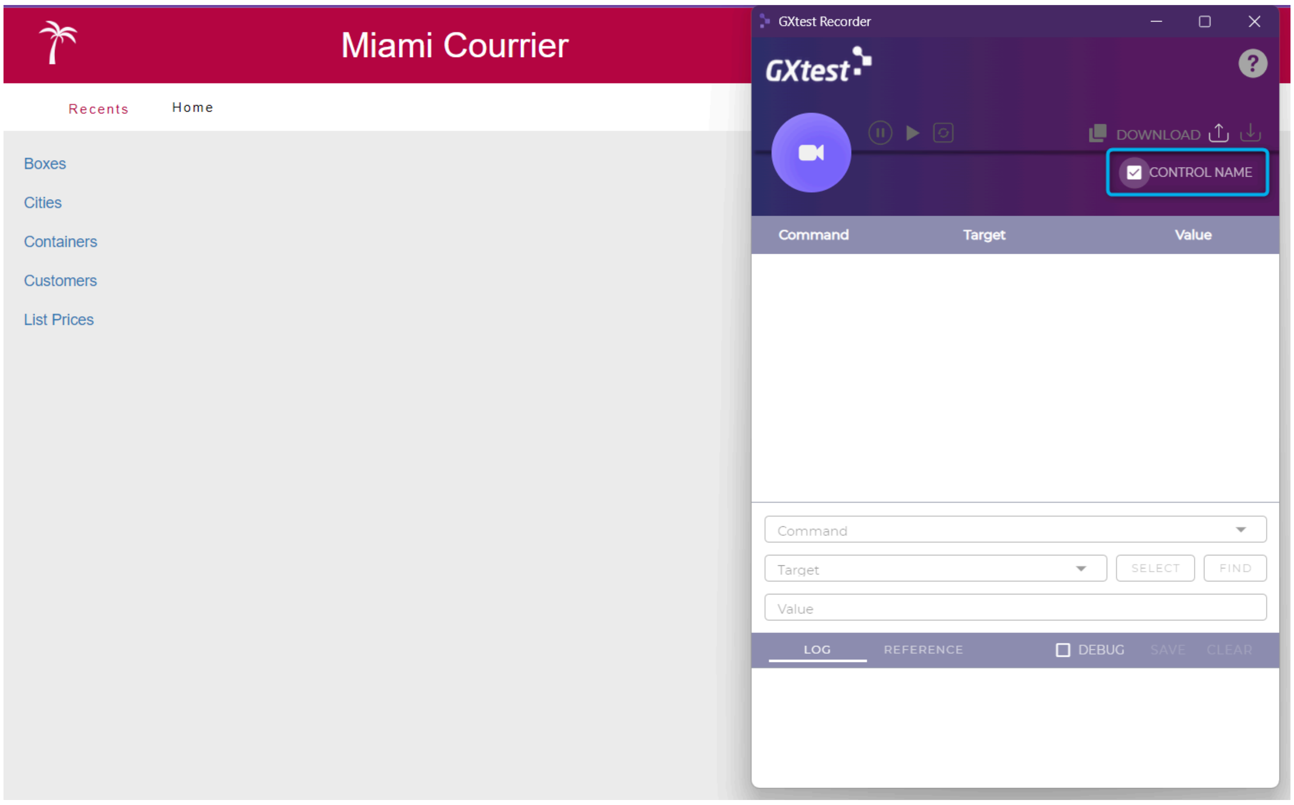
Your browser will open, and the GXtest Recorder extension will be ready to begin recording.

3) Click on “Play” and wait while it runs the application again on the Chrome browser.



```
No new applications for registration.
Success: GAM Applications Registration
========== Pattern generation (Work With for Web) started ==========
Instance 'WorkWithBox' is up to date.
Instance 'WorkWithContainer' is up to date.
Instance 'WorkWithListPrice' is up to date.
Instance 'WorkWithCustomer' is up to date.
```

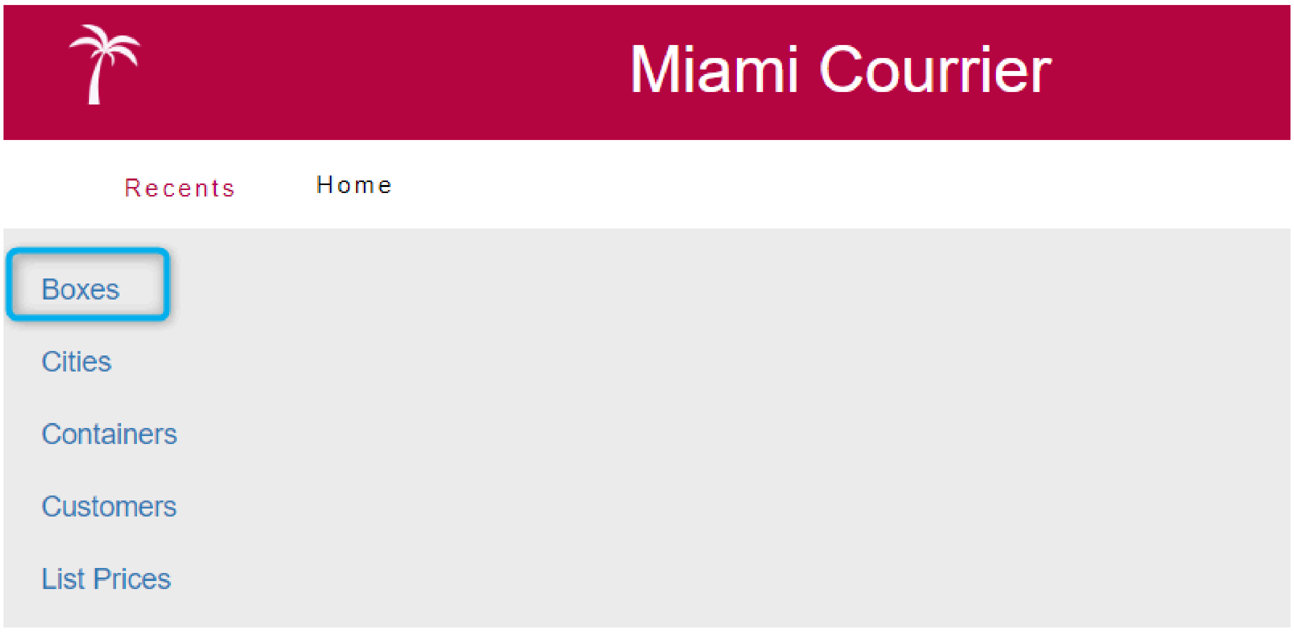
4) Before you begin recording, check the box “CONTROL NAME” so that the recorder uses these selectors if available.



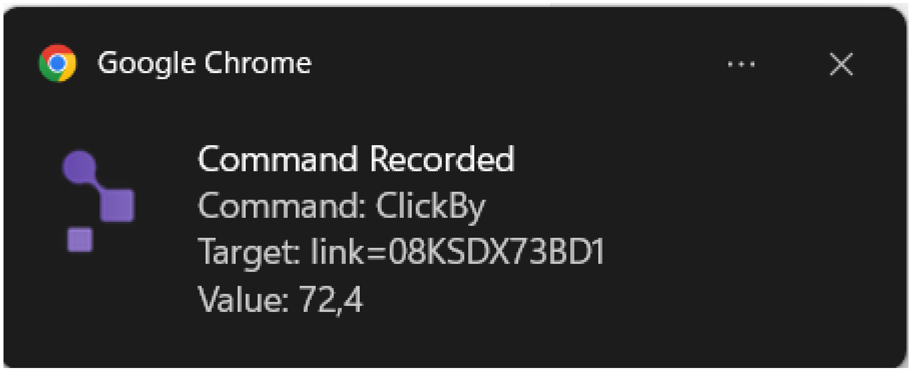
Let's begin the sequence recording!

1) Enter User = **admin** and Password = **admin123**. Click on “SIGN IN” button. Application login will be recorded.

2) Click on the “Boxes” option in the main menu.

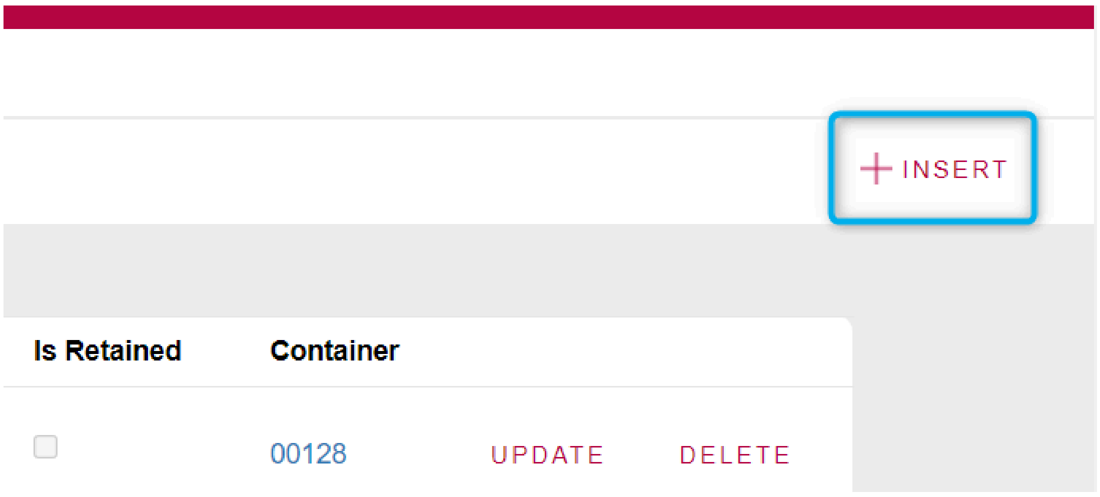


You will be able to verify that the sequence is being recorded by looking at the notifications which will appear at the bottom-right corner of the screen. For each recorded action, a message like this one is displayed:

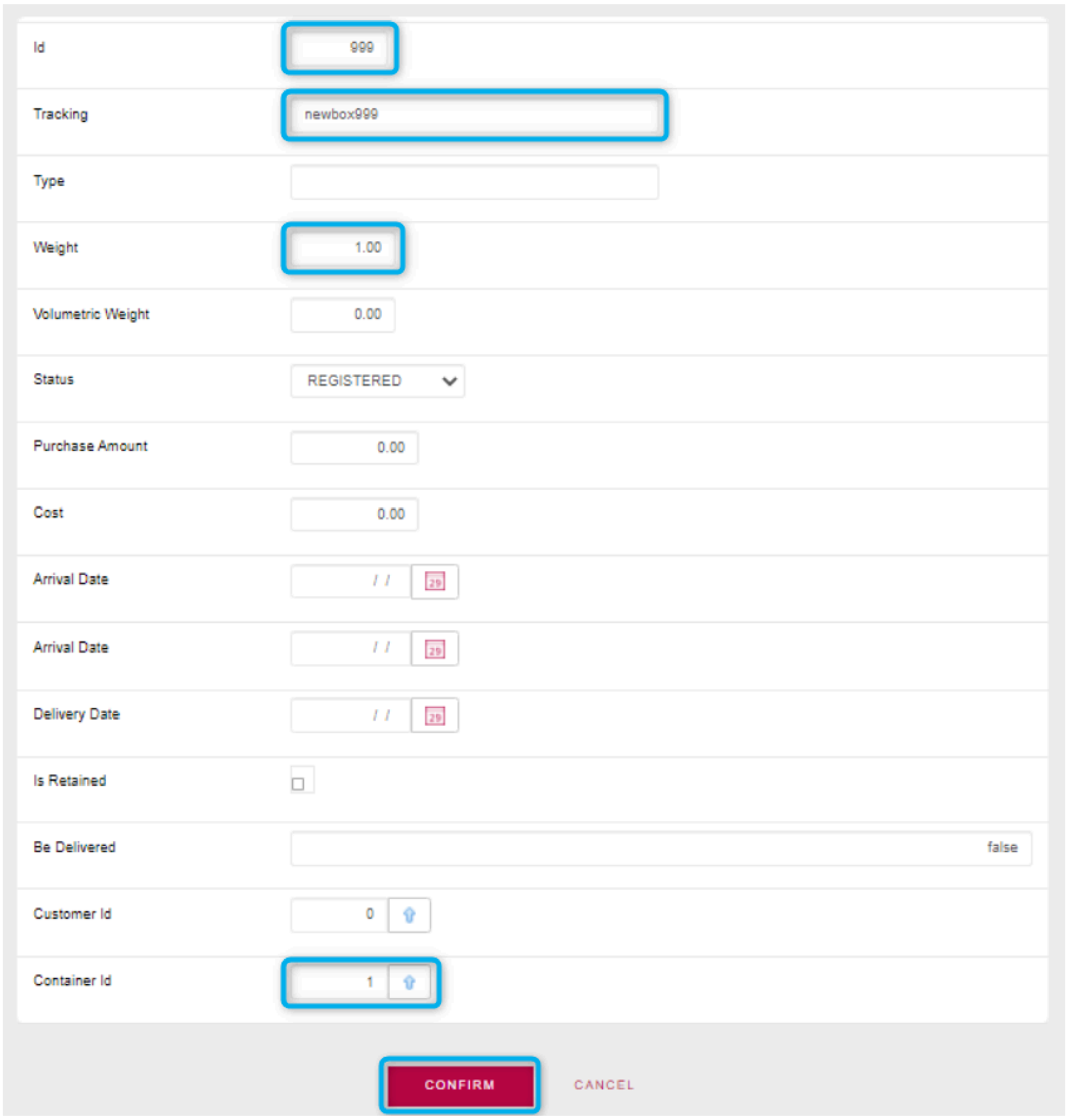


Command Recorded
Command: ClickBy
Target: link=08KSDX73BD1
Value: 72,4

Click on “INSERT” to create a new box.

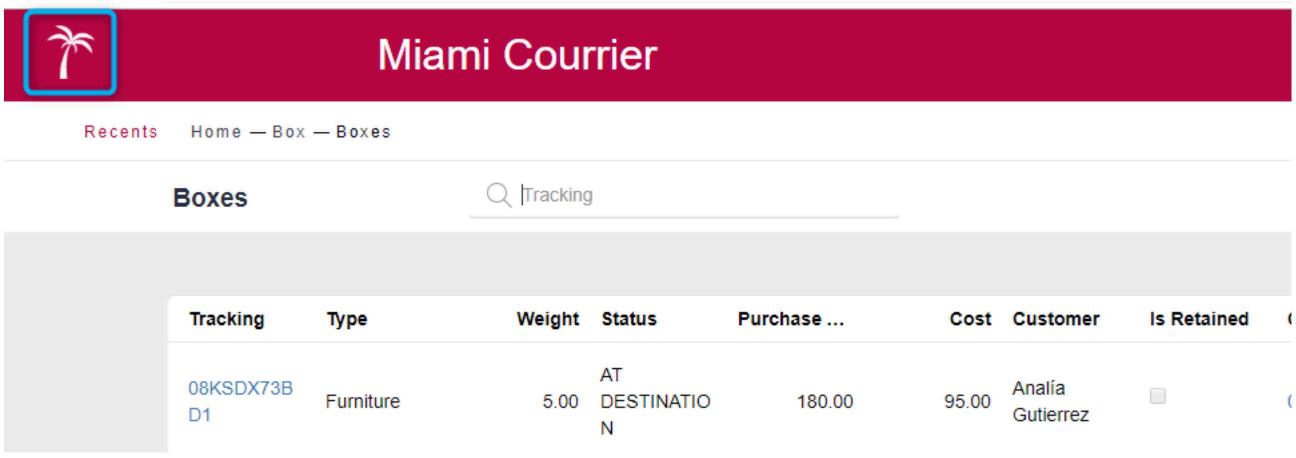


Only fill in the fields **Id**=999, **Tracking**= NEWBOX999, **Weight**= 1, and **Container Id**= 1 of this box and click on "CONFIRM".



Id	999
Tracking	newbox999
Type	
Weight	1.00
Volumetric Weight	0.00
Status	REGISTERED
Purchase Amount	0.00
Cost	0.00
Arrival Date	/ /
Arrival Date	/ /
Delivery Date	/ /
Is Retained	☐
Be Delivered	false
Customer Id	0
Container Id	1

Click on the palm tree to get back to the homepage.



Miami Courrier

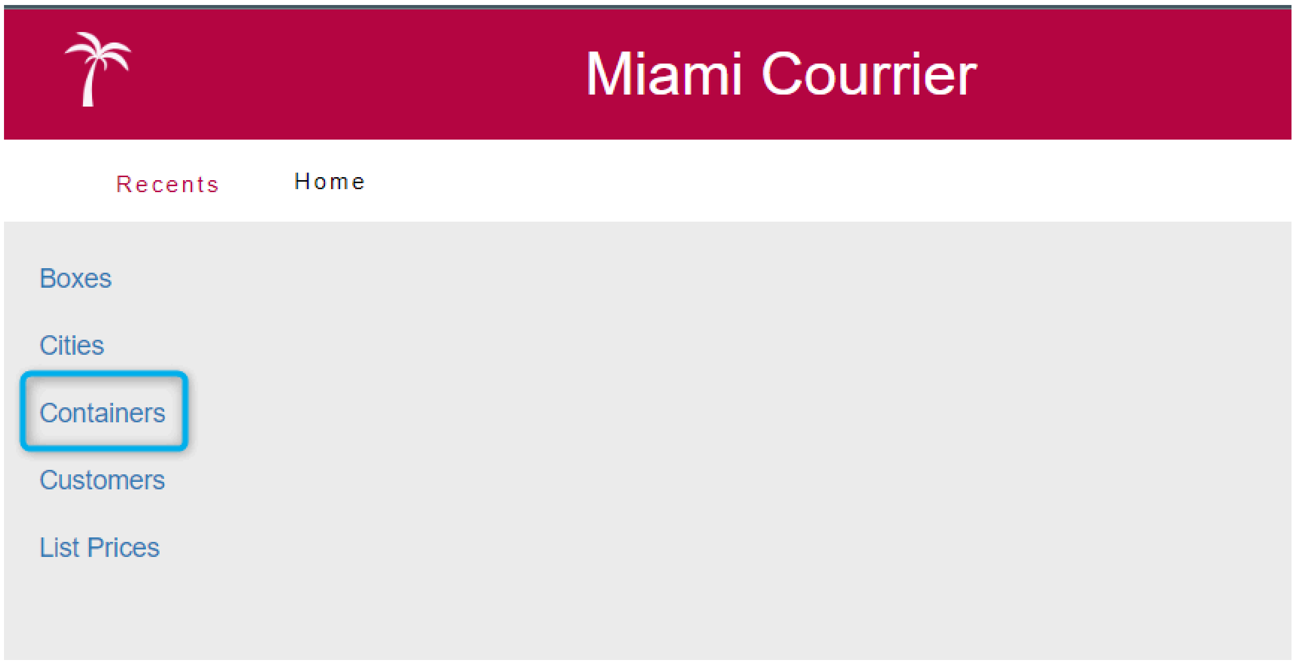
Recents Home — Box — Boxes

Boxes

Tracking

Tracking	Type	Weight	Status	Purchase ...	Cost	Customer	Is Retained
08KSDX73B D1	Furniture	5.00	AT DESTINATIO N	180.00	95.00	Analía Gutierrez	☐

Select the option "Containers" in the menu.



Miami Courrier

Recents Home

Boxes

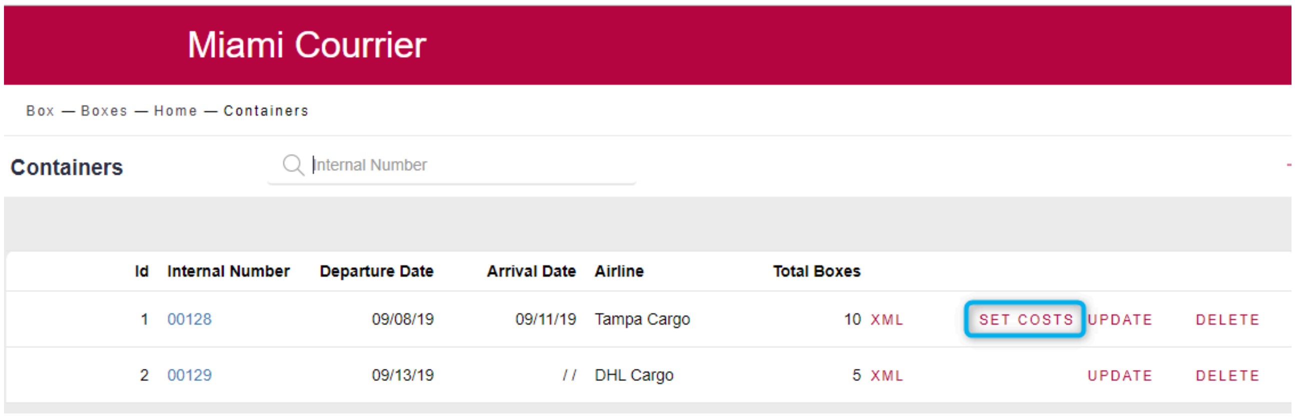
Cities

Containers

Customers

List Prices

Click on “SET COSTS”.



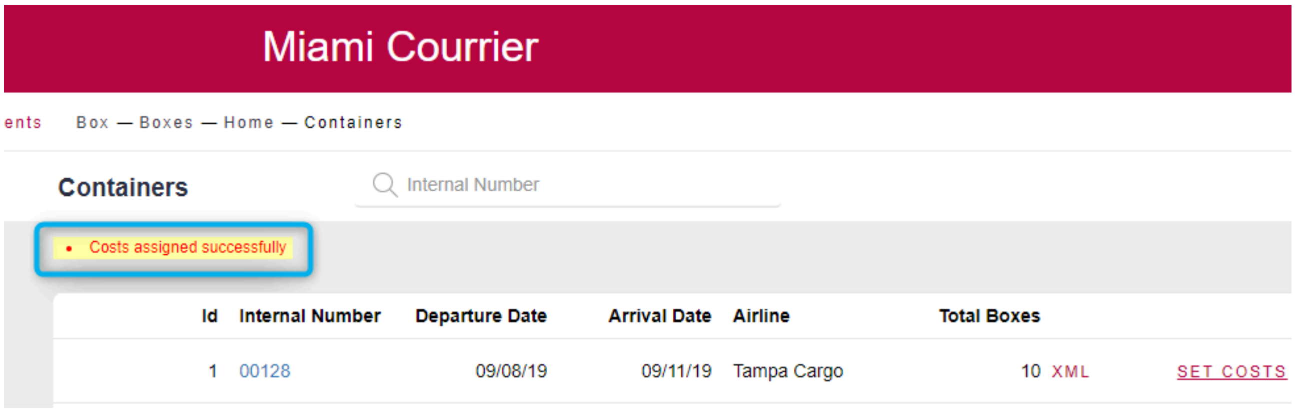
Miami Courrier

Box — Boxes — Home — Containers

Containers

Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes				
1	00128	09/08/19	09/11/19	Tampa Cargo	10	XML	SET COSTS	UPDATE	DELETE
2	00129	09/13/19	/ /	DHL Cargo	5	XML		UPDATE	DELETE

The message “Costs assigned successfully” will be displayed.



Miami Courrier

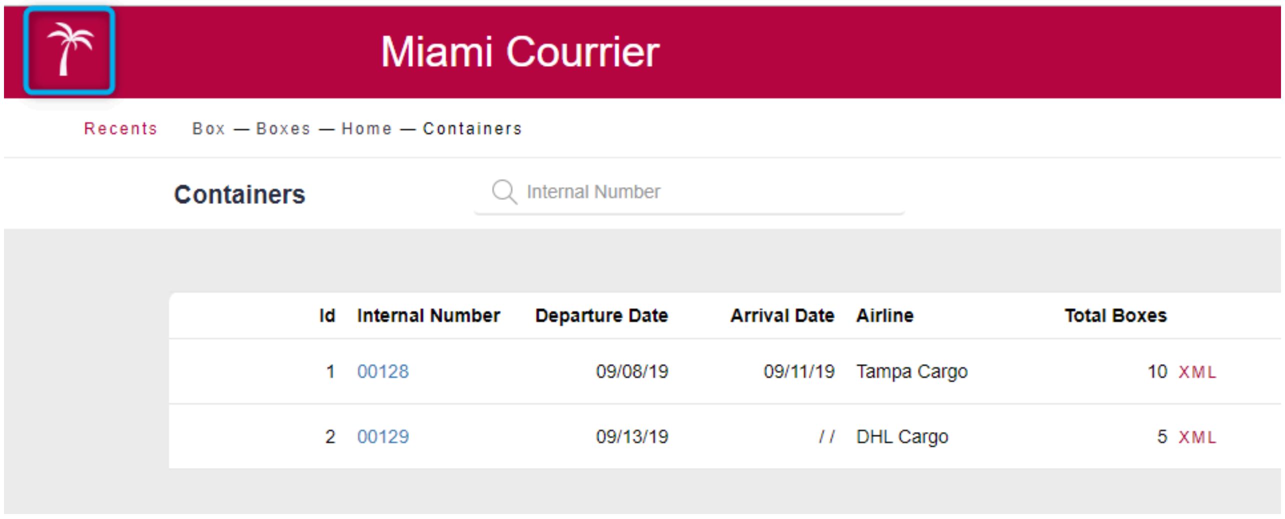
ents Box — Boxes — Home — Containers

Containers

• Costs assigned successfully

Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes		
1	00128	09/08/19	09/11/19	Tampa Cargo	10	XML	SET COSTS

Click on the palm tree to return to the homepage again.



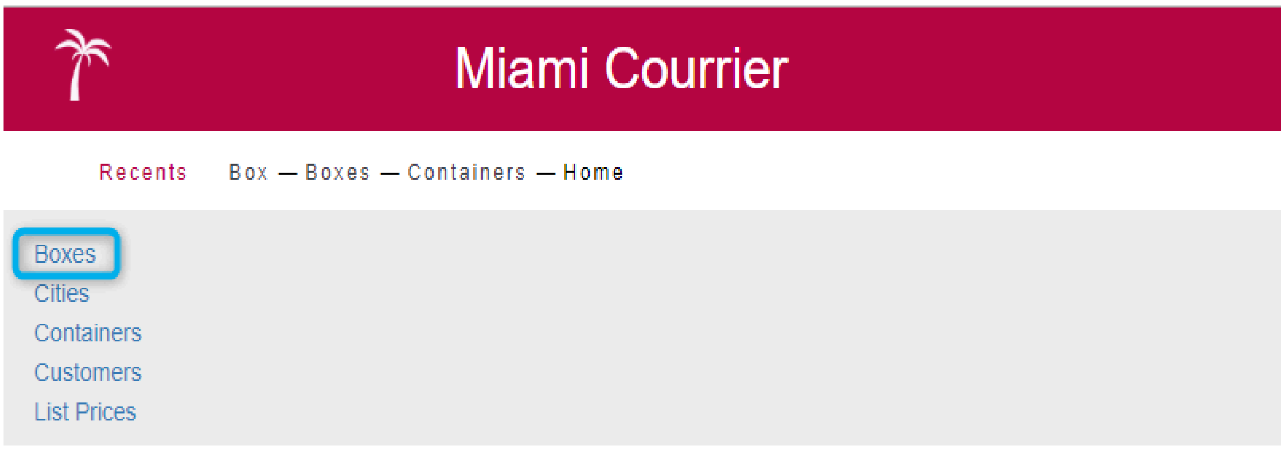
Miami Courrier

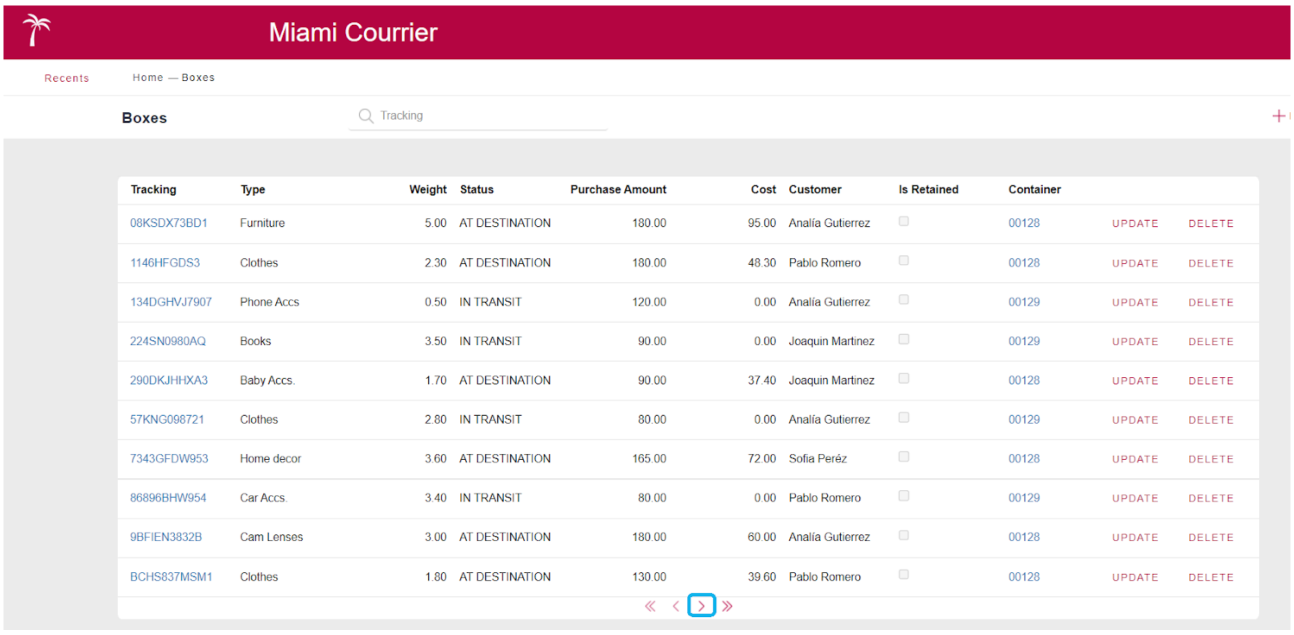
Recents Box — Boxes — Home — Containers

Containers

Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes	
1	00128	09/08/19	09/11/19	Tampa Cargo	10	XML
2	00129	09/13/19	/ /	DHL Cargo	5	XML

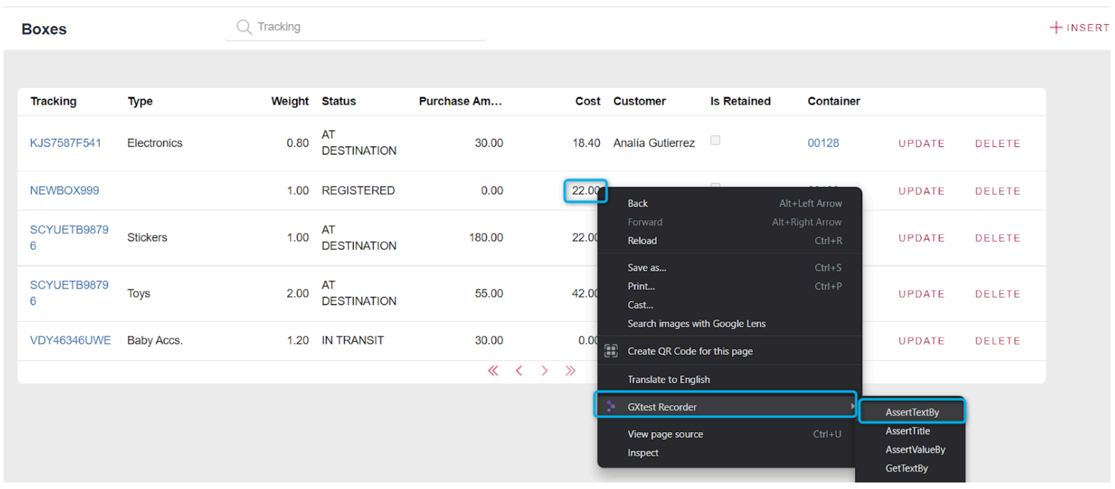
Click on “Boxes” in the main menu.





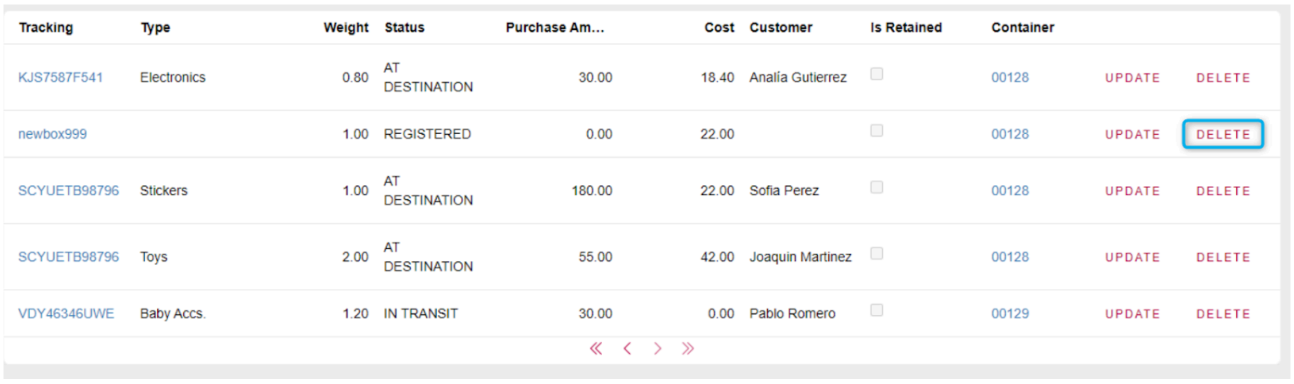
Tracking	Type	Weight	Status	Purchase Amount	Cost	Customer	Is Retained	Container		
08KSDX73BD1	Furniture	5.00	AT DESTINATION	180.00	95.00	Analía Gutierrez	☐	00128	UPDATE	DELETE
1146HFGDS3	Clothes	2.30	AT DESTINATION	180.00	48.30	Pablo Romero	☐	00128	UPDATE	DELETE
134DGHVJ7907	Phone Accs	0.50	IN TRANSIT	120.00	0.00	Analía Gutierrez	☐	00129	UPDATE	DELETE
224SN0980AQ	Books	3.50	IN TRANSIT	90.00	0.00	Joaquin Martinez	☐	00129	UPDATE	DELETE
290DKJHHXA3	Baby Accs.	1.70	AT DESTINATION	90.00	37.40	Joaquin Martinez	☐	00128	UPDATE	DELETE
57KNG098721	Clothes	2.80	IN TRANSIT	80.00	0.00	Analía Gutierrez	☐	00129	UPDATE	DELETE
7343GFDW953	Home decor	3.60	AT DESTINATION	165.00	72.00	Sofia Peréz	☐	00128	UPDATE	DELETE
86896BHW954	Car Accs.	3.40	IN TRANSIT	80.00	0.00	Pablo Romero	☐	00129	UPDATE	DELETE
9BFIEN3832B	Cam Lenses	3.00	AT DESTINATION	180.00	60.00	Analía Gutierrez	☐	00128	UPDATE	DELETE
BCHS837MSM1	Clothes	1.80	AT DESTINATION	130.00	39.60	Pablo Romero	☐	00128	UPDATE	DELETE

Right-click on the value 22.00 to be validated (in this case, the cost of box “newbox999”). Inside the option “GXtest Recorder”, select “AssertTextBy”.

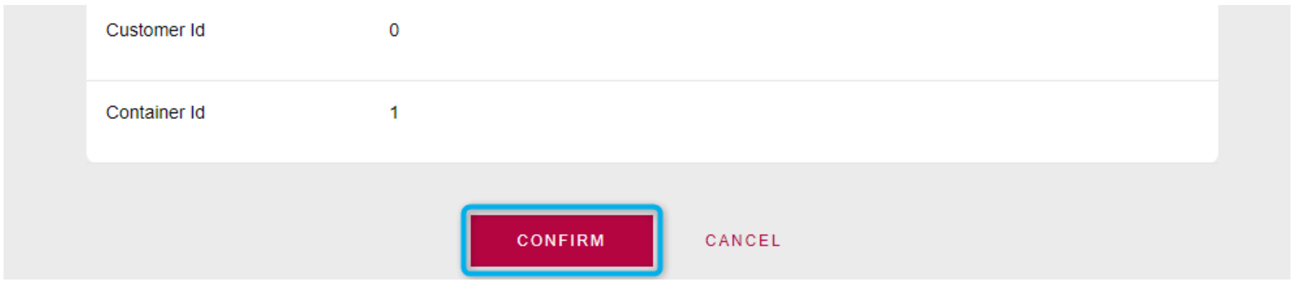


Tracking	Type	Weight	Status	Purchase Am...	Cost	Customer	Is Retained	Container		
KJS7587F541	Electronics	0.80	AT DESTINATION	30.00	18.40	Analía Gutierrez	☐	00128	UPDATE	DELETE
NEWBOX999		1.00	REGISTERED	0.00	22.00				UPDATE	DELETE
SCYUETB98796	Stickers	1.00	AT DESTINATION	180.00	22.00				UPDATE	DELETE
SCYUETB98796	Toys	2.00	AT DESTINATION	55.00	42.00				UPDATE	DELETE
VDY46346UWE	Baby Accs.	1.20	IN TRANSIT	30.00	0.00				UPDATE	DELETE

Click on “DELETE” and then click on "CONFIRM”.

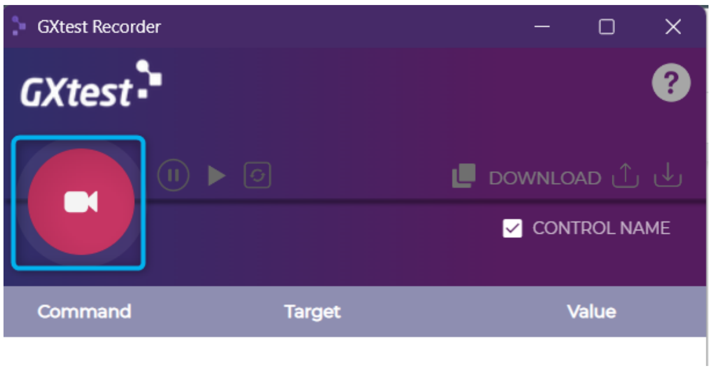


Tracking	Type	Weight	Status	Purchase Am...	Cost	Customer	Is Retained	Container		
KJS7587F541	Electronics	0.80	AT DESTINATION	30.00	18.40	Analía Gutierrez	☐	00128	UPDATE	DELETE
newbox999		1.00	REGISTERED	0.00	22.00		☐	00128	UPDATE	DELETE
SCYUETB98796	Stickers	1.00	AT DESTINATION	180.00	22.00	Sofia Perez	☐	00128	UPDATE	DELETE
SCYUETB98796	Toys	2.00	AT DESTINATION	55.00	42.00	Joaquin Martinez	☐	00128	UPDATE	DELETE
VDY46346UWE	Baby Accs.	1.20	IN TRANSIT	30.00	0.00	Pablo Romero	☐	00129	UPDATE	DELETE



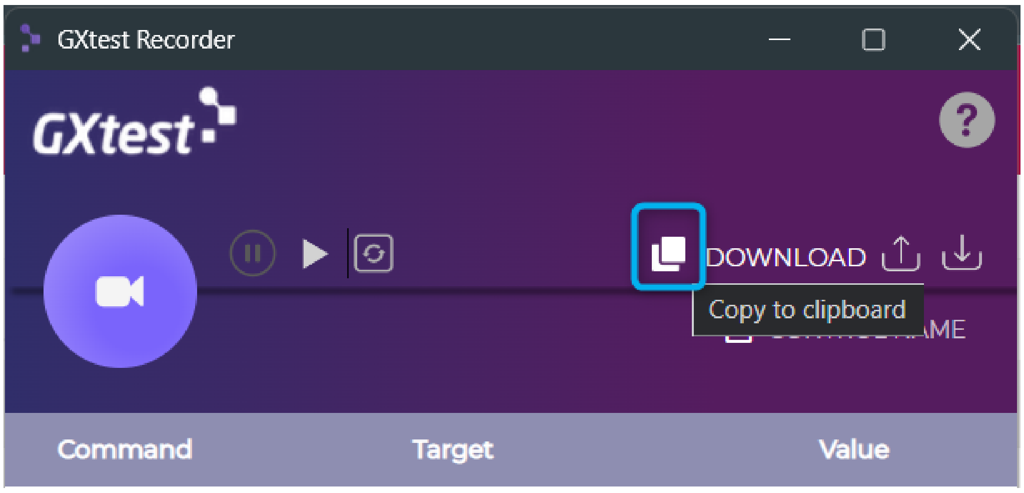
Customer Id	0
Container Id	1

Stop the recording by pressing the button “Stop recording”.

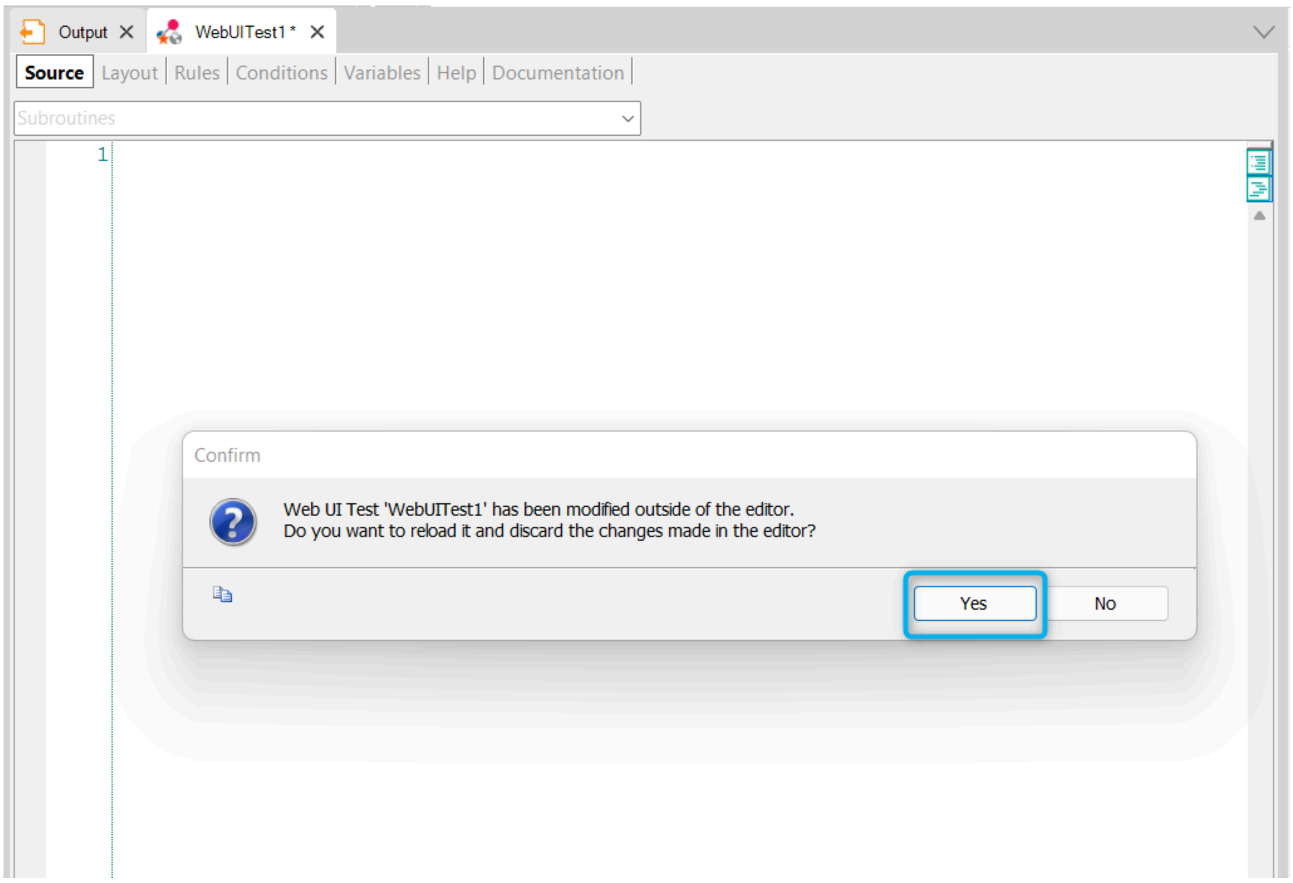


Importing the UI Test to the KB

1) Click on the “Copy to clipboard” button to export the UI Test code.

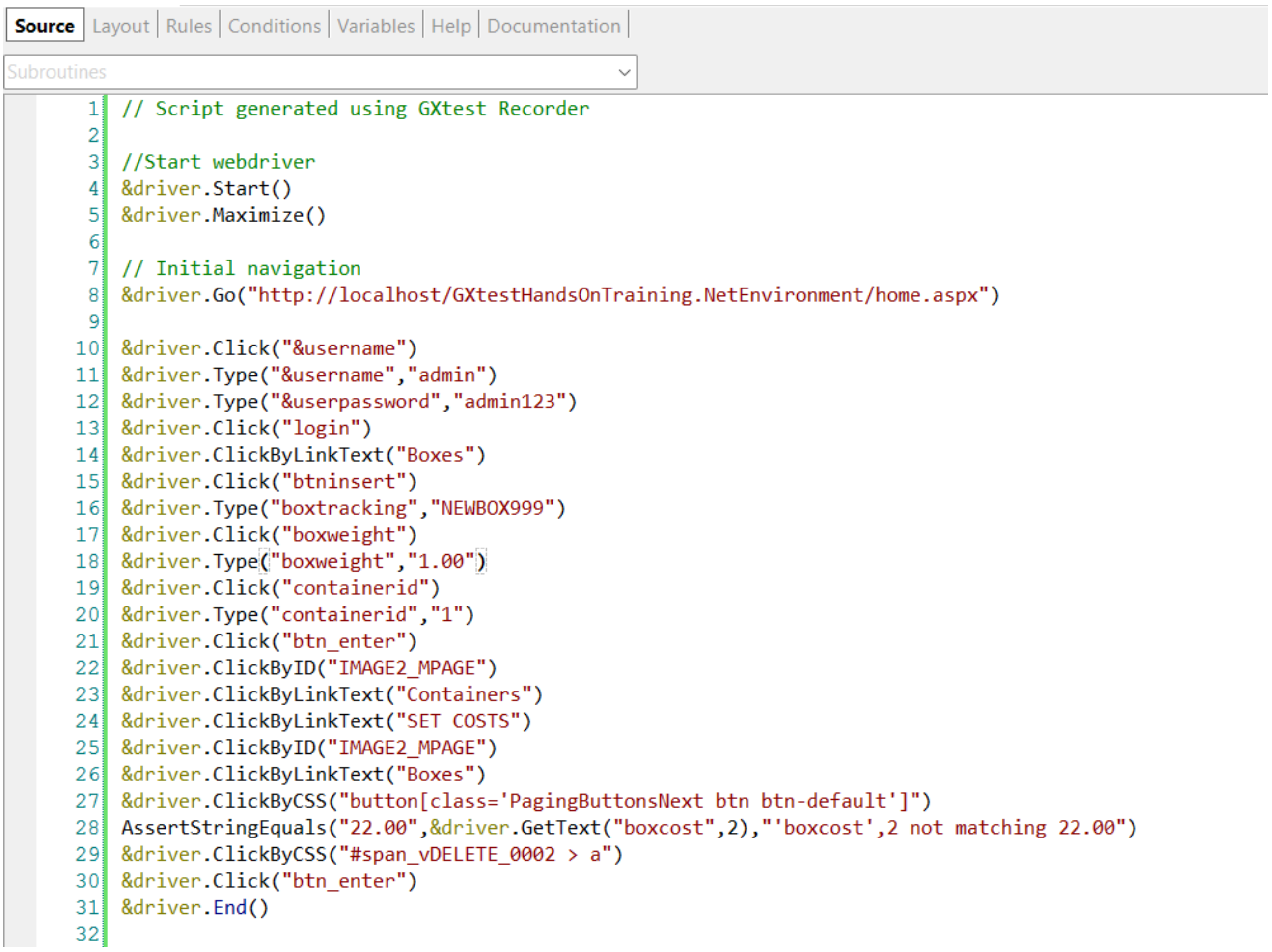


2) Return to the IDE. A message will state that the test has been modified outside the IDE and will request confirmation to load the object. Click on "Yes" to accept.



Web UI Test 'WebUITest1' has been modified outside of the editor.
Do you want to reload it and discard the changes made in the editor?

3) The test object will be updated and the script captured with the recorder extension could be executed after updating the URL of Go command. The Go command have to receive as parameter the home panel URL (*home.aspx*).



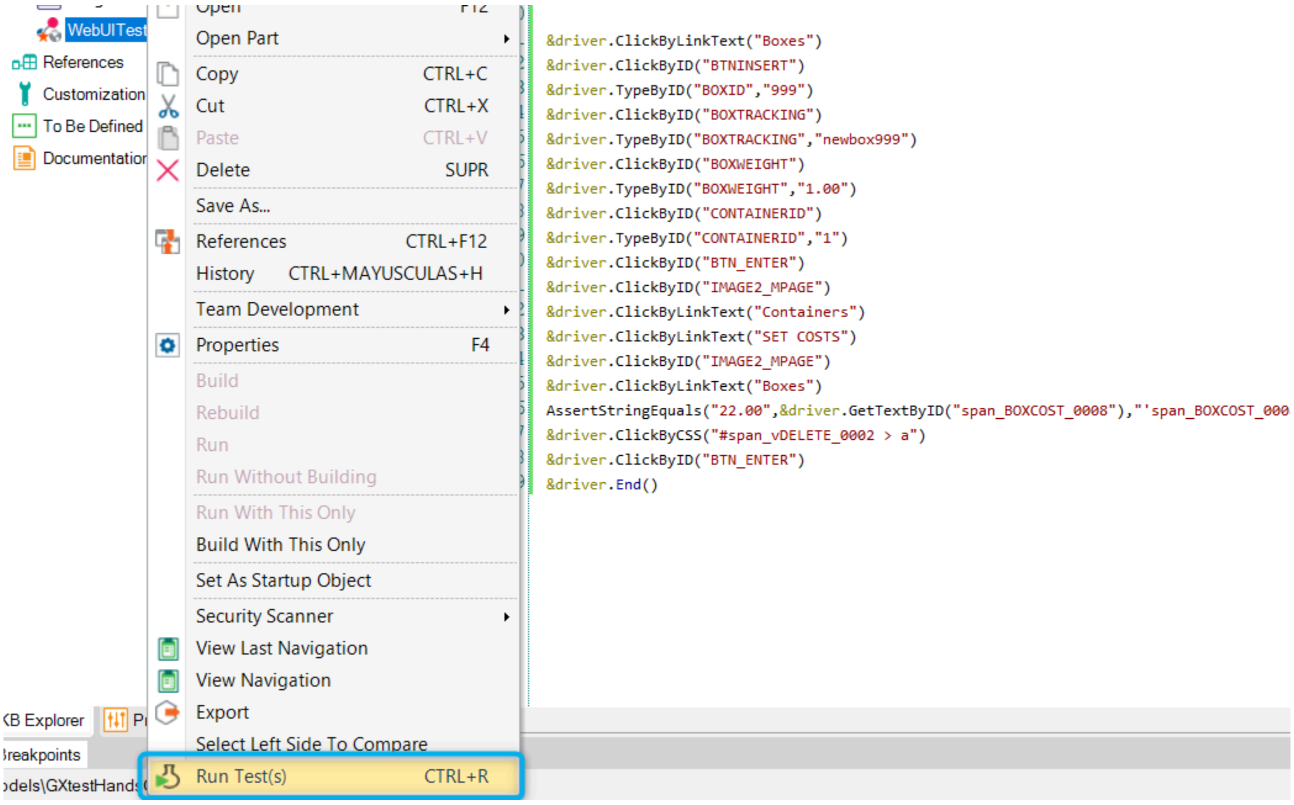
```
// Script generated using GXtest Recorder

//Start webdriver
&driver.Start()
&driver.Maximize()

// Initial navigation
&driver.Go("http://localhost/GXtestHandsOnTraining.NetEnvironment/home.aspx")

&driver.Click("&username")
&driver.Type("&username","admin")
&driver.Type("&userpassword","admin123")
&driver.Click("login")
&driver.ClickByLinkText("Boxes")
&driver.Click("btninsert")
&driver.Type("boxtracking","NEWBOX999")
&driver.Click("boxweight")
&driver.Type("boxweight","1.00")
&driver.Click("containerid")
&driver.Type("containerid","1")
&driver.Click("btn_enter")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Containers")
&driver.ClickByLinkText("SET COSTS")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Boxes")
&driver.ClickByCSS("button[class='PagingButtonsNext btn btn-default']")
AssertStringEquals("22.00",&driver.GetText("boxcost",2),"'boxcost',2 not matching 22.00")
&driver.ClickByCSS("#span_vDELETE_0002 > a")
&driver.Click("btn_enter")
&driver.End()

```

The generated code is:

```
// Script generated using GXtest Recorder

//Start webdriver
&driver.Start()
&driver.Maximize()

// Initial navigation
//update the URL before running the test!!!!
&driver.Go("http://localhost/GXtestHandsOnTraining.NetEnvironment/home.aspx")

&driver.Click("&username")
&driver.Type("&username","admin")
&driver.Type("&userpassword","admin123")
&driver.Click("login")
&driver.ClickByLinkText("Boxes")
```

```
&driver.Click("btninsert")
&driver.Type("boxtracking","NEWBOX999")
&driver.Click("boxweight")
&driver.Type("boxweight","1.00")
&driver.Click("containerid")
&driver.Type("containerid","1")
&driver.Click("btn_enter")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Containers")
&driver.ClickByLinkText("SET COSTS")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Boxes")
&driver.ClickByCSS("button[class='PagingButtonsNext btn btn-default']")
AssertStringEquals("22.00",&driver.GetText("boxcost",2),"'boxcost',2 not matching 22.00")
&driver.ClickByCSS("#span_vDELETE_0002 > a")
&driver.Click("btn_enter")
&driver.End()
```

4) Save changes. Right-click on the object and select the option "Run Test(s)".

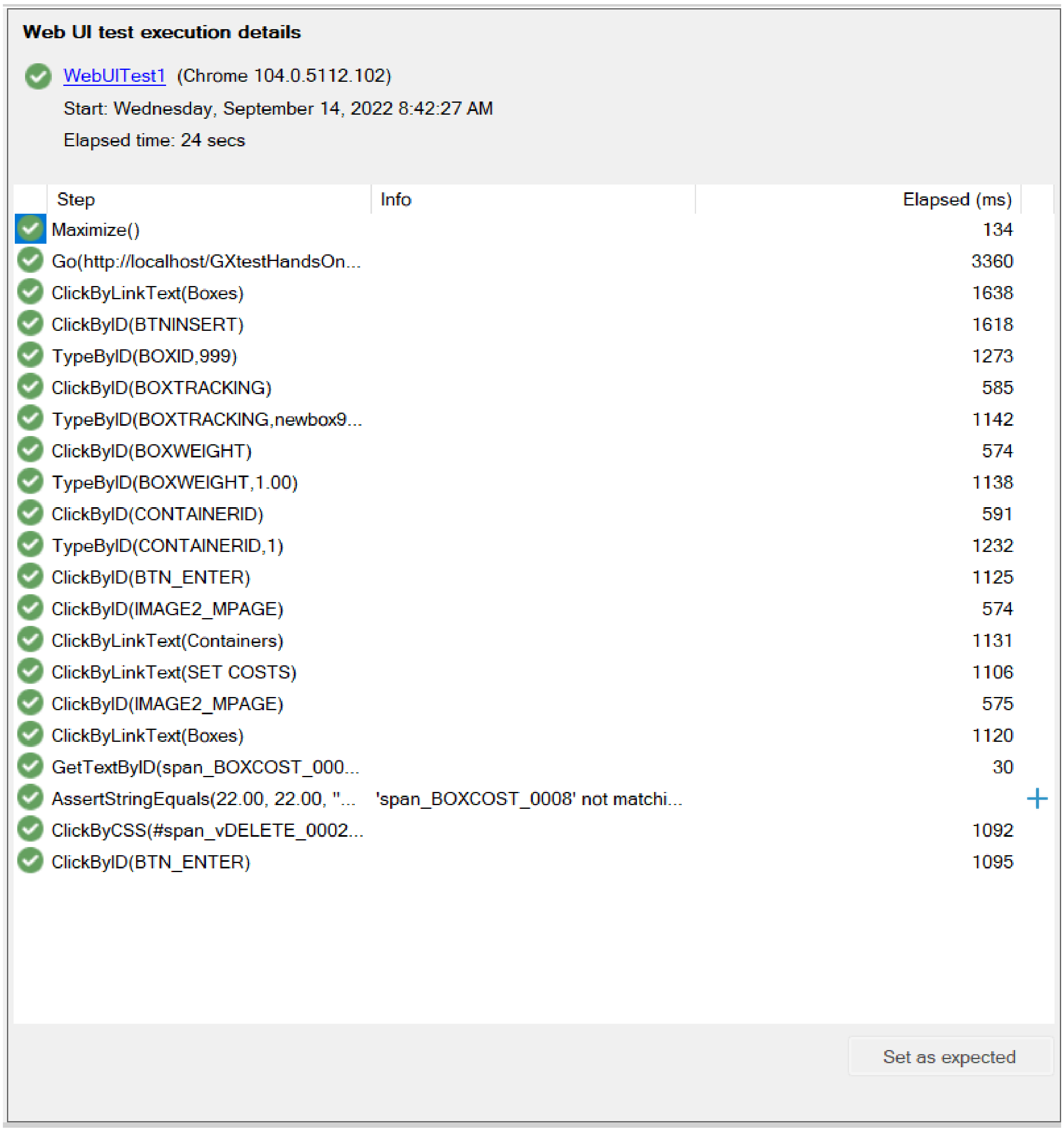


```
&driver.ClickByLinkText("Boxes")
&driver.ClickByID("BTNINSERT")
&driver.TypeByID("BOXID","999")
&driver.ClickByID("BOXTRACKING")
&driver.TypeByID("BOXTRACKING","newbox999")
&driver.ClickByID("BOXWEIGHT")
&driver.TypeByID("BOXWEIGHT","1.00")
&driver.ClickByID("CONTAINERID")
&driver.TypeByID("CONTAINERID","1")
&driver.ClickByID("BTN_ENTER")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Containers")
&driver.ClickByLinkText("SET COSTS")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Boxes")
AssertStringEquals("22.00",&driver.GetTextByID("span_BOXCOST_0008"),"'span_BOXCOST_000
&driver.ClickByCSS("#span_vDELETE_0002 > a")
&driver.ClickByID("BTN_ENTER")
&driver.End()
```

If execution fails, verify there are no differences between your code and the code in step 3.

Observe the results of the test when execution is finished.

Explore the information and buttons in the "Test Results" screen.



Web UI test execution details

✓ WebUITest1 (Chrome 104.0.5112.102)

Start: Wednesday, September 14, 2022 8:42:27 AM

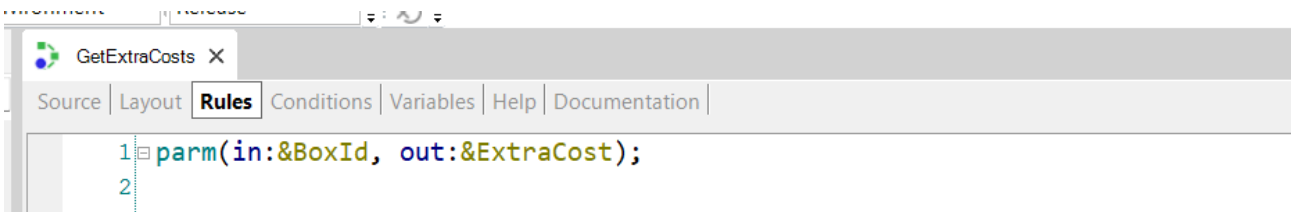
Elapsed time: 24 secs

	Step	Info	Elapsed (ms)
✓	Maximize()		134
✓	Go(http://localhost/GXtestHandsOn...		3360
✓	ClickByLinkText(Boxes)		1638
✓	ClickByID(BTNINSERT)		1618
✓	TypeByID(BOXID,999)		1273
✓	ClickByID(BOXTRACKING)		585
✓	TypeByID(BOXTRACKING,newbox9...		1142
✓	ClickByID(BOXWEIGHT)		574
✓	TypeByID(BOXWEIGHT,1.00)		1138
✓	ClickByID(CONTAINERID)		591
✓	TypeByID(CONTAINERID,1)		1232
✓	ClickByID(BTN_ENTER)		1125
✓	ClickByID(IMAGE2_MPAGE)		574
✓	ClickByLinkText(Containers)		1131
✓	ClickByLinkText(SET COSTS)		1106
✓	ClickByID(IMAGE2_MPAGE)		575
✓	ClickByLinkText(Boxes)		1120
✓	GetTextByID(span_BOXCOST_000...		30
✓	AssertStringEquals(22.00, 22.00, "...	'span_BOXCOST_0008' not matchi...	+
✓	ClickByCSS(#span_vDELETE_0002...		1092
✓	ClickByID(BTN_ENTER)		1095

Test Coverage

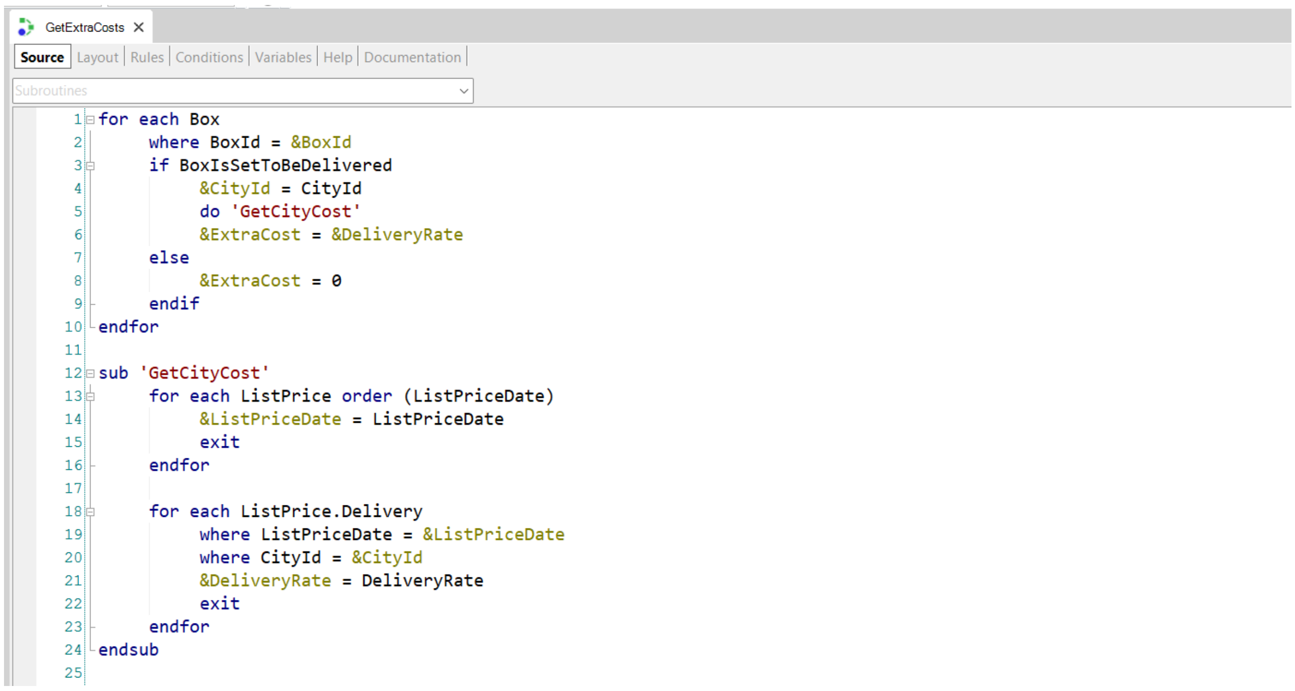
It is possible to know the percentage of lines of code executed for each object consumed by a Unit Test. In this section, you will be guided through creating and executing a Unit Test which consumes other objects in the KB, as well as analyzing its coverage. Let's learn how!

1) Open the procedure "**GetExtraCost**" and click on the "Rules" tab. You will see it receives *&BoxId* as input parameter (this is the ID of the box which will get an extra cost) and it returns *&ExtraCost* as output parameter (this is the extra cost for each box according to the city assigned for shipment).


GetExtraCosts
Source | Layout | Rules | Conditions | Variables | Help | Documentation

```
parm(in:&BoxId, out:&ExtraCost);
```

2) Click on the "Source" tab. The code of the procedure assigns an extra cost for each box which matches the input parameter *&BoxId* if and only if the box is ready for delivery. In case it is not, it returns "ExtraCost = '0'".

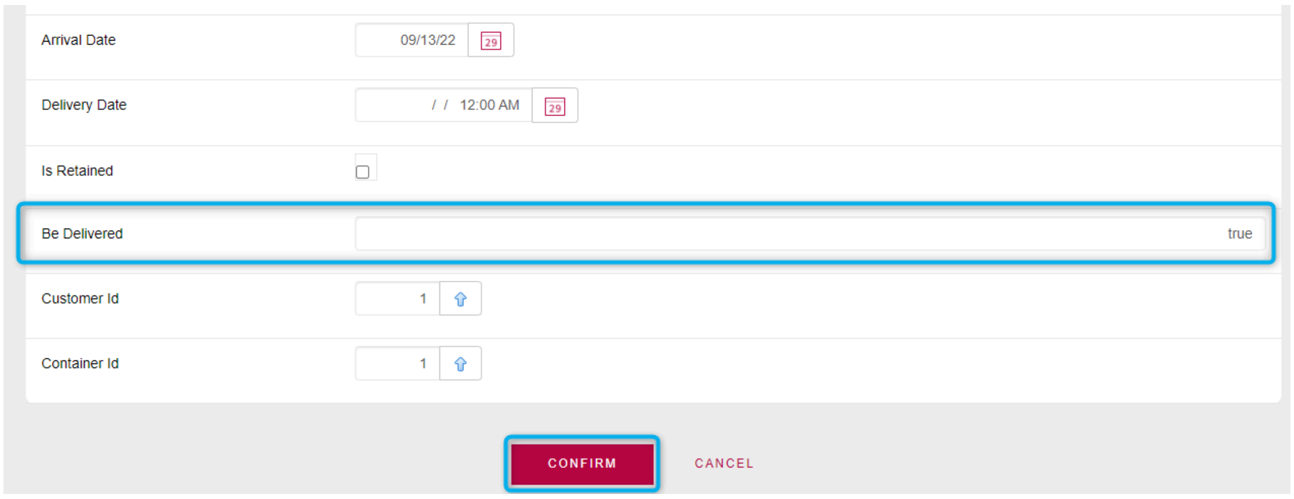

GetExtraCosts
Source | Layout | Rules | Conditions | Variables | Help | Documentation
Subroutines

```
for each Box
      where BoxId = &BoxId
      if BoxIsSetToBeDelivered
            &CityId = CityId
            do 'GetCityCost'
            &ExtraCost = &DeliveryRate
      else
            &ExtraCost = 0
      endif
endfor

sub 'GetCityCost'
      for each ListPrice order (ListPriceDate)
            &ListPriceDate = ListPriceDate
            exit
      endfor

      for each ListPrice.Delivery
            where ListPriceDate = &ListPriceDate
            where CityId = &CityId
            &DeliveryRate = DeliveryRate
            exit
      endfor
endsub
```

For this procedure, we have 12 executable lines of code. If the *BoxId* inputted has the variable "BoxIsSetToBeDelivered" assigned as "True", 11 lines of this procedure will be executed. If not, only the sentence inside "else" will be executed.

You will try to reach total coverage, creating test cases which cover the entire procedure code. Let's begin!

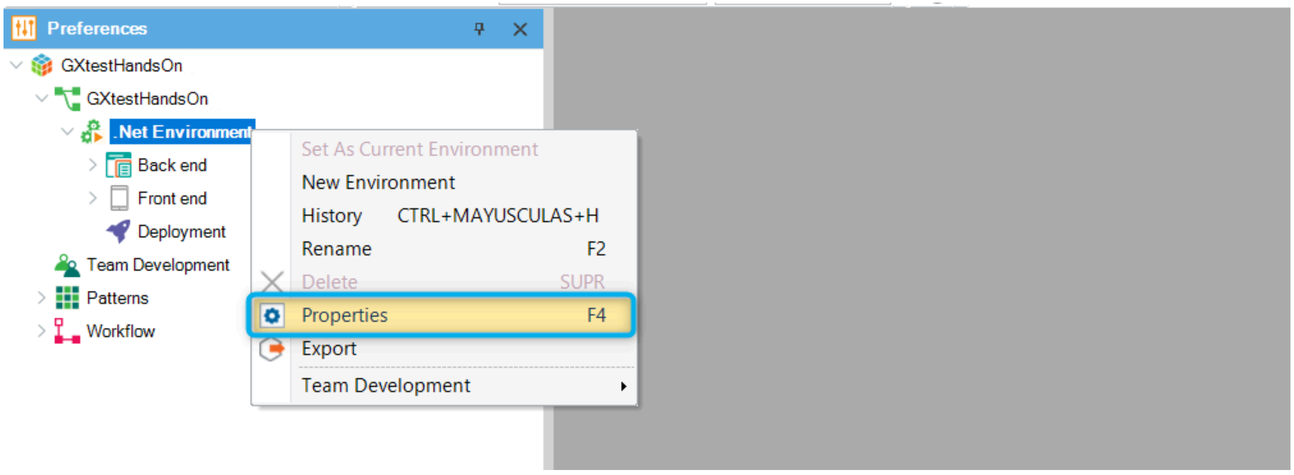
3) Before starting, open the application and select the **Box with Id=1**. Open the “Be Delivered” field, which is set to “False” by default, write “True”, and select “Confirm”.



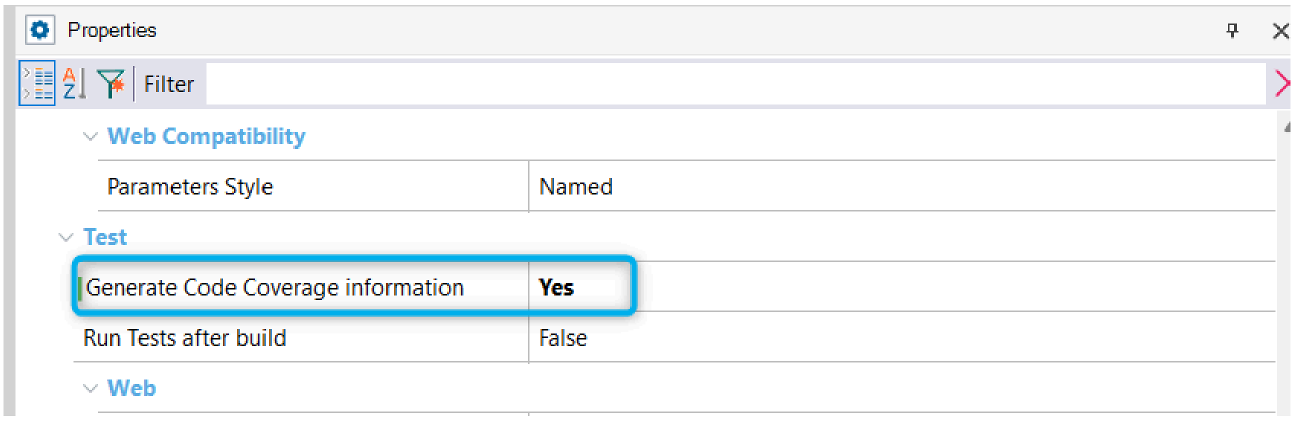
Arrival Date	09/13/22
Delivery Date	/ / 12:00 AM
Is Retained	☐
Be Delivered	true
Customer Id	1
Container Id	1

4) Return to the KB and create a Unit Test on the procedure “**GetExtraCost**” by right-clicking on it and selecting "Create Unit Test”.

5) Open the environment properties by right-clicking and selecting the option “Properties”.



6) Identify the property “Generate Code Coverage Information” and change it to “Yes”.



Web Compatibility

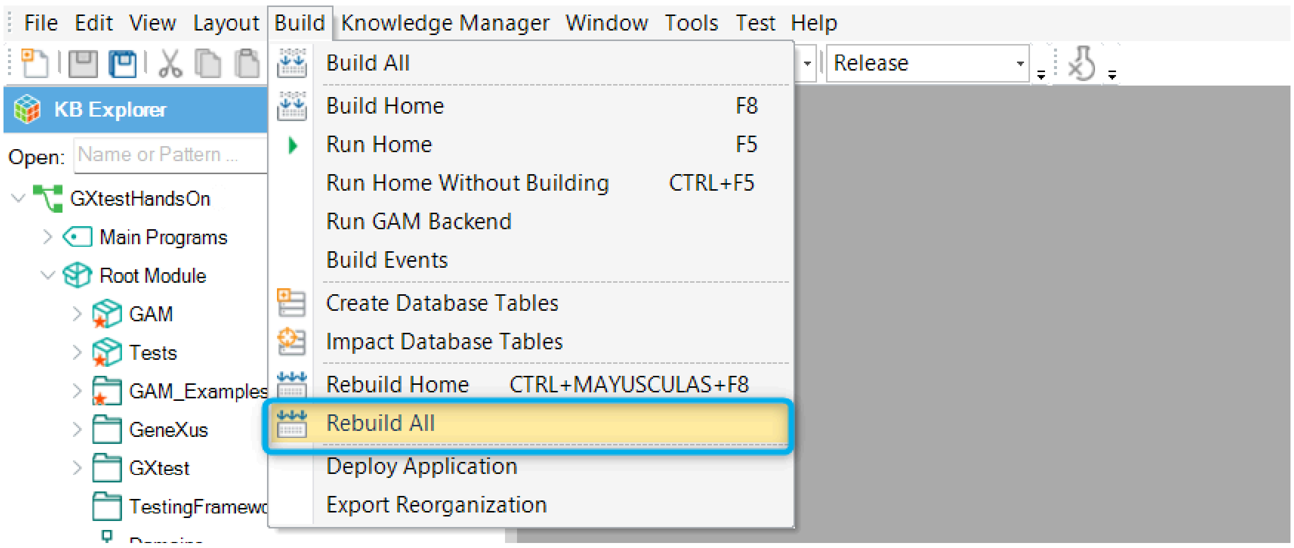
Parameters Style	Named

Test

Generate Code Coverage information	**Yes**
Run Tests after build	False

Web

7) Once enabled, select “Rebuild All” from the “Build” menu. This step is necessary for the objects to be created again with code coverage trackers.



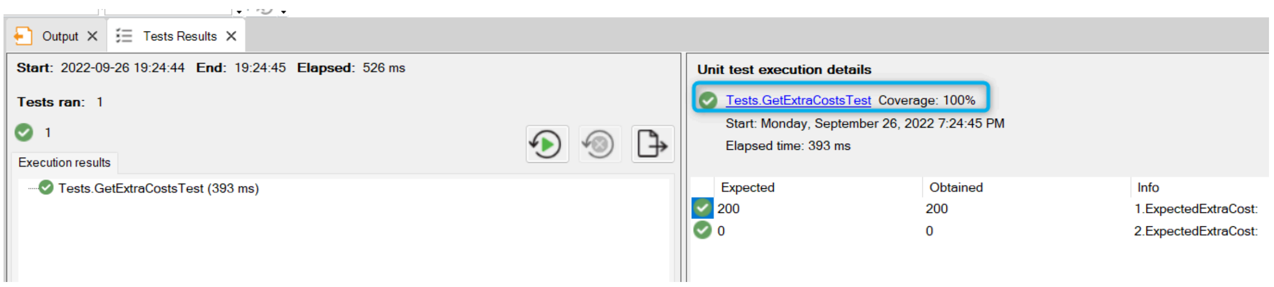
8) Open “GetExtraCostsTestData” and modify the values **BoxId = 1** and **ExpectedExtraCost = 200** in “TestCaseId=1”. In “TestCaseId=2”, enter values **BoxId = 2** and **ExpectedExtraCost = 0**.

The Data Provider code is:

```
GetExtraCostsTestSDT
{
        TestCaseId = '1'
        BoxId = 1
        ExpectedExtraCost = 200
        MsgExtraCost = '200'
}
GetExtraCostsTestSDT
{
        TestCaseId = '2'
        BoxId = 2
        ExpectedExtraCost = 0
        MsgExtraCost = '0'
}
```

9) Save changes and run “GetExtraCostsTest”.

When the test execution is finished, the results will be displayed in the Test Results screen. You will see that 100% of the code of the objects consumed by the test was covered.



Start: 2022-09-26 19:24:44 **End:** 19:24:45 **Elapsed:** 526 ms

Tests ran: 1

✓ 1

Execution results

✓ Tests.GetExtraCostsTest (393 ms)

Unit test execution details

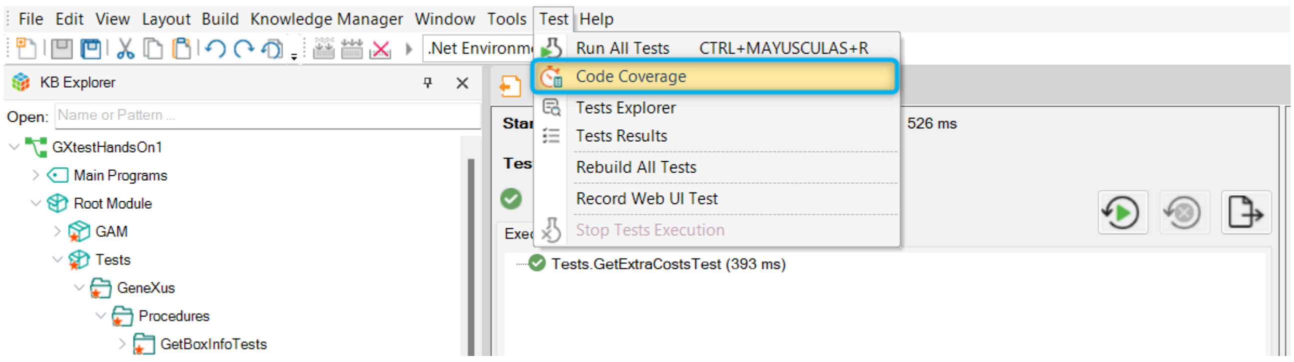
✓ Tests.GetExtraCostsTest Coverage: 100%

Start: Monday, September 26, 2022 7:24:45 PM

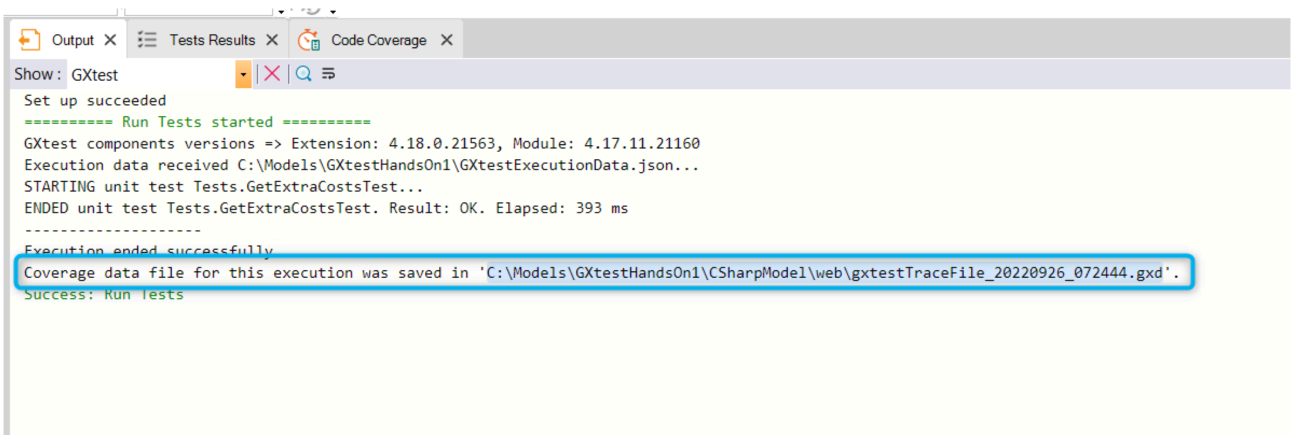
Elapsed time: 393 ms

	Expected	Obtained	Info
✓	200	200	1.ExpectedExtraCost:
✓	0	0	2.ExpectedExtraCost:

10) Select the “Test” tab and click on the option “Code Coverage”.

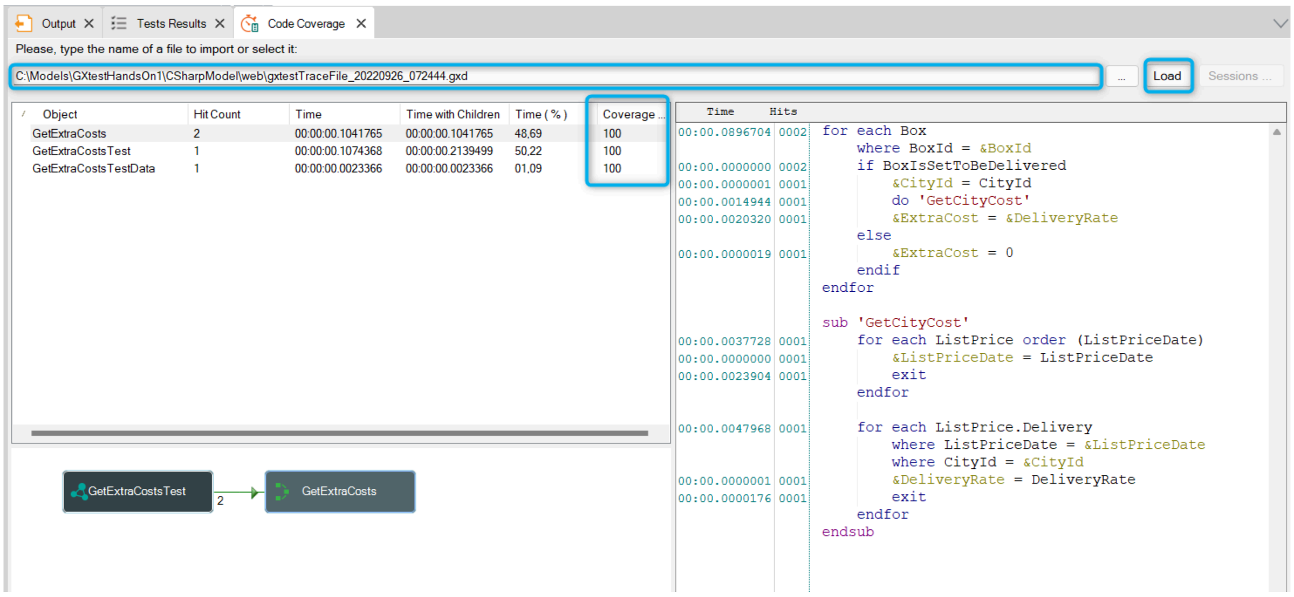


11) Copy the path indicated in the GXtest output, where the execution information was saved.



```
Set up succeeded
========== Run Tests started ==========
GXtest components versions => Extension: 4.18.0.21563, Module: 4.17.11.21160
Execution data received C:\Models\GXtestHandsOn1\GXtestExecutionData.json...
STARTING unit test Tests.GetExtraCostsTest...
ENDED unit test Tests.GetExtraCostsTest. Result: OK. Elapsed: 393 ms
--------------------
Execution ended successfully
Coverage data file for this execution was saved in 'C:\Models\GXtestHandsOn1\CSharpModel\web\gxtestTraceFile_20220926_072444.gxd'.
Success: Run Tests
```

12) Paste the path on the indicated field inside “Code Coverage” and click on “Load”.



Please, type the name of a file to import or select it:

C:\Models\GXtestHandsOn1\CSharpModel\web\gxtestTraceFile_20220926_072444.gxd

Object	Hit Count	Time	Time with Children	Time (%)	Coverage ...
GetExtraCosts	2	00:00:00.1041765	00:00:00.1041765	48,69	100
GetExtraCostsTest	1	00:00:00.1074368	00:00:00.2139499	50,22	100
GetExtraCostsTestData	1	00:00:00.0023366	00:00:00.0023366	01,09	100

```
    Time       Hits
00:00.0896704 0002 for each Box
                       where BoxId = &BoxId
00:00.0000000 0002     if BoxIsSetToBeDelivered
00:00.0000001 0001         &CityId = CityId
00:00.0014944 0001         do 'GetCityCost'
00:00.0020320 0001         &ExtraCost = &DeliveryRate
                       else
00:00.0000019 0001         &ExtraCost = 0
                       endif
                   endfor

                   sub 'GetCityCost'
00:00.0037728 0001     for each ListPrice order (ListPriceDate)
00:00.0000000 0001         &ListPriceDate = ListPriceDate
00:00.0023904 0001         exit
                       endfor

00:00.0047968 0001     for each ListPrice.Delivery
                           where ListPriceDate = &ListPriceDate
                           where CityId = &CityId
00:00.0000001 0001         &DeliveryRate = DeliveryRate
00:00.0000176 0001         exit
                       endfor
                   endsub
```

The coverage percentage will be displayed in the “Coverage” column.

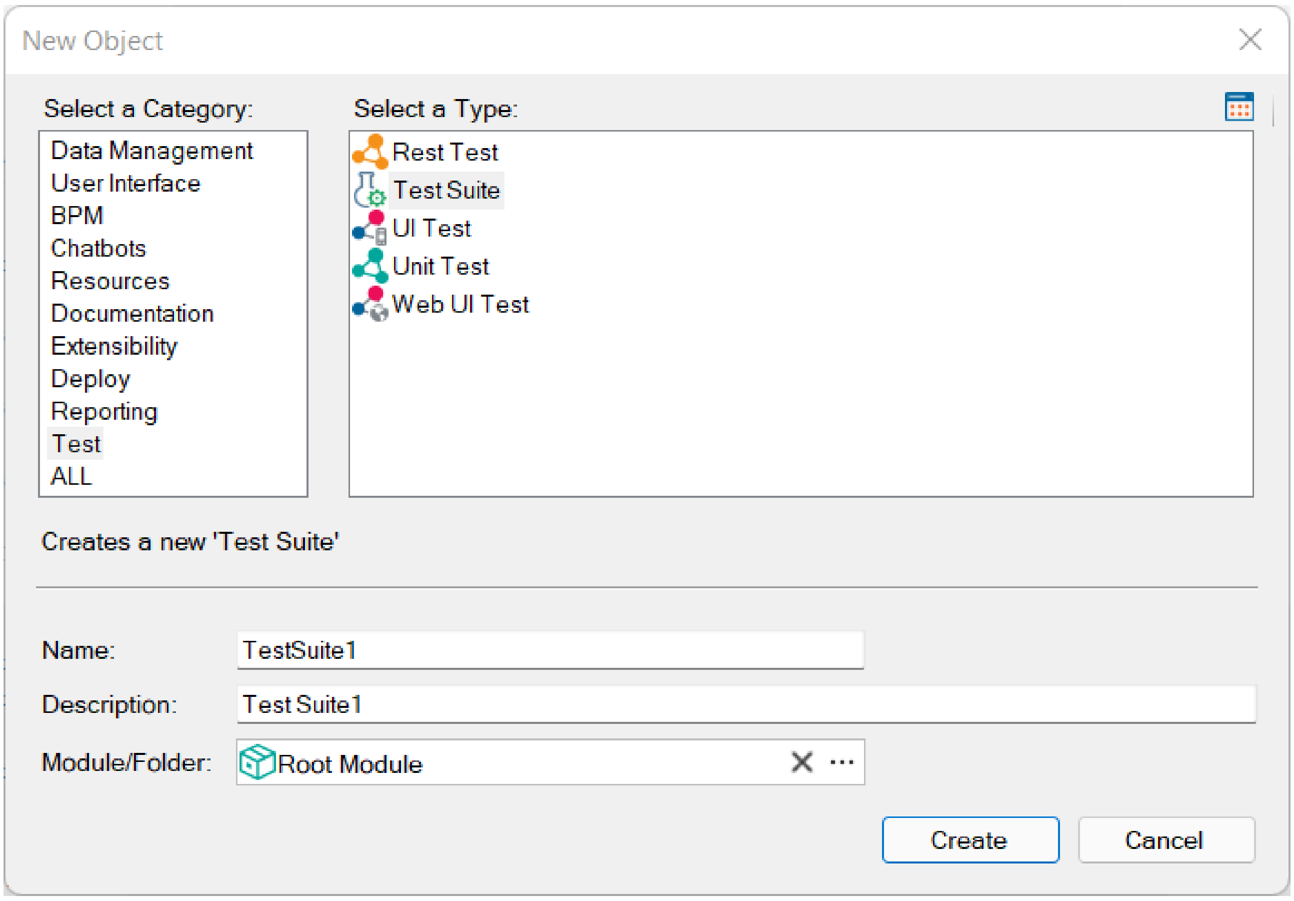
100% of the code was covered because, in the test cases, a *BoxId=1* was entered which meets the first condition and, therefore, it traces the 12 lines for that scenario, while *BoxId=2* corresponds to the execution of the sentence “else”, reaching total coverage.

Test Suites

It is possible to group tests by creating a *Test Suite* object, to execute them as a unit in a specific order. Let's implement a Test Suite object where you will place the tests you've created!

Creating and Executing a Test Suite

1) Create a Test Suite object in File > New > Object, selecting the “Test” category, selecting the “Test Suite” object, and pressing "Create”.


New Object

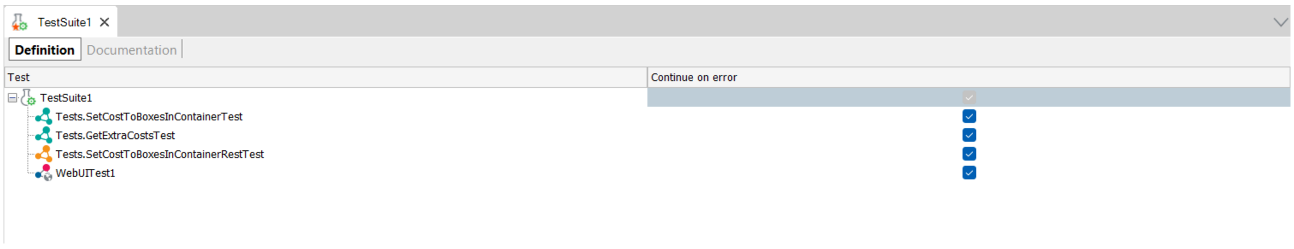
Select a Category: Data Management, User Interface, BPM, Chatbots, Resources, Documentation, Extensibility, Deploy, Reporting, Test, ALL

Select a Type: Rest Test, Test Suite, UI Test, Unit Test, Web UI Test

Creates a new 'Test Suite'

Name:	TestSuite1
Description:	Test Suite1
Module/Folder:	Root Module

2) Select and drag from the *KB Explorer* screen the previously created Unit Tests “SetCostToBoxesInContainerTest” and “GetExtraCostTest”, the Rest Test “SetCostToBoxesInContainerRestTest”, and the Web UI test “WebUITest1”. It should look like this:

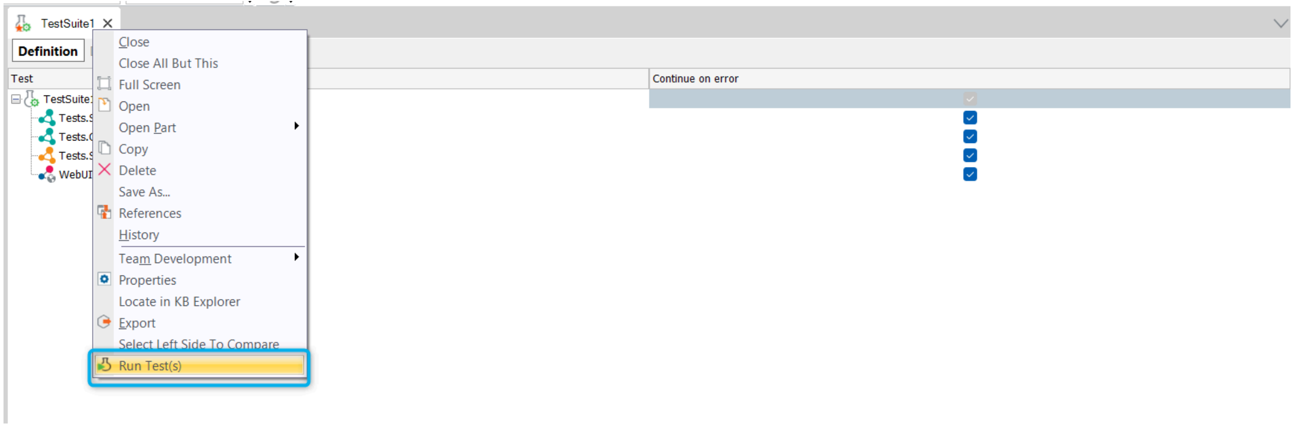


TestSuite1

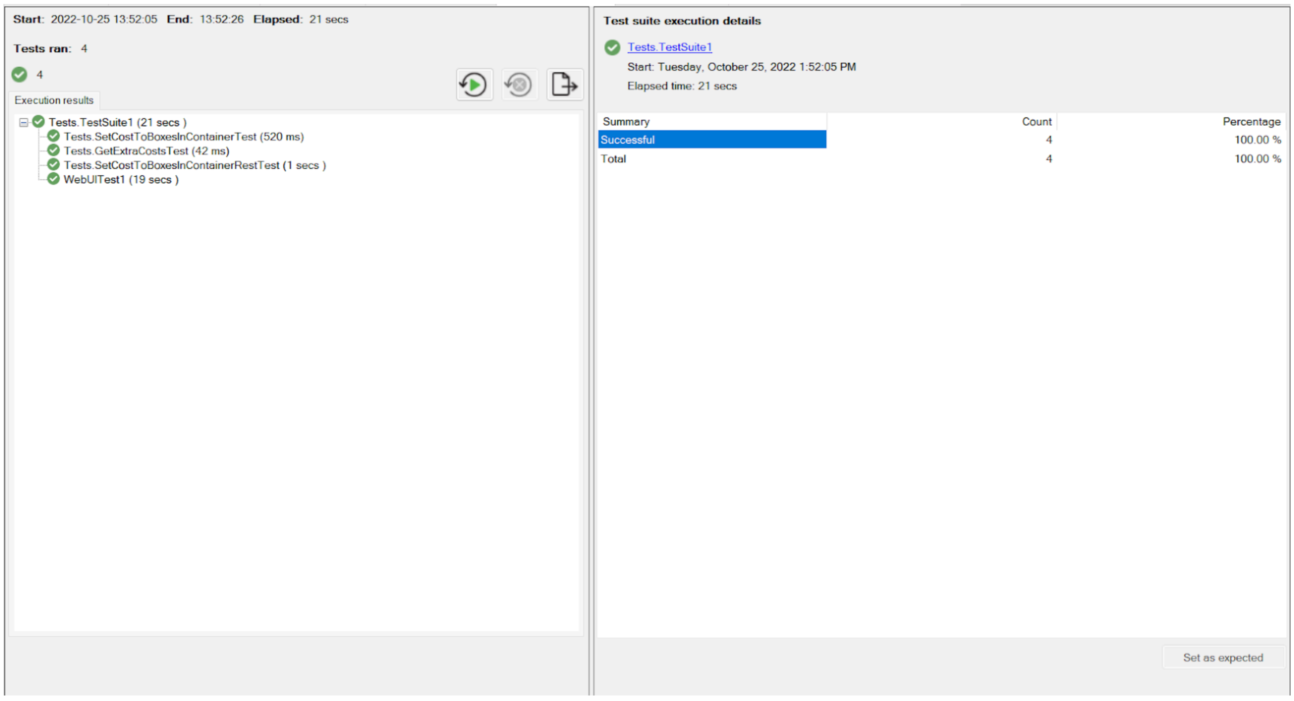
Definition | Documentation

Test	Continue on error
TestSuite1	
Tests.SetCostToBoxesInContainerTest	✓
Tests.GetExtraCostsTest	✓
Tests.SetCostToBoxesInContainerRestTest	✓
WebUITest1	✓

3) Right-click on the new *Test Suite* object and click on "Run Test(s)".



The tests included in the suite will be executed in the specified order. In this case, the execution is successful, since all the *Test Suite* tests have passed.



Start: 2022-10-25 13:52:05 **End:** 13:52:26 **Elapsed:** 21 secs

Tests ran: 4

✓ 4

Execution results

- Tests.TestSuite1 (21 secs)
  - Tests.SetCostToBoxesInContainerTest (520 ms)
  - Tests.GetExtraCostsTest (42 ms)
  - Tests.SetCostToBoxesInContainerRestTest (1 secs)
  - WebUITest1 (19 secs)

Test suite execution details

Tests.TestSuite1

Start: Tuesday, October 25, 2022 1:52:05 PM

Elapsed time: 21 secs

Summary	Count	Percentage
Successful	4	100.00 %
Total	4	100.00 %

MONTEVIDEO - URUGUAY	Sarmiento 2465, 11300	+598 2711 0561
SAN FRANCISCO - USA	100 Pine St., Ste. 1250, CA 94111	+1 415 745 3678
LONDRES - UK	71-75 Shelton Street, WC2H 9JQ	+44 203 696 6682