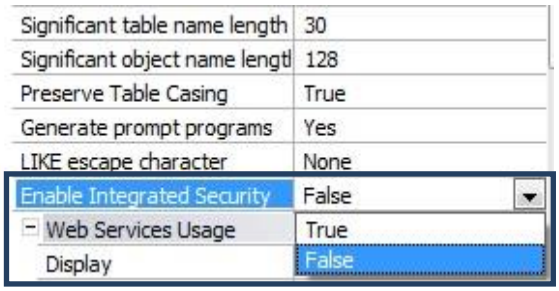
Introduction to the GAM

Most modern applications need some type of login, authentication and authorization scheme.

To meet these requirements, GeneXus provides a security module called GeneXus Access Manager (GAM). This module takes care of the authentication and authorization features for Web and Smart Device applications.

To use this module with all the security controls available, just set the Enable integrated security property to True in your knowledge base, at the active version level.



After that, this dialog box is displayed for you to enable the GAM. Click on “Install”.

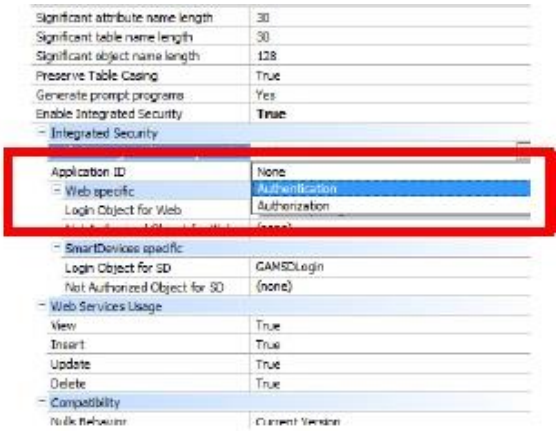
As a result, a security module developed with GeneXus will be imported. This module is integrated into our application to solve everything related to security.

Note that the Output window shows several objects being imported. These objects belong to the GAM module.

Once the security feature is enabled, we can select whether to use only the Authentication feature or Authentication+Authorization.

This is done by setting the **Integrated Security Level** property.

For now, we'll only use the Authentication feature.



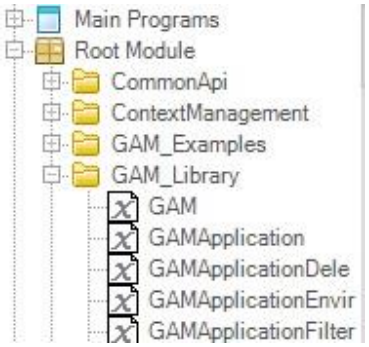
When the GAM is enabled, several changes are made in addition to importing objects.

For example, properties are enabled to configure the login object for Web and Smart Device applications.

Look at the **Login Object for Web** property. It has the value GAMExampleLogin to indicate that this object will be used for Web application login.

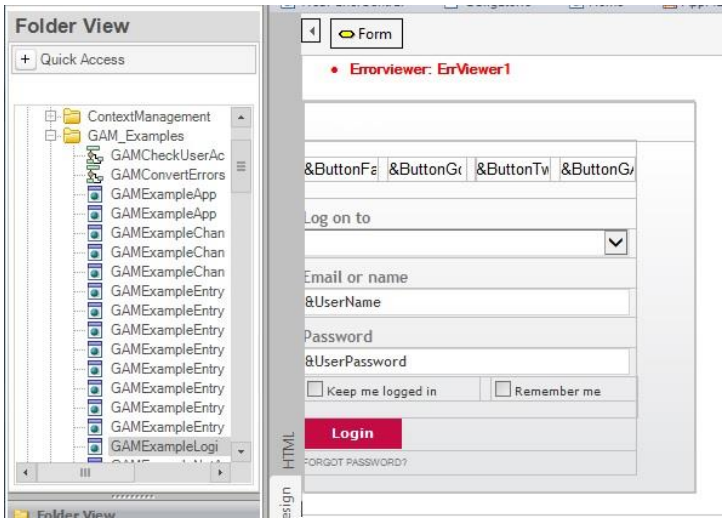
In addition, the **Login Object for SD** property is set to GAMSDLogin, indicating the name of the object that will perform the login in Smart Device applications.

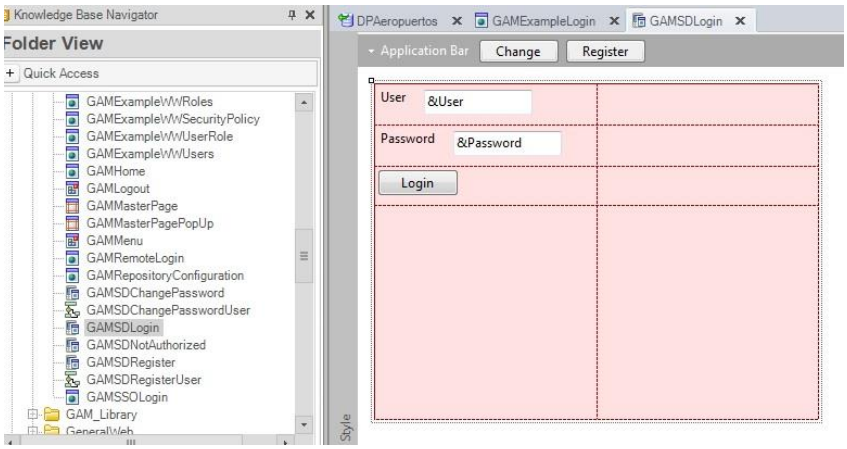
The objects imported after enabling the GAM can be found in the folders GAM_Examples and GAM_Library.



The GAM_Examples folder contains all the sample objects imported. Let's take a look at the contents of Web Panels and Panels for Smart Devices. These objects will be used for the authentication and authorization of users.

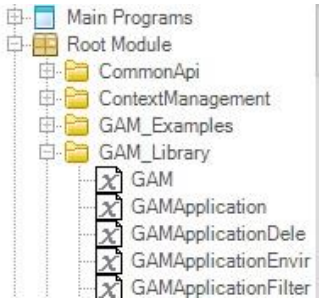
More precisely, the objects GAMExampleLogin and GAMSDLogin are configured, as we've seen before, in the properties Login Object for Web and Login Object for Smart Devices.





Several objects make up the GAM backend. That is to say, the backend is a Web application used to manage and configure users, roles, permissions, and so on. We will see it in a few minutes.

In the GAM_Library folder we can see external objects with the necessary settings to run the GAM APIs. APIs are functions that allow our KB to communicate with the GAM database, which is different than the database associated with our application. The GAM database contains information about users, roles, etc.



Remember that the **Rebuild All** action must be run in the KB after enabling the GAM.

At this moment, we are asked to create the database associated with the GAM.
We click on Yes.

When the Rebuild All operation is completed, we can run the application with the GAM enabled.
So, we press F5.
Let's try, for example, to access Work With Country.

Note that a login object is executed first.

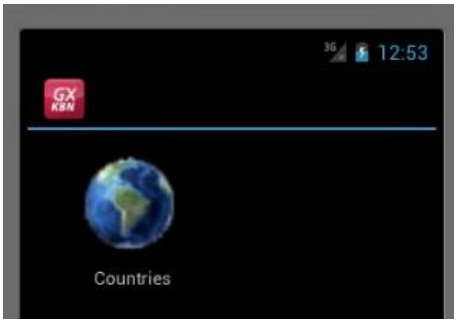
This object is automatically run as needed.

In this case, since we haven't defined any type of authentication such as Facebook or Twitter, only local authentication is enabled by default. We can enter with the user "admin" and password: "admin123".

The login object was executed simply by setting the properties required to enable the GAM, without programming anything else. The reason is that when the GAM is used, an **automatic access control is performed in each object.**

Now, let's run the Smart Device application. We can see that here the login panel is also displayed first. So, we enter with username admin and password admin123.

Once the login data has been entered, redirection is made to the object that it was trying to run. In this case, it is the Dashboard.

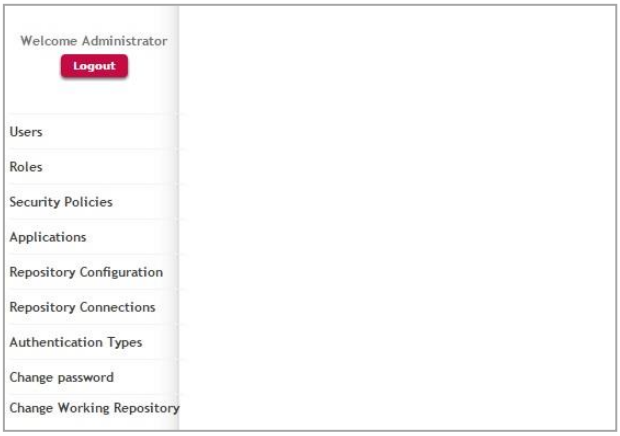


This is the default behavior for Web and Smart Device applications.

As we've said before, among the objects imported after enabling the GAM there's a group that implements the Backend to manage users, roles, and so on.

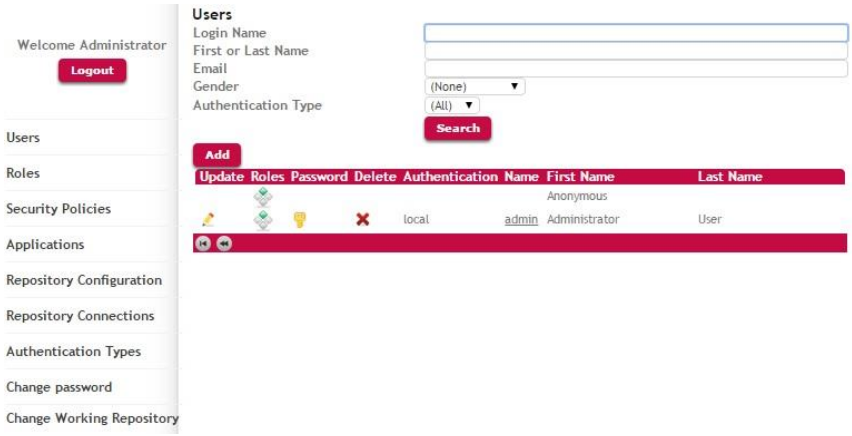
To access the Backend at runtime, from the Developer Menu we need to run GAMHome, the main object of the GAM Backend.

We can see that on the left there's a menu to access the various Backend options.



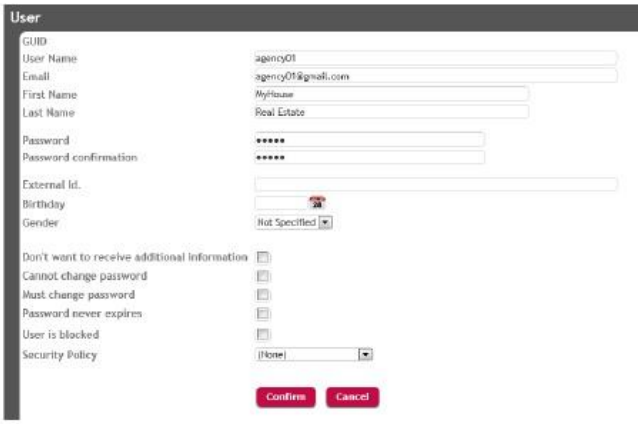
Let's open the Users option.

Here we can see all the users defined. Only the admin user is included by default. It is automatically created when the GAM is applied, and it is what we're using to log in.



We will create a new user for one of the travel agents who will use the application that we're building.

To do so, we press the Add button... and enter the user's details...



We indicate that the authentication is local; the username is “pjones”, the email address is pjones@gmail.com, the name is “Peter” and the last name is “Jones”. In addition, we enter the password “pjones123” and confirm it by entering it again: “pjones123”.

Our user is now created.

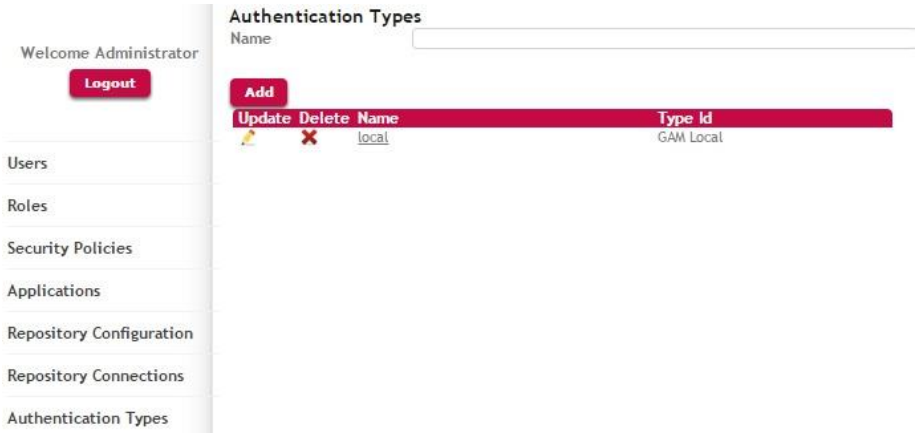
Next, we will associate it with a Role.

So, we click on Role and select the Administrator role.

We click on Add, and in this way the Administrator role is assigned to user pjones.

Now we open the Authentication Types option and see that only the local authentication option is enabled by default.

Here we need to define the different authentication types that we want to use in our application, such as Facebook or Twitter.



This video has shown that we can easily implement secure GeneXus applications using the GAM -GeneXus Access Manager-, which provides a comprehensive and integrated solution for Authentication and Authorization in Web and Smart Device applications.