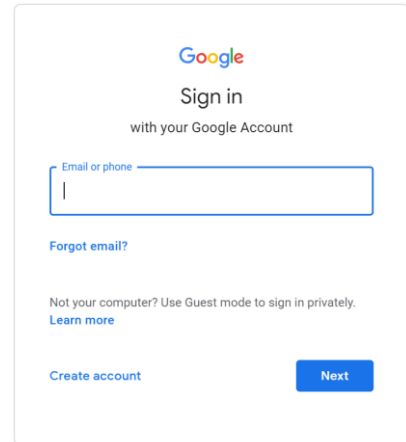
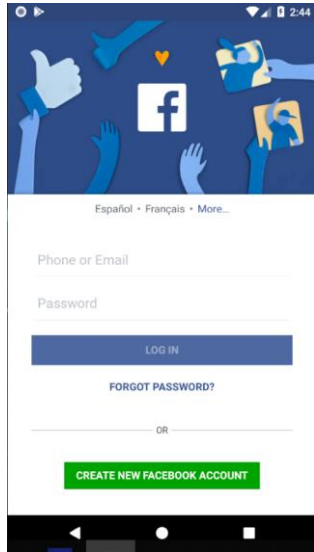


Security in Angular Applications

GeneXus™

Security



Most current applications require a security scheme so that only allowed users can access it, and also to authorize or restrict access to parts of the application according to the permissions assigned to each user.

Security

LOGIN

admin

.....

Keep me logged in

Remember Me

LOG IN

[FORGOT YOUR PASSWORD?](#)
or [create an account](#)

← Login CHANGE PASSWORD REGISTER

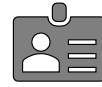
User

User

Password

Password

LOGIN



Authorization

USERS ROLES SETTINGS - Administrator

Add Permission Try a Search ← BACK + ADD SELECTED

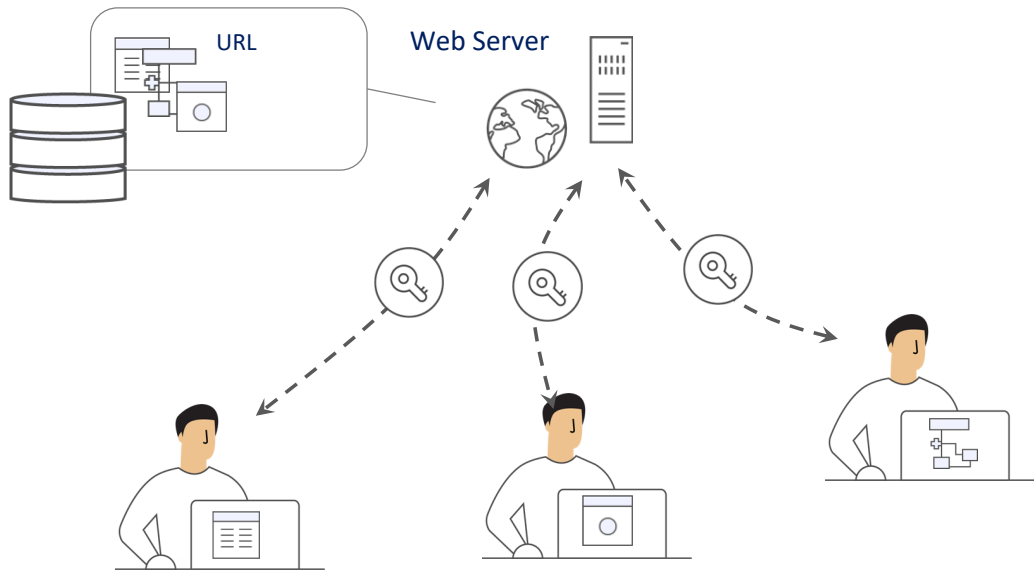
APPLICATION	GAM Backend	ROLE	BackedUser
Sel...	Permission name	Description	Permissions options
<input checked="" type="checkbox"/>	gamexamplechangerepository_Execute	Change Working Repository	Allow
<input type="checkbox"/>	gamexamplechangeyourpassword_Execute	Change Password	Allow
<input checked="" type="checkbox"/>	gamexamplewapplications_Execute	Application	Restricted
<input checked="" type="checkbox"/>	gamexamplewaulhtypes_Execute	Authentication Types	Deny
<input type="checkbox"/>	gamexamplewconnections_Execute	Connections	Allow



Authentication

This means making sure that all users who access the application are properly authenticated (that is to say, they are who they claim to be) and authorized. Once a user is authenticated, they are granted or denied access to certain parts of the application.

Security in Web Applications



In web applications, since they have several points of entry, any object that can be accessed from a URL must check authentication permissions.

This implies that each one of these objects must have security checking incorporated in order to make the corresponding verification.

Integrated Security Solution

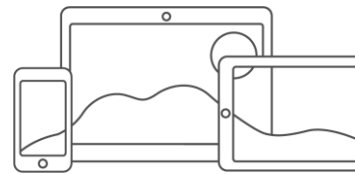


GeneXus Access Manager



Authentication

Authorization



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 both web and smart device applications.

The GAM has been developed with GeneXus, so it can be easily integrated into the application's KB in order to solve everything related to its security in a centralized manner. Its objective is to have the Security solution used in the most declarative way possible within the application, without adding complexity.

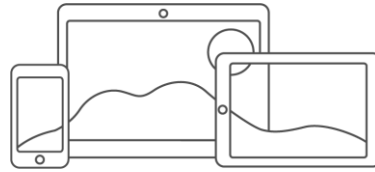
The GAM also provides a backend that allows creating users, permissions, security policies and access to objects, among other things.

In addition, it provides an API to access many of these features programmatically.

GAM Features

 Authentication

Web Sessions



OAuth



Authorization

RBAC

**Role-Based Access Control**

Internally, Authentication is implemented through:

- Web sessions for Web application security.
- OAuth for SD application security.

Authorization is implemented through Roles using the Role-Based Access Control model that encapsulates methods, properties, and everything necessary to manage the application's authorization.

GAM Features



The GAM provides several Authentication Types, which are as follows:

Local authentication using GAM where users and all their credentials are stored in a database that we own. It can also be done remotely, since an application using GAM can become an identity provider. In this case, other applications with GAM can remotely connect to this server and obtain authentication from there.

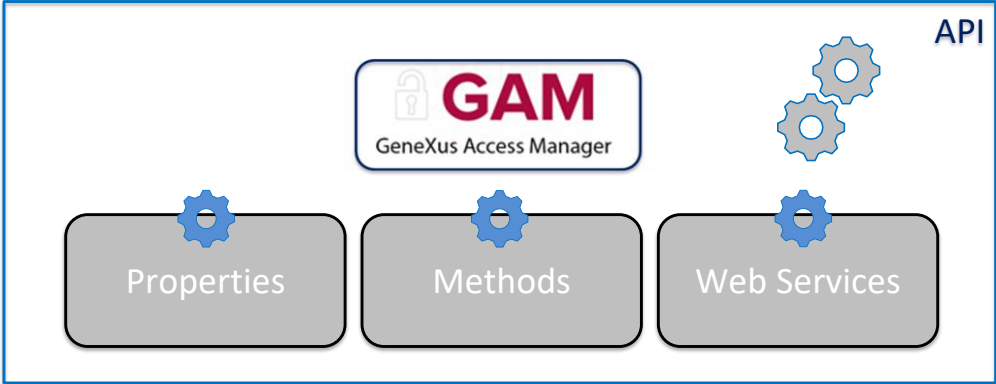
We can also use other external identity providers that provide authentication based on the OAuth 2.0 protocol such as Facebook, Twitter, Google, Instagram, Office 365, Mercado Libre, or LinkedIn. Here we use the standard authentication mechanisms based on this protocol implemented by these applications. In this case, there's no need to create local users.

Many times we need to integrate our application with other applications in order to exchange data, so we must implement the users' authentication using an external authentication method.

One example of external authentication is to use a SOAP web service provided by the other application and configure the GAM to consume this web service.

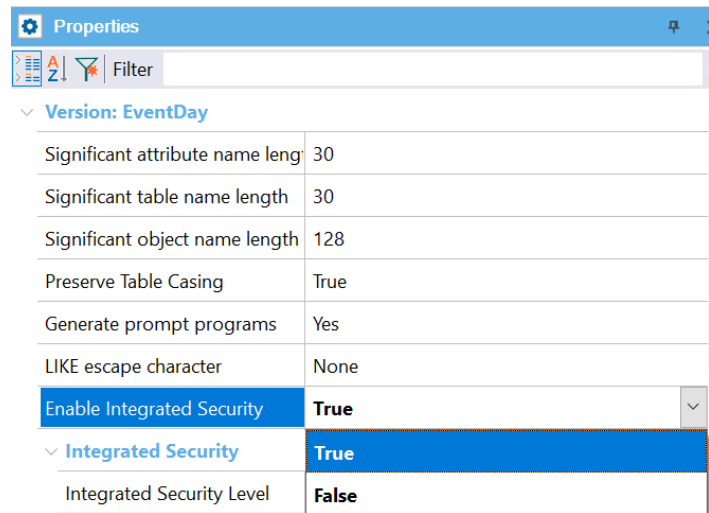
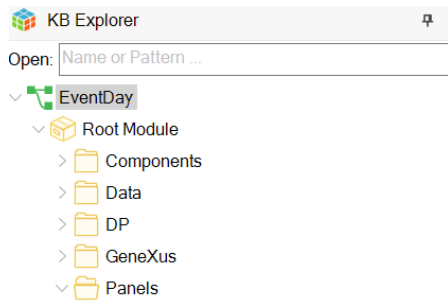
Also, the other application may provide an external program for authentication purposes and it may not be a web service. In this case, I configure the GAM to accept an authentication of Custom type.

GAM Features



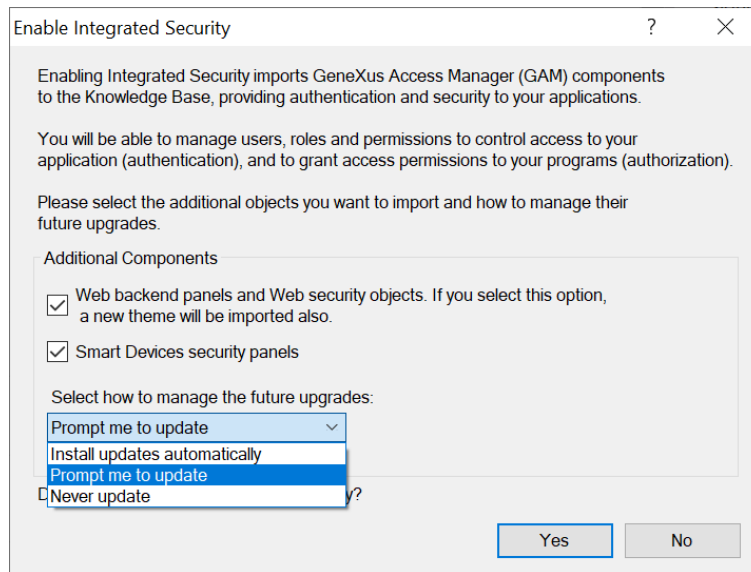
The GAM also exposes an API (Application Program Interface) to access its properties and methods in case it is necessary to do so from our application, and a series of Web services that can be used from other applications.

Enable Integrated Security



To enable GAM, go to the active version level of the KB and set the Enable Integrated Security property to True. In the Trial version, it is located in the first node of KB Explorer with the name of the KB.

Enable Integrated Security



This dialogue indicates that the integration will be made. Here we can indicate if we want to have the web backend integrated, and with this other one if we want security for the Smart Device panels to be integrated.

With this combo box, we can choose how to update this module. The options are to update it automatically, to prompt us for updates, or to never update it. We click on Yes.

Integrated Security Level

Integrated Security

Integrated Security Level	Authentication
Application ID	None
Web specific	Authentication
Login Object for Web	Authorization

Application ID	b3356369-b037-4216-982e-cbf8cfdc6a73
Web specific	
Login Object for Web	GAMExampleLogin
Not Authorized Object for	GAMExampleNotAuthorized
SmartDevices specific	
Login Object for SD	GAMSDLogin
Not Authorized Object for	GAMSDNotAuthorized
Change Password Object f	GAMSDChangePassword

Once GAM is enabled, we see another property called Integrated Security Level, which allows setting the default value for the security of KB objects. This property is also available at the object level, so it will be possible to customize how security will be implemented in that object.

It has three possible values:

- None: indicates that the object will be public; that is to say, it will have no security features.
- Authentication: indicates that only authenticated users will be able to run it.
- Authorization: indicates that, in addition to being authenticated, users will have to be authorized to run this object. That is to say, they must have the corresponding role to run it.

GAM Integration

The screenshot shows the GeneXus KB Explorer interface. On the left, the 'EventDay' project is expanded to show the 'GAM_Library' folder. In the center, a dialog box titled 'Enable Integrated Security' is open. The dialog contains the following text:

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.

Additional Components

- Web backend panels and Web security objects. If you select this option, a new theme will be imported also.
- Smart Devices security panels

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

Do you want to Enable Integrated Security?

Yes No

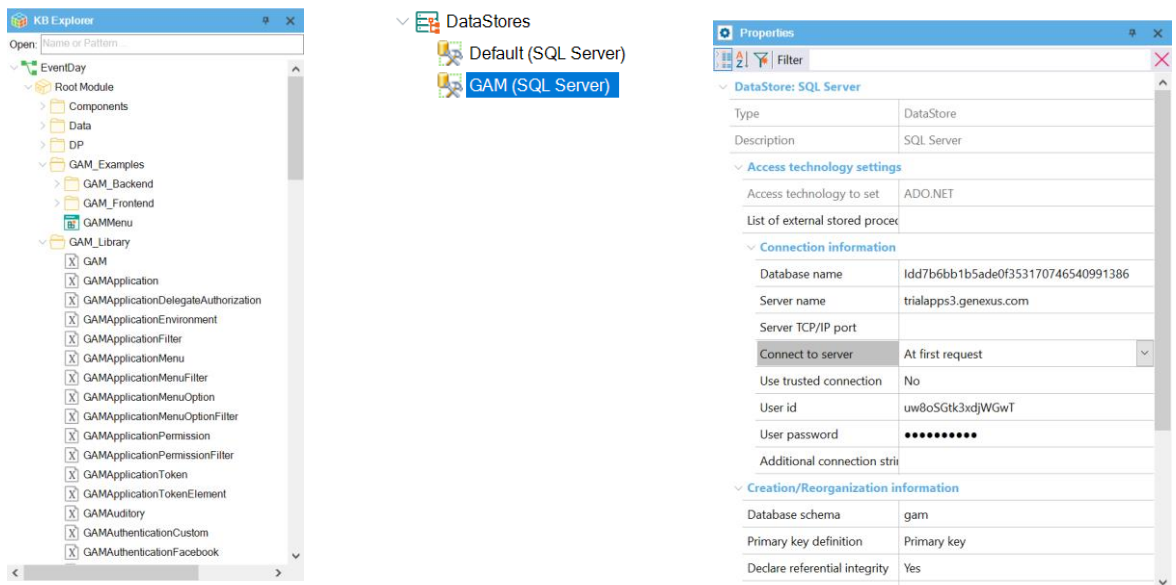
On the right, the 'Properties' window for a 'DataStore: SQL Server' is visible. It shows the following configuration:

DataStore: SQL Server	
Type	DataStore
Description	SQL Server
Access technology settings	
Access technology to set	ADO.NET
List of external stored proc	
Connection information	
Database name	Idd7b6bb1b5ade0f353170746540991386
Server name	trialapps3.genexus.com
Server TCP/IP port	
Connect to server	At first request
Use trusted connection	No
User id	uw8oSGtk3xdjWGwT
User password	●●●●●●●●
Additional connection st	
Creation/Reorganization information	
Database schema	gam
Primary key definition	Primary key
Declare referential integrity	Yes

Once we have the security properties configured, GAM objects will be automatically imported into the KB and we will have to do a Rebuild All. When doing so, a dialog box will be opened informing that the GAM module will be installed in the KB, with the solution ready to run on the Web and on Smart Devices.

GAM is also prepared to run on a database that is independent of the application database. In this case, we will not have to worry about this structure because it has its own Schema and will be associated with an independent Data Store in the KB, so all the configuration is independent. In addition, GAM will be responsible for initializing and then keeping the entire database up to date.

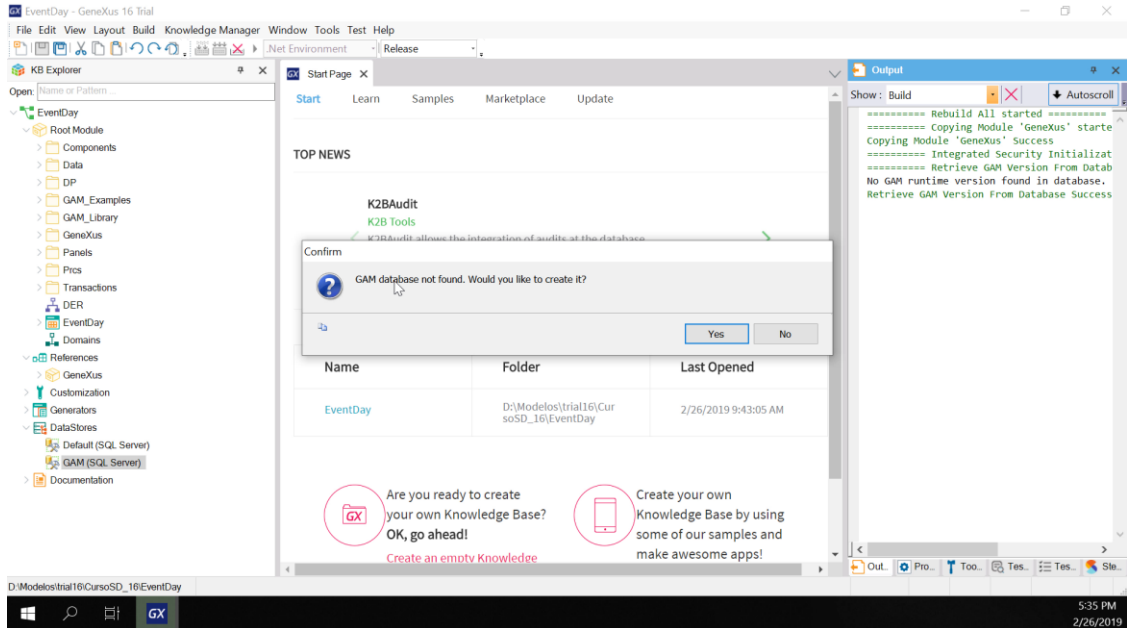
GAM Objects & Data Store



In the KB, we can already see that some folders were created in the root module. Gam_Examples are sample objects that we can modify. The back-end and front-end objects for web and SD will be here. Also, Gam_Library with the API; these are all external objects. We also have a new Data Store—GAM—with the information of that connection.

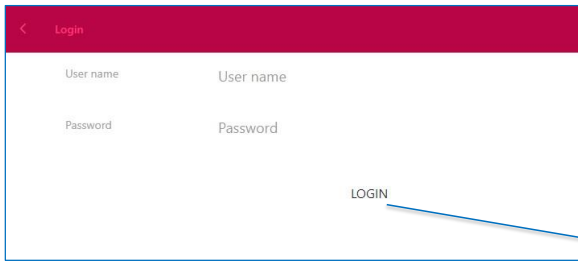
By default, the same base as the default Data Store is assumed but we have our own schema for the tables.

Creating GAM Database

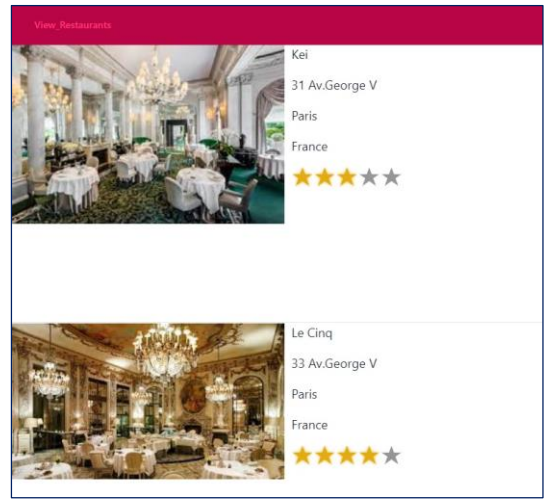


Now that the import process is finished, we do a Rebuild All of the application. GeneXus indicates that the GAM database was not found and asks if we want to create it. After we confirm, the database is created with all the tables and initialized.

When running it:



User: admin
Password: admin123



We run our View_Restaurants panel and the first thing we see is the login screen. If we want to enter without user credentials we get an error. All this is provided automatically by GAM.

To log in, we use a username that is created by default: "admin." The password is "admin123." Then the application opens. We don't save these credentials, and the application keeps on working properly.

All ▾

[Recents](#) [GAM Authentication](#) [Search](#) [Wiki Home](#) [GAM platforms](#) [HowTo: GAM](#)[Other document versions ▾](#)**GENEXUS ACCESS MANAGER (GAM)** ▾

- **GAM Built-in Security Module**
 - Getting Started
- **Authentication and Authorization**
- **GAM services**
- **Repository features**
- **GAM in Mobile**
- **GAM deployment**
- **Advanced**
- **Compatibility**
- **Media**
- **Hardening of GeneXus Systems and Deployments with GAM**

< GeneXus Access Manager

This documentation is valid for:

[GeneXus 15 Help](#) [GeneXus 16 Help](#) [GeneXus 17 Help](#)



The majority of modern applications need some scheme of authentication/authorization. To cover these aspects, GeneXus provides a mechanism (called GeneXus Access Manager) to offer a single, centralized scheme with everything related to application authentication and authorization.

The GeneXus Access Manager (GAM) provides APIs to manage all the security issues concerning an application. Therefore, the security module of any application (web applications and mobile applications) is provided by GAM. Also, security controls are automatically performed by configuring [Enable Integrated Security property](#).

For more information about GAM, visit the [GeneXus Access Manager page](#) on the wiki.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications