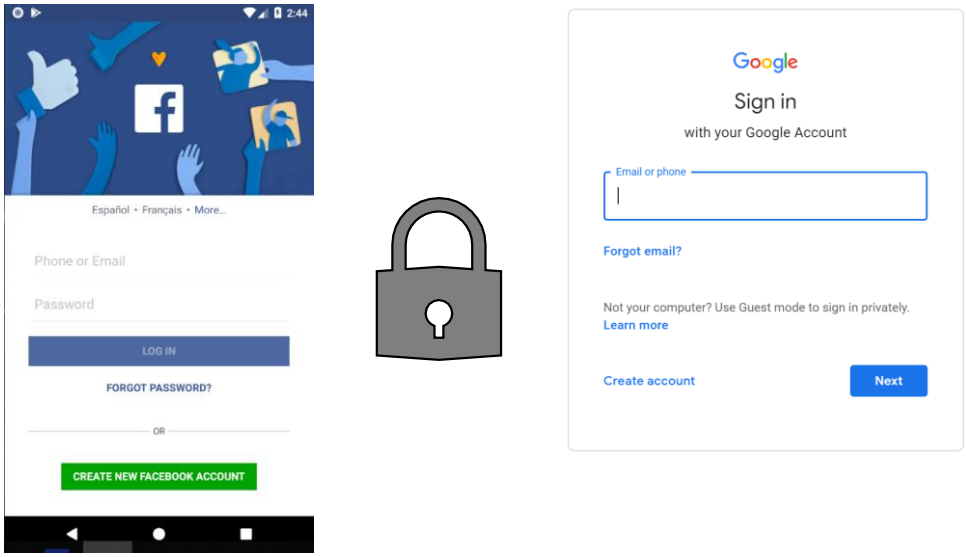
Segurança nas aplicações Angular

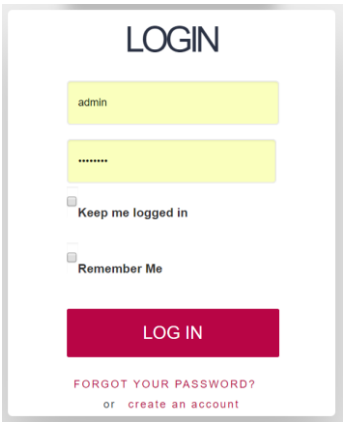
GeneXus™

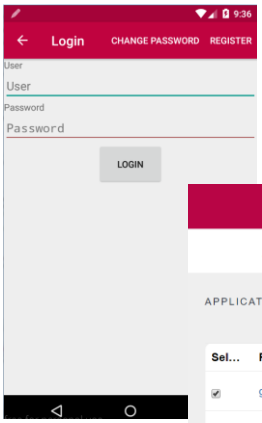
Security



Como já sabemos, a grande maioria das aplicações modernas precisa de um esquema de segurança, para que possam entrar apenas os usuários permitidos e também autorizar ou restringir o acesso a partes da aplicação, de acordo com as permissões atribuídas ao usuário.

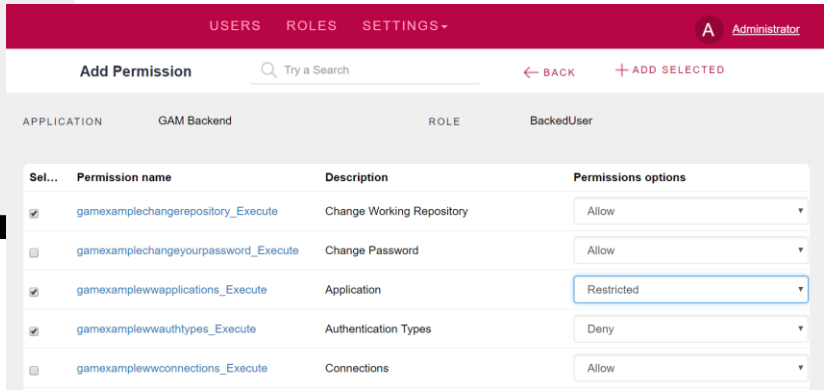
Security





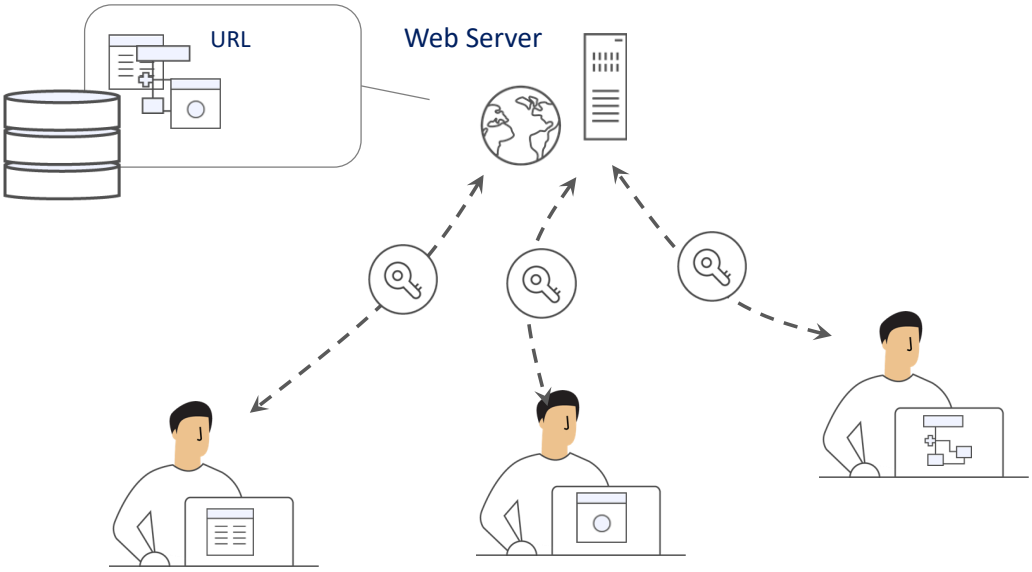
Authentication

Authorization



Isto significa garantir que todos os usuários que entrem, estejam devidamente autenticados (ou seja, que o usuário seja quem diz ser); e autorizados (ou seja, uma vez que o usuário é autenticado, seja permitido ou não o acesso a determinadas partes da aplicação).

Security in Web Applications



No caso das aplicações Web, como estas aplicações possuem vários pontos de entrada, qualquer objeto acessível por URL deve verificar permissões de autenticação.

Isso implica que cada um destes objetos deve ter incorporada a verificação de segurança para realizar a verificação correspondente.

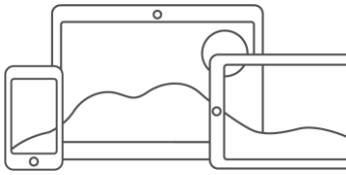
Integrated Security Solution

GAM

GeneXus Access Manager

Authentication

Authorization



Para atender a estas necessidades, GeneXus oferece um módulo de segurança, chamado GeneXus Access Manager (GAM), que resolve as funcionalidades de autenticação e autorização, tanto para aplicações Web como para aplicações para Smart Devices.

O GAM é desenvolvido em GeneXus, por isso se integra facilmente à KB da aplicação e permite resolver de forma centralizada tudo que é referente à Segurança da mesma. O objetivo é que a solução de Segurança seja utiliza da forma mais declarativa possível dentro da aplicação, sem criar complexidade adicional.

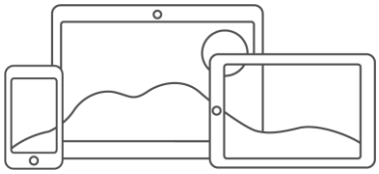
O GAM também oferece um back-end que permite definir usuários, permissões, políticas de segurança e acesso a objetos, entre outras coisas.

Além disso, fornece uma API para acessar muitas destas funcionalidades de forma programática.

GAM Features

Authentication

Web Sessions



Oauth

Authorization

RBAC

Role Based Access Control

Para resolver a Autenticação, internamente se usa:
- Web sessions para a segurança de aplicações Web
- Oauth para resolver a segurança no caso de aplicações para SD

No caso da Autorização, sua implementação é baseada em Roles utilizando o modelo Role Based Access Control, através do qual são encapsulados os métodos, propriedades e tudo o que for necessário para o gerenciamento de autorização na aplicação.

GAM Features

Local / Remote Authentication

External Identity Providers

Legacy / Custom Providers

External Web Service (SOAP)

Custom

O GAM fornece diferentes Tipos de Autenticação, os tipos disponíveis são:

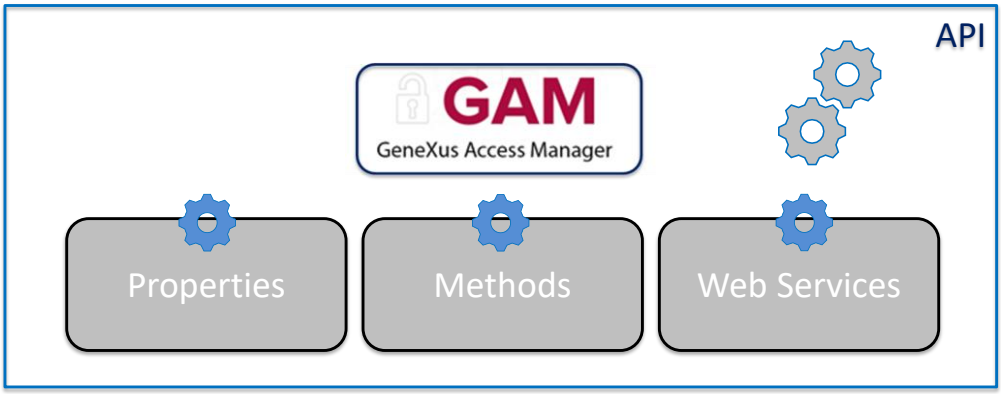
Autenticação local usando GAM onde os usuários e todas as suas credenciais são armazenados em uma base de dados da qual somos proprietários ou também de forma Remota, pois uma aplicação que usa GAM pode se tornar um provedor de identidades e, neste caso, outras aplicações com GAM podem se conectar remotamente a este server e obter autenticação a partir dali.

Podemos utilizar também outros provedores de identidade externos, estes, fornecem uma autenticação baseada no protocolo Oauth 2.0, como Facebook, Twitter e Google, Instagram, Office 365, Mercado Livre ou Linkedin, aqui são utilizados os mecanismos de autenticação padrão baseados neste protocolo implementado por estas aplicações. Neste caso não há necessidade de definir usuários locais.

Em muitas ocasiões é necessário integrar nossa aplicação com outras, com as quais temos que trocar informação e é necessário garantir a autenticação dos usuários através de uma autenticação externa à aplicação.
Uma forma de autenticação externa é utilizar um web service SOAP fornecido pela outra aplicação e configurar o GAM para que consuma esse web service.

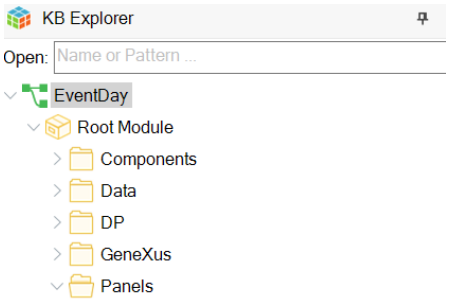
Pode ser possível que a outra aplicação forneça um programa externo para resolver a autenticação, mas que não é necessariamente um web service. Nesse caso configuro o GAM para aceitar uma autenticação do tipo Custom.

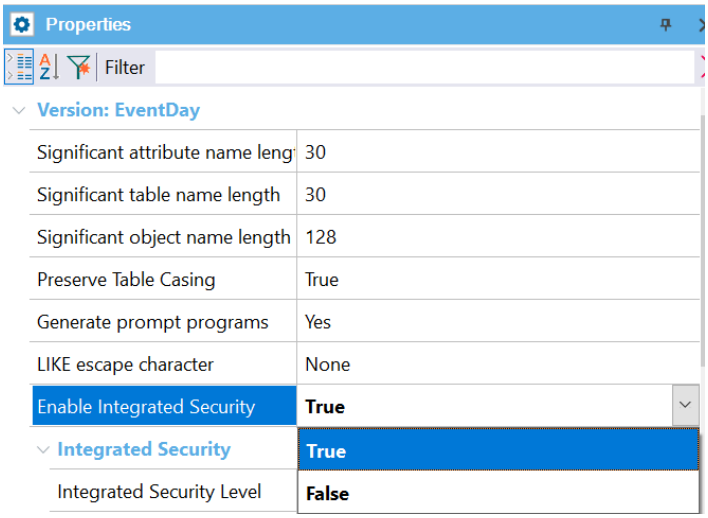
GAM Features



O GAM também expõe uma API (Application Program Interface) para acessar suas propriedades e métodos caso seja necessário fazê-lo a partir de nossa aplicação e uma série de serviços Web que podem ser utilizados a partir de outras aplicações.

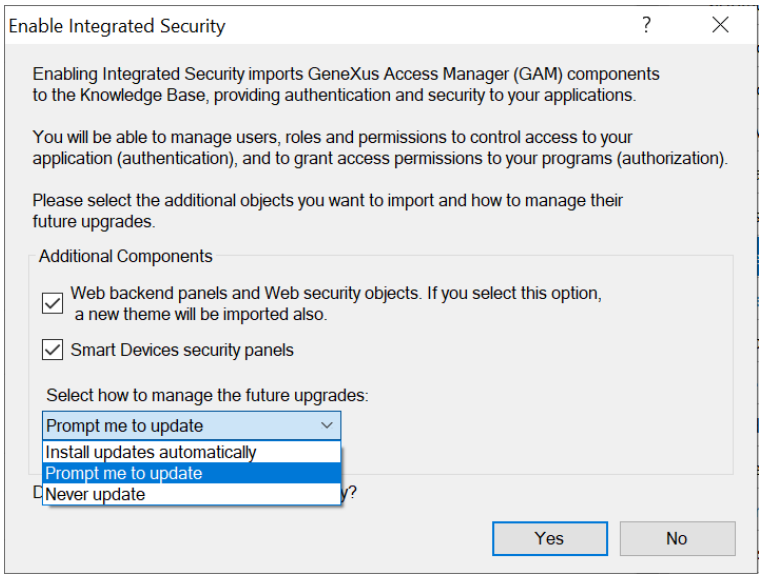
Enable Integrated Security





Para habilitar o GAM, deve-se ir ao nível da versão ativa da KB e configurar a propriedade Enable Integrated Security com o valor True. Na versão Trial encontra-se no primeiro nó do KB Explorer com o nome da KB.

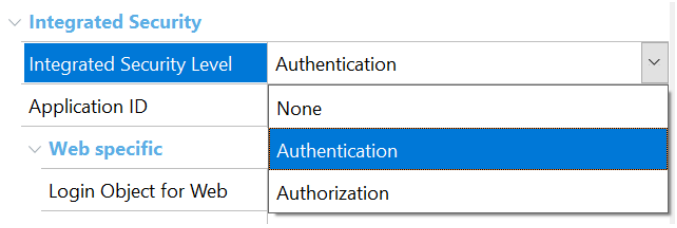
Enable Integrated Security

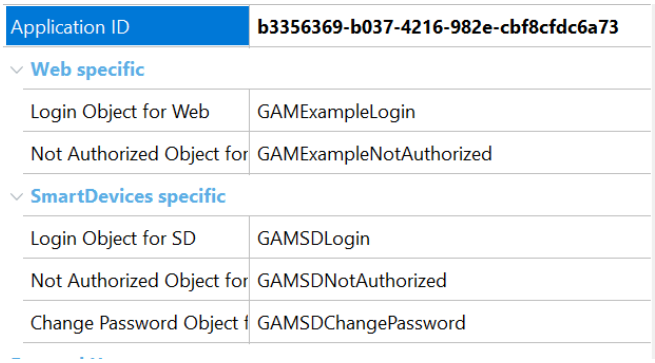


Este diálogo nos indica que irá prosseguir com a integração. Aqui podemos indicar se queremos que seja integrado o back-end web e com este outro se queremos que seja integrada a segurança para os painéis de Smart Devices.

Com este combo podemos escolher como queremos que este módulo seja atualizado, pode ser automático, podemos escolher que nos pergunte ou que nunca seja atualizado. pressionamos Yes.

Integrated Security Level



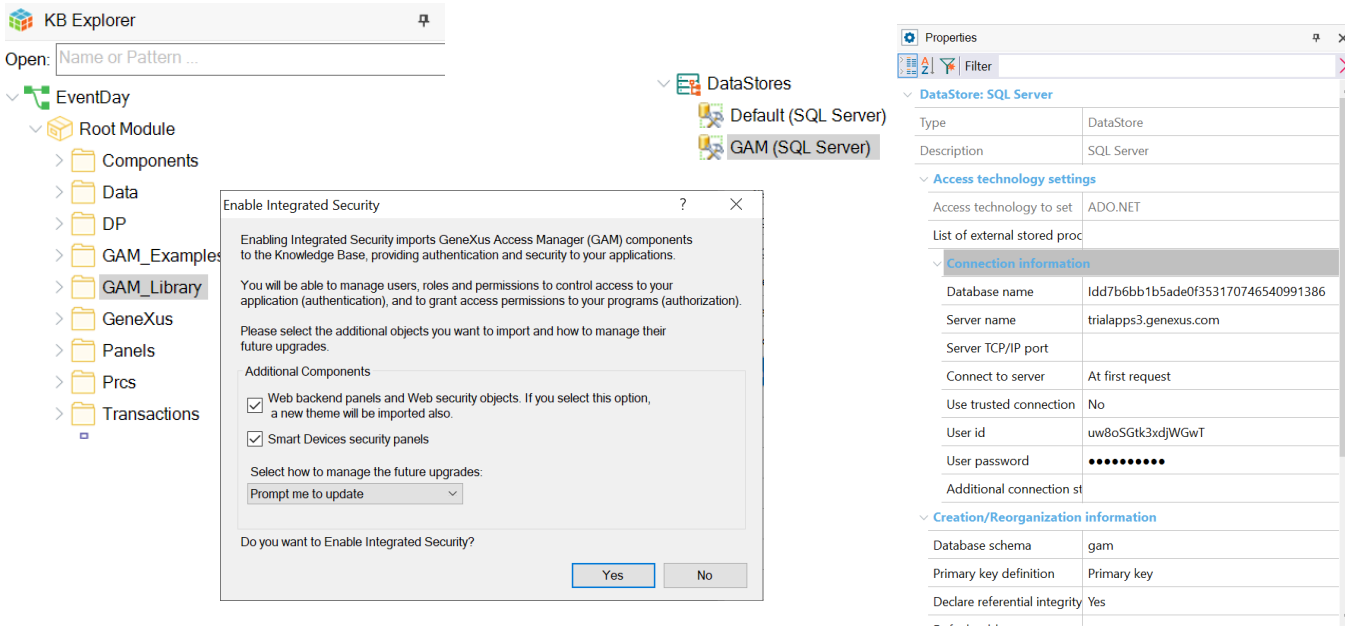


Uma vez que tenhamos habilitado o GAM, veremos outra propriedade chamada Integrated Security Level que nos permite indicar o valor padrão da segurança dos objetos da KB. Esta propriedade também se encontra no nível do objeto, pelo que será possível personalizar a forma como será implementada a segurança nesse objeto.

Existem três valores possíveis:

- None: indica que o objeto será público, ou seja, não terá segurança.
- Authentication: indica que somente usuários autenticados poderão executá-lo.
- Authorization: indica que o usuário, além de ter se autenticado, deverá estar autorizado a executar o referido objeto, ou seja, ter a função adequada para executá-lo.

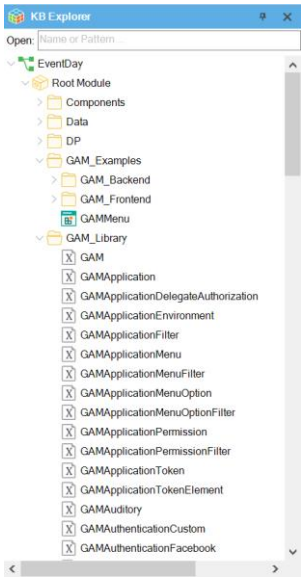
GAM Integration

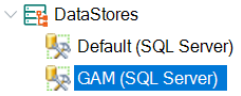


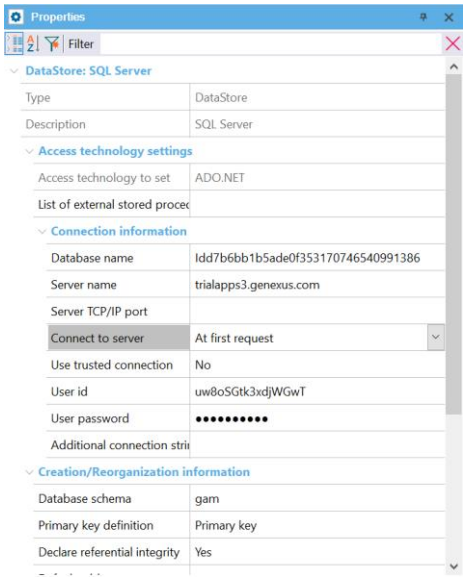
Uma vez que tenhamos as propriedades de segurança configuradas, serão importados de forma automática os objetos do GAM para a KB e então teremos que fazer um Rebuild All da mesma. Ao fazê-lo será aberta uma caixa de diálogo que nos avisa que será instalado o módulo GAM em nossa KB, com a solução pronta tanto para web como para Smart Devices.

O GAM também está preparado para executar em uma base de dados independente da base de dados da aplicação se assim o desejarmos, não teremos que nos preocupar com esta estrutura nesse caso pois conta com um Schema próprio e estará associado a um Data Store Independente na KB, com o qual toda a configuração é independente. Além disso, o GAM se encarregará de inicializar e então manter toda a base de dados atualizada.

GAM Objects & Data Store



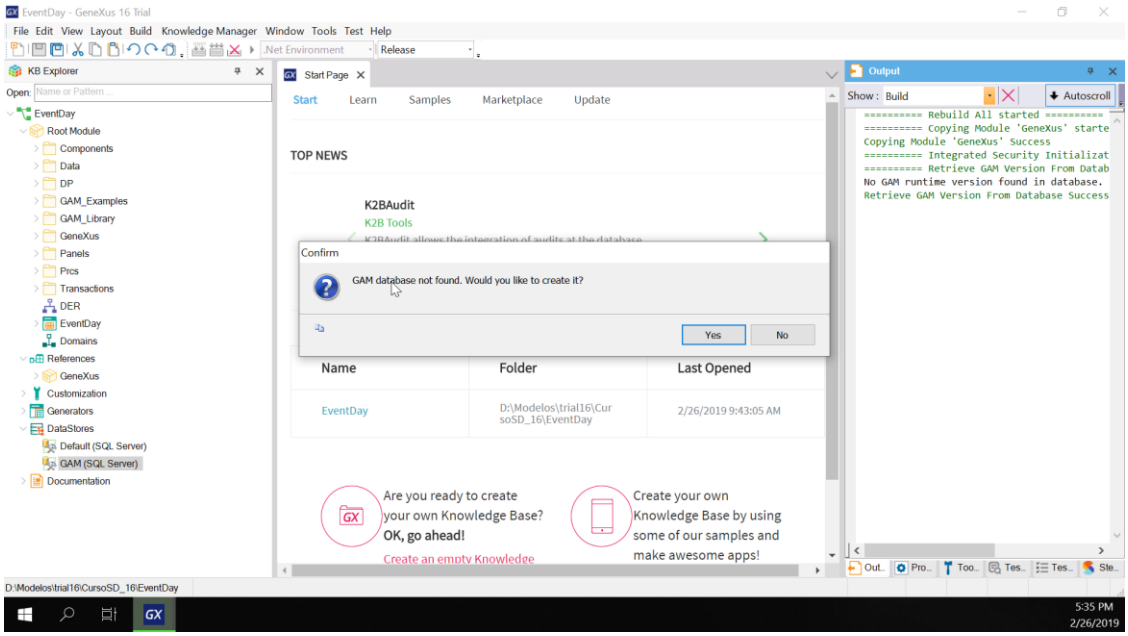




Na KB já podemos ver que foram criadas algumas pastas no root module. Gam_Examples que são objetos de exemplo que podemos modificar. Aqui estarão os objetos do back-end e do front-end tanto para web como para SD. E Gam_Library com a API, estes são todos objetos externos. Também temos um novo Data Store, GAM, com as informações dessa conexão.

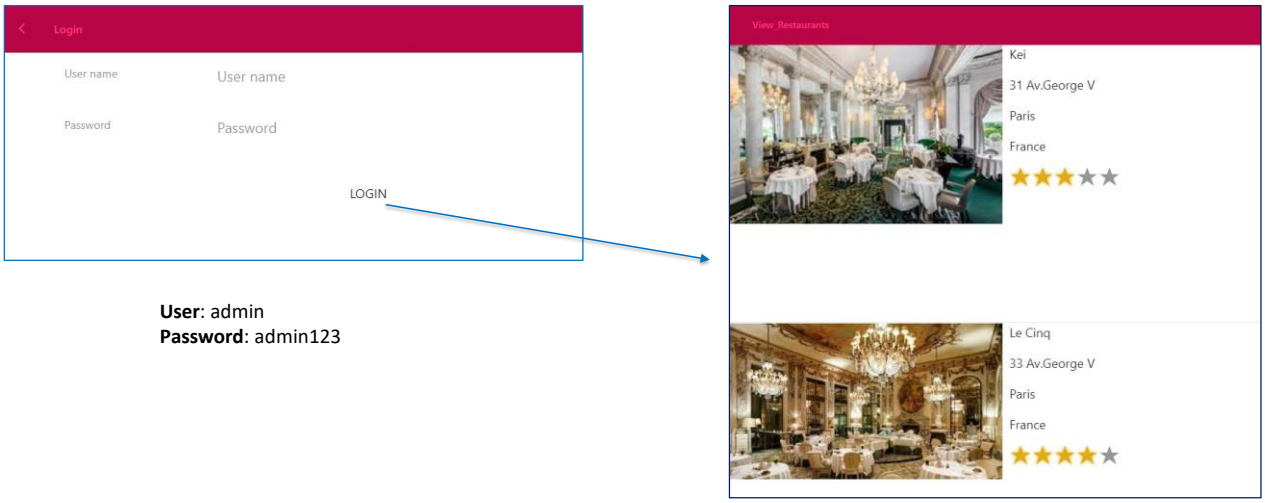
Por padrão, assume-se a mesma base que o Data Store default, mas temos um schema próprio para as tabelas.

Creating GAM Database



Bem, como terminou o processo de importação, vamos fazer um Rebuild All da aplicação. GeneXus nos indica que a base de dados do GAM não foi encontrada e se queremos criá-la. vamos dizer sim. bem, aí é criada a base de dados com todas as tabelas e então é inicializada a base de dados.

Ao executar:



User: admin
Password: admin123

Executamos nosso panel View_Restaurants e a primeira coisa que vemos é a tela de login. E se quisermos entrar sem usuário, dá um erro. Tudo isso nos fornece o GAM de forma automática.

Para entrar usaremos um usuário que é criado por padrão, “admin”, a senha é “admin123”. E aí se abre nossa aplicação, não vamos salvar as credenciais. E a aplicação continua funcionando normalmente.

Para mais informações sobre GAM, visite a página do GeneXus Access Manager na wiki.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications