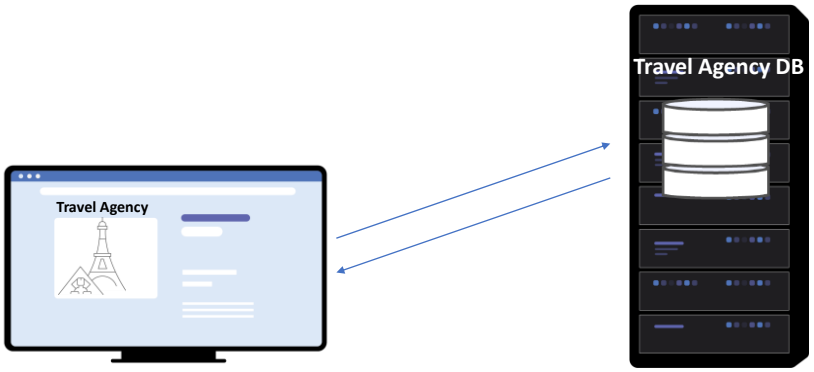
How to Access External Data

Introduction

GeneXus™

GeneXus

Application created with GeneXus

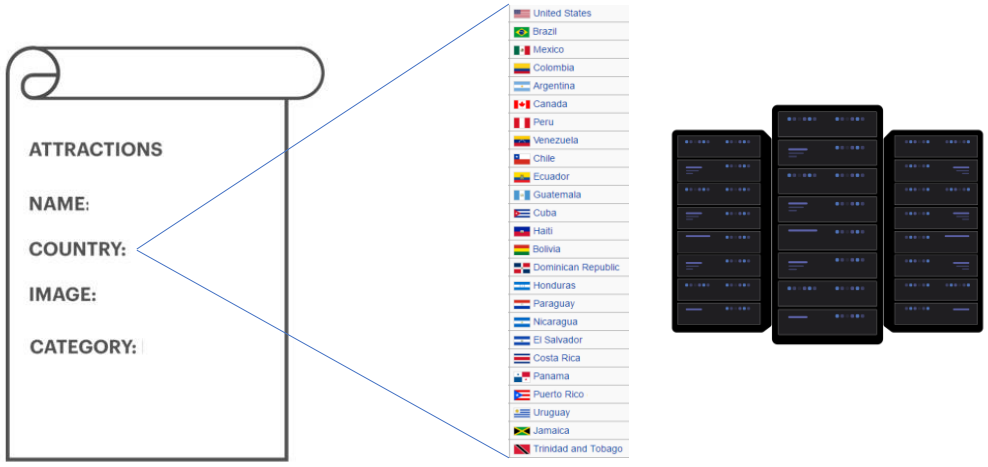


When we develop an application with GeneXus, the database is automatically created, where the main information handled by the application is stored.

However, many times we need to access other data sources for different purposes.

GeneXus

Example of use of a list of countries from an external source

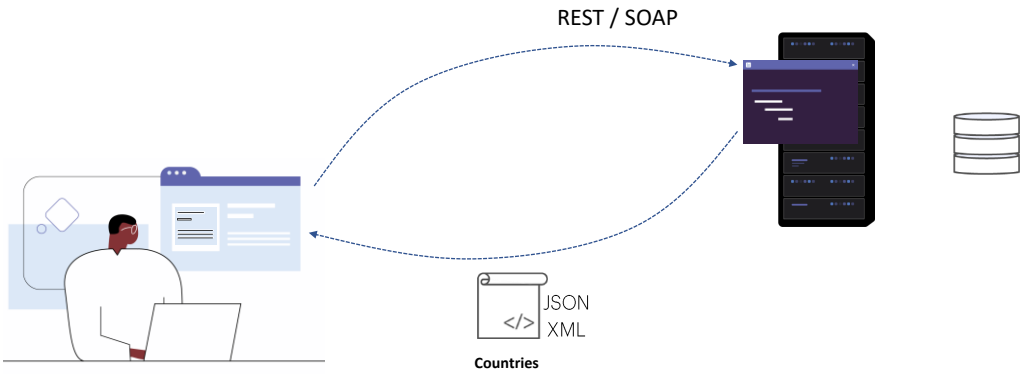


For example, suppose we need to choose a country from a list of countries, to assign it to some entity. In this case, it is more convenient to access a repository where we can obtain the entire list of countries, instead of entering them one by one through the application screens.

In addition, this list will probably be kept up to date and save us from having to keep the data up to date.

GeneXus

Obtaining a list of countries through a web service



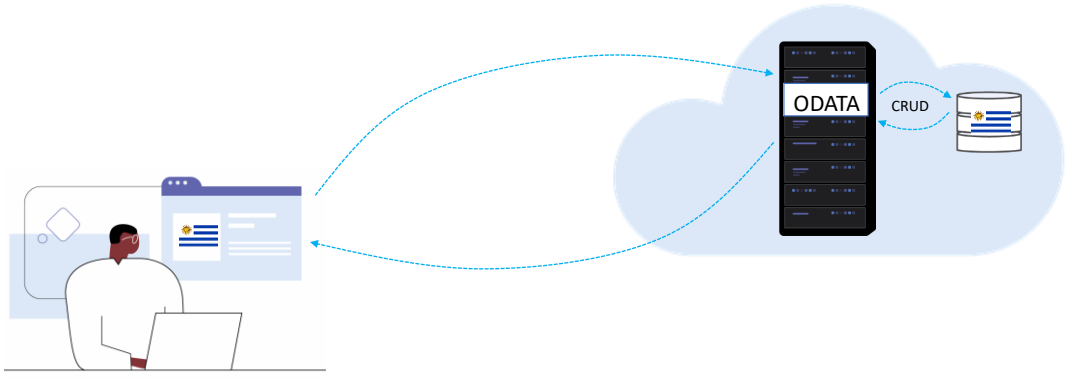
With GeneXus, we can also expose a webservice using the API object

One solution for this is to connect to a web service that publishes the list of countries as a service. Consuming that resource we can obtain a file with the data in structured form, for example, JSON or XML.

This data is used in our application, either to populate an in-memory structure or a table. We will update this information from time to time.

GeneXus

Accessing a remote database with ODATA protocol



Otra posibilidad es conectarnos a servidores web que provean servicios con protocolos que nos permite realizar operaciones de lectura o incluso escritura sobre los datos almacenados en una base de datos remota.

GeneXus cuenta con un wizard que crea todos los objetos GeneXus necesarios para conectarnos a estos servicios, y con esta opción siempre estaremos accediendo a datos actualizados.

GeneXus

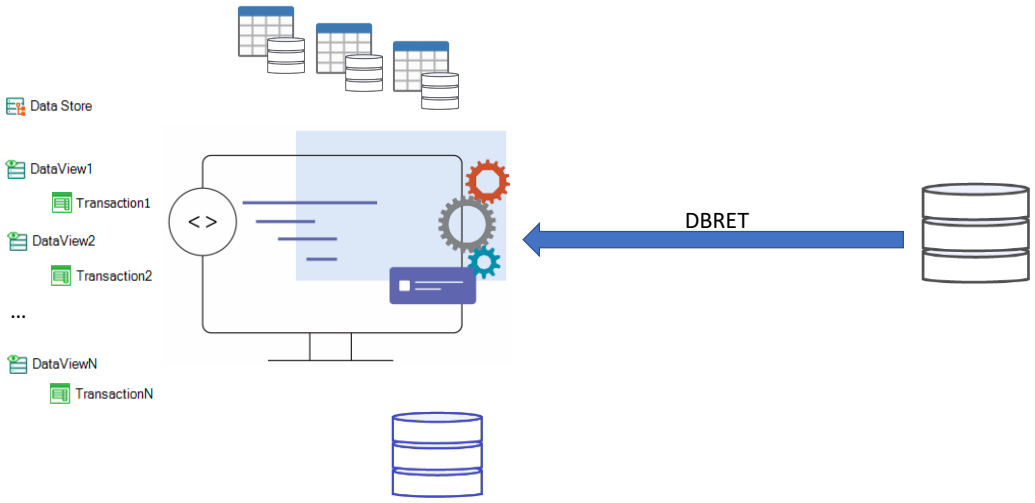
Acceso a una base de datos de un sistema “legacy”

Algo que es común que pase cuando desarrollamos una aplicación para una empresa, es que la empresa ya cuente con un sistema anteriormente creado, cuyos datos deben reutilizarse, ya sea porque ambos sistemas deben coexistir o porque debemos migrar los datos del sistema viejo al nuevo.

Este sistema anterior, al que llamamos “heredado” (“legacy”), probablemente tiene su propia base de datos y es posible que la información que debamos utilizar, deba almacenarse y mantenerse exclusivamente desde dicha plataforma. O bien que también podamos modificar esos datos desde nuestra aplicación, por lo que debemos tener acceso a la base de datos y sus estructuras.

GeneXus

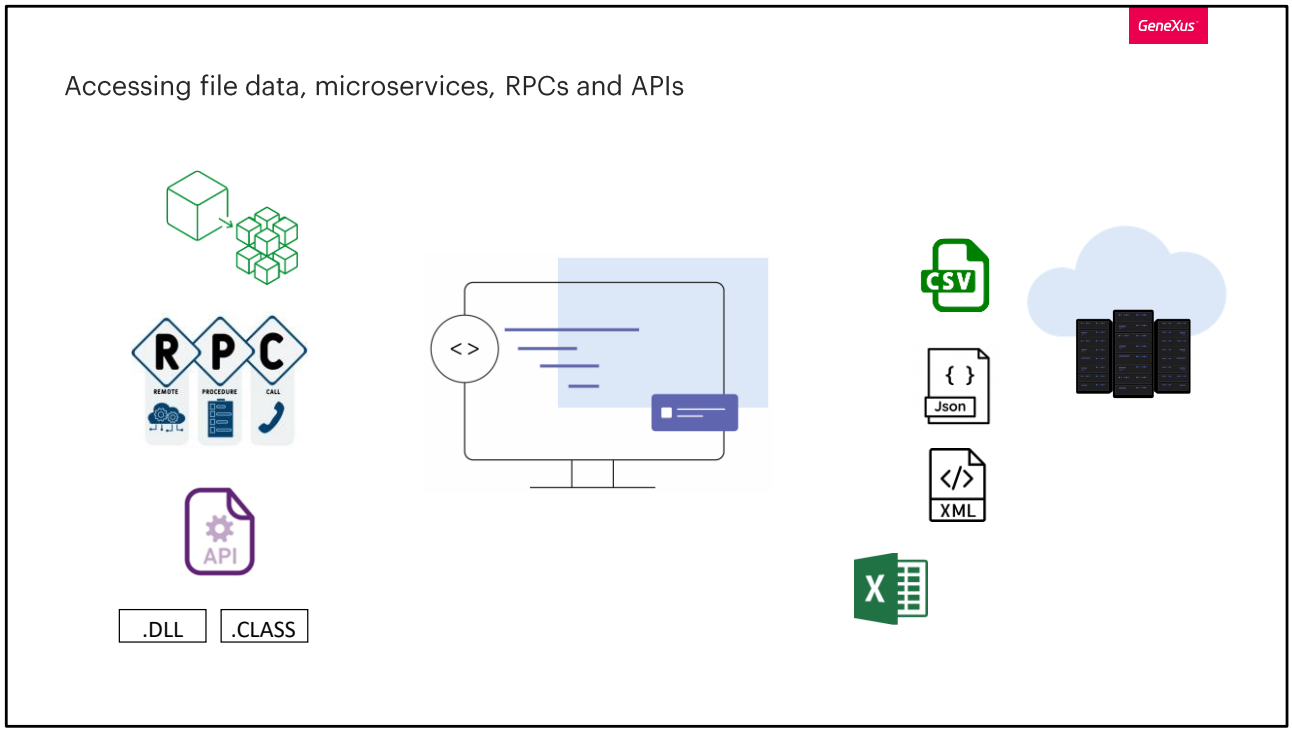
Using reverse engineering to connect to a database



GeneXus has a wizard that allows connecting to external databases to our application and from its tables and indexes, applying reverse engineering, creates the necessary GeneXus objects associated with these structures.

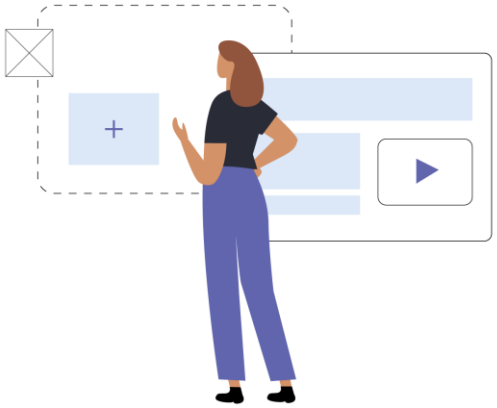
With this mechanism, a Data Store object will be automatically created in the default environment, which will contain the necessary database connection credentials and configuration data. Also, Data View objects connected to the external database tables will be created and, if we want, the associated transactions will also be automatically created, which will allow us to add, delete, modify or access the information, as if it were from our own database.

This type of solution can also be applied to connect to individual tables in different databases, creating the Data View objects ourselves or using a wizard to import a particular external data store.



In addition, as with most common programming languages, with GeneXus we can also access individual data files, stored in an inbox or in remote resources (on premise or in the cloud), as text files (.txt), with comma-separated data (.csv), or in structured formats (such as .json or xml files), or even data stored in Excel spreadsheets.

GeneXus also allows us to connect to other programs that provide us with external data, such as interacting with microservices, invoking remote procedures from other applications, or accessing APIs with functions published in binary objects such as .DLLs created in C, C++, C Sharp or .NET or functions within a Java .CLASS file.



In the following videos, we will see in more detail how to use some of these methods to access external data and the references to the information needed to use those mechanisms that are beyond the scope of this course.

GeneXus™

training.genexus.com
wiki.genexus.com