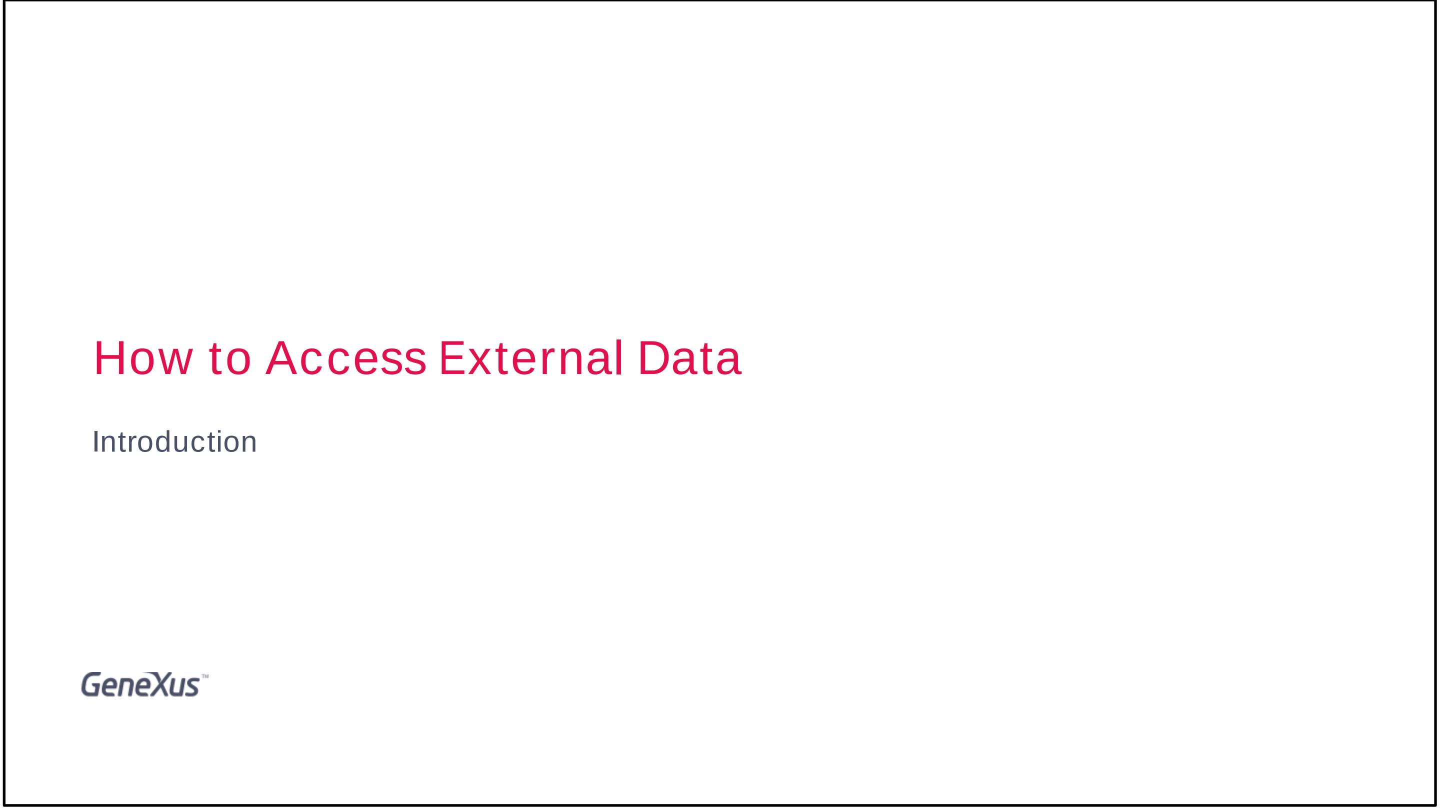
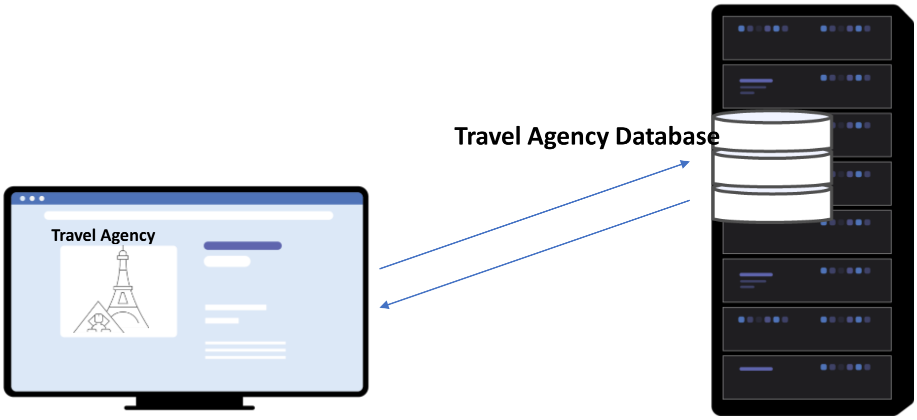


How to Access External Data

Introduction

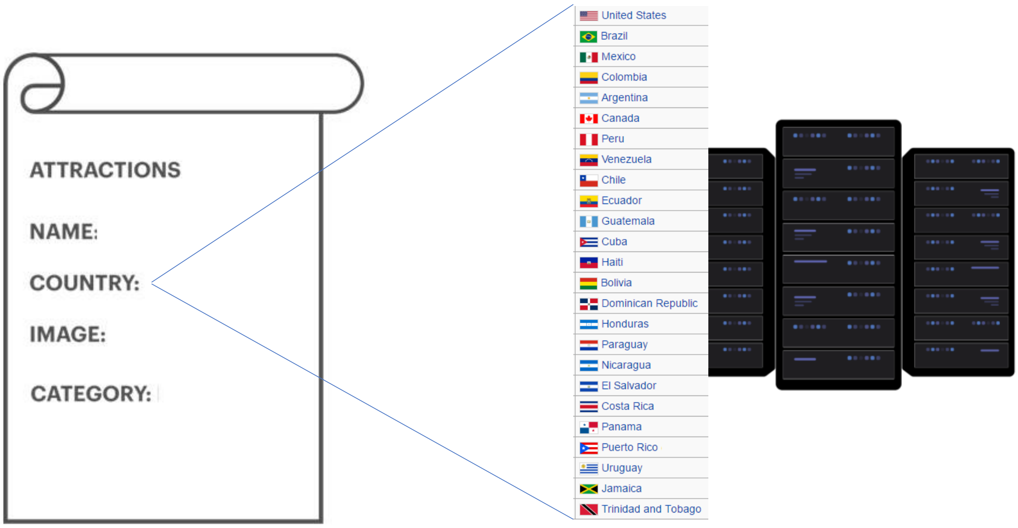
Application created with GeneXus



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.

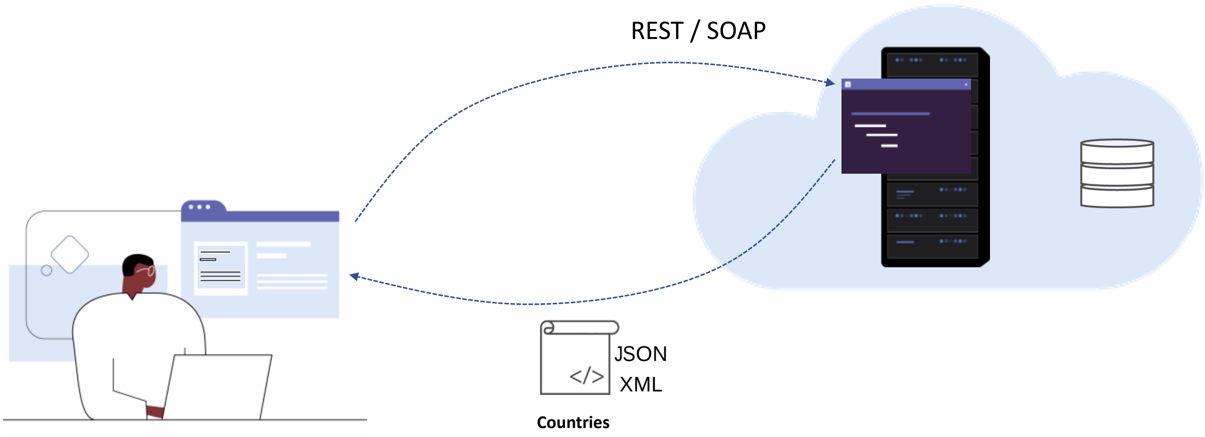
Example of use of a list of countries from an external source



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.

Obtaining a list of countries through a web service

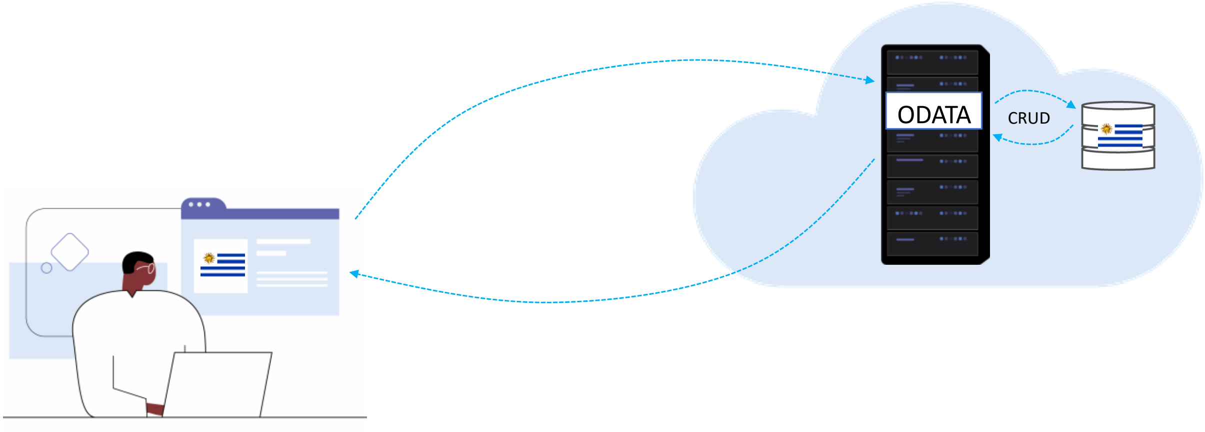


With GeneXus, we can also expose a web service using the API object

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.

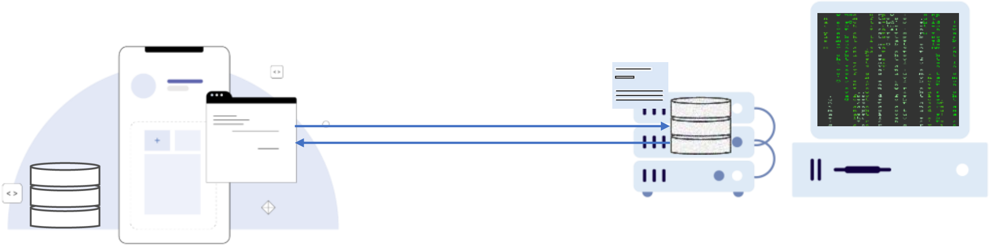
Accessing a remote database with ODATA protocol



Another possibility is to connect to web servers that provide services with protocols that allow us to perform read or even write operations on data stored in a remote database.

GeneXus has a wizard that creates all the GeneXus objects necessary to connect to these services, and with this option we will always be accessing updated data.

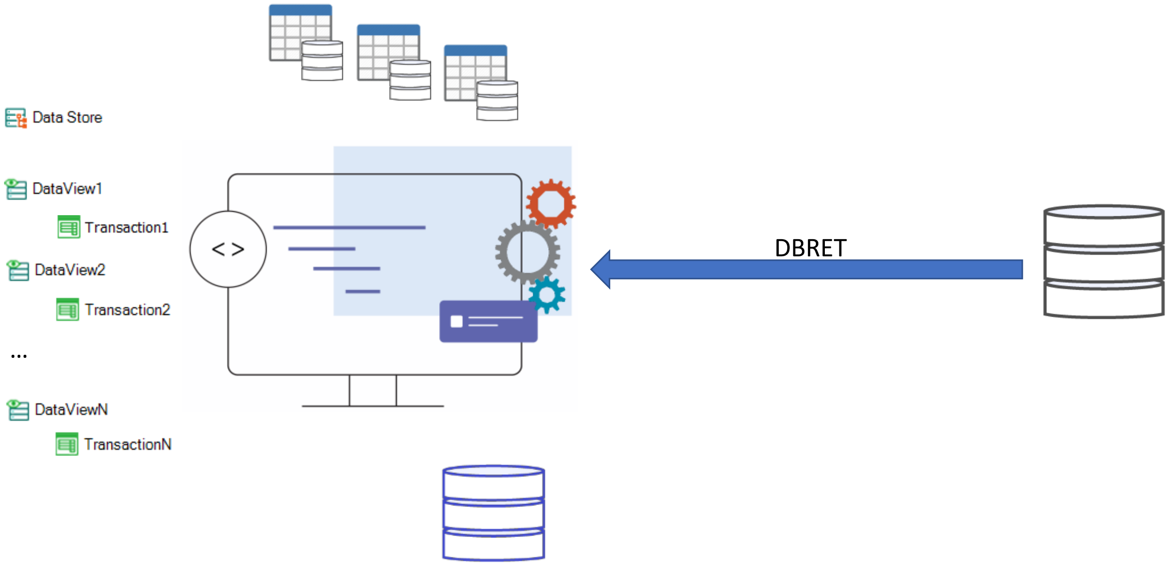
Accessing a database in a “legacy” system



Something that frequently happens when developing an application for a company is that the company already has a previously created system, whose data must be reused, either because both systems must coexist or because we must migrate the data from the old system to the new one.

The previous system, which is called “legacy”, probably has its own database and it is possible that the information we need to use may have to be stored and maintained exclusively from that platform. Also, we may be able to modify that data from our application, so we must have access to the database and its structures.

Using reverse engineering to connect to a database

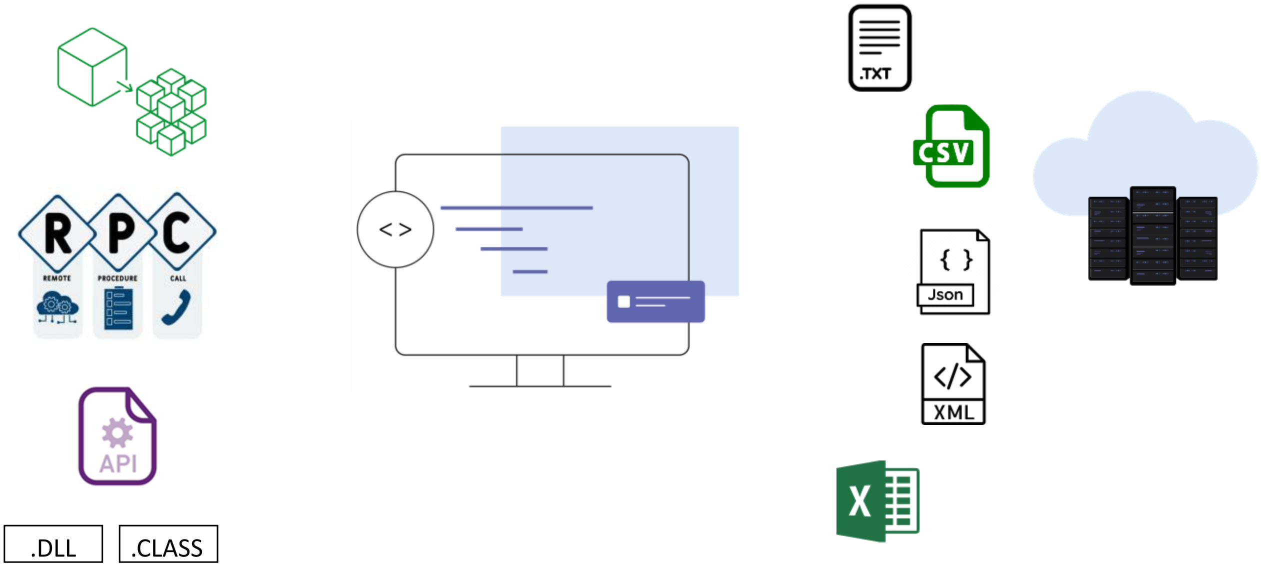


GeneXus has a wizard that allows connecting to external databases to our application and from its tables and indexes, applying reverse engineering, creates the necessary GeneXus objects associated with these structures.

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.

Accessing file data, microservices, RPCs and APIs



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