# Prototyping and Deployment of an Angular Application

GeneXus™

Here you will see different ways to prototype an Angular application and how to deploy it, that is to say, the commissioning for it to function on the client's facilities.

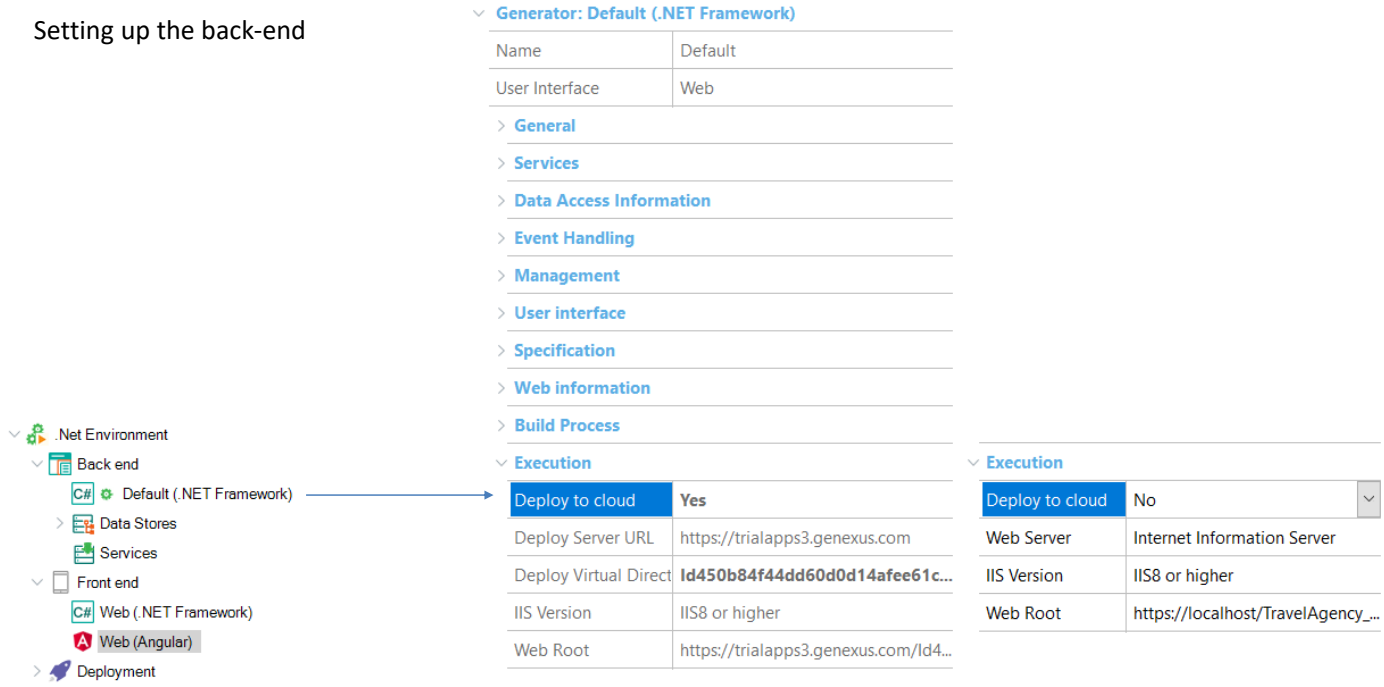
# Prototyping an Angular App

The stage where you start building the app and adjust its operation in accordance with the requirements to be fulfilled is called Prototyping.

Since an Angular app has a front-end and a back-end, now you will see the prototyping options available for each part.

Start by prototyping the back-end.

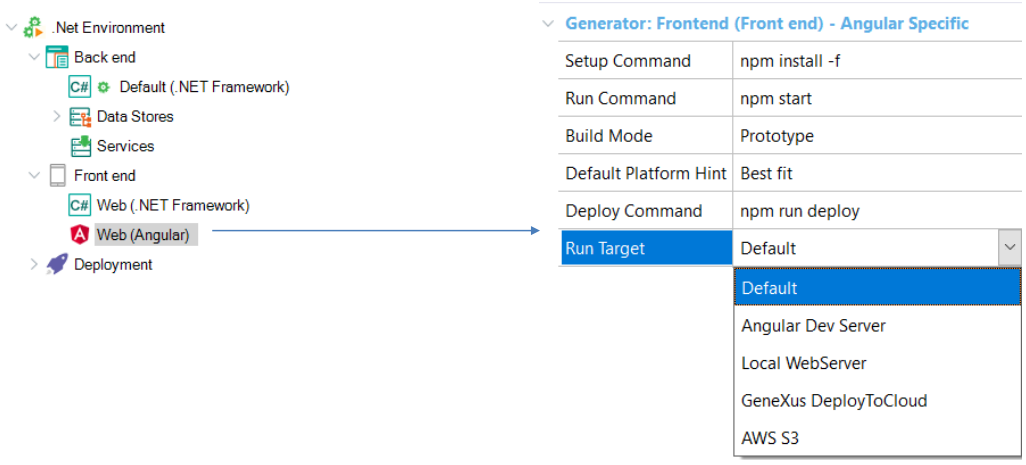
## Setting up the back-end




So far, you have used the default mechanisms for prototyping. Since you are using GeneXus Trial, the backend part is prototyped automatically in the GeneXus cloud and you cannot change the value of the Deploy to Cloud property.

With the full version you could select the value of that property and then prototype in the Cloud with value Yes, or also set the Deploy property to cloud as No, and prototype on a web server installed in our local computer, with the DBMS installed locally as well, in which case the access to data in the database would take place only on your computer.

## Setting up the front-end




You have seen that, for the front-end, a local server (localhost) is instanced, by default, at a port that is assigned automatically.

But you have other choices for prototyping.

If you go to the properties of the Angular generator (in the properties of the environment / Front end), you will see that the Run Target property is available. If you press the combo you will see several values possible:

**Default:** by default, the Angular app is executed on the Angular Dev Server, locally (localhost) and at a random port. In the future, upon selecting Deploy To Cloud in the Backend, the default option will behave as GeneXusDeployToCloud.

**Angular Dev Server:** is the Angular Http Server of the Angular Platform running locally on the development machine. This is a simple server designed for SPA (Single Page App) developers.

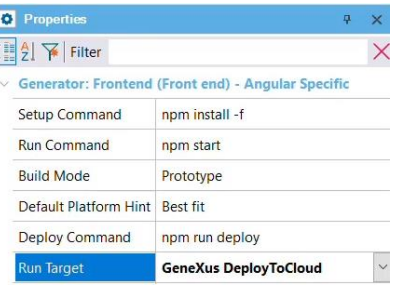
**Local Web Server:** the app is transpiled (which means translation from a source language into another source language: in this case from TypeScript into JavaScript) and copied to the local web server of the GeneXus Environment (for example, Tomcat in Java or Internet Information Server in .Net). The application is executed directly under the web backend URL.

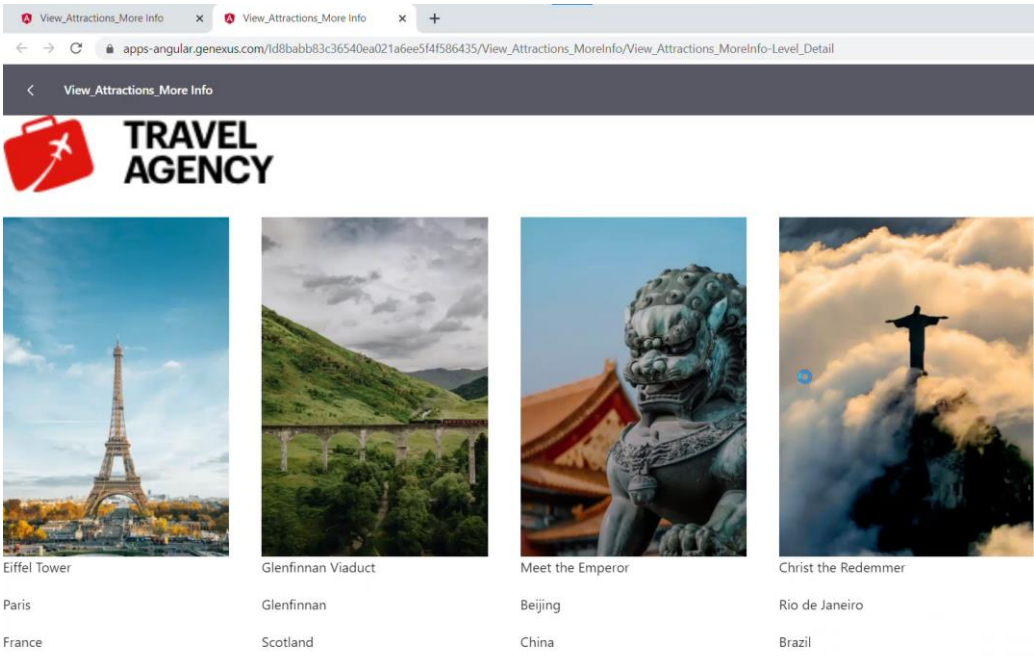
**GeneXusDeployToCloud:** indicates that the app will be uploaded to the cloud of DeployToCloud. In this case, the app will be executed on an AWS

S3 (Amazon Simple Storage Service) Server, which is a service by Amazon Web Services that provides storage for objects by means of a web service interface and is accessed through the content delivery network (CDN) of Amazon Cloudfront. This web service enables a more agile distribution to users of static and dynamic web content such as .html, .css, .js files and image files.

**AWS S3**: prepares the Angular project to be **manually** uploaded to AWS S3 via command line.

# Prototyping the front-end in the cloud




Now try to prototype the front-end in the cloud instead of on your local server.

Do that by changing the value of the Run target property, from the Default value you used to the GeneXus DeployToCloud value, and then execute the View_Attractions_MoreInfo panel.

Now note that no more local prototyping takes place in your computer (the url does not read localhost) but rather in the cloud. This turns it unnecessary to install the software to instance the local web server for Angular.

This option allows you to prototype the backend in the cloud and also prototype the front-end without the need to install any other source in the local computer, and making the setup of the development environment much easier.

# Deployment of an Angular App

Once you conclude development of the app on your computer, you must deploy it for operation on the client's facilities, either with an internal client (in our own organization) or with an external client. In this example, that would take place in the travel agency's servers.

Deployment Stages of an Angular App

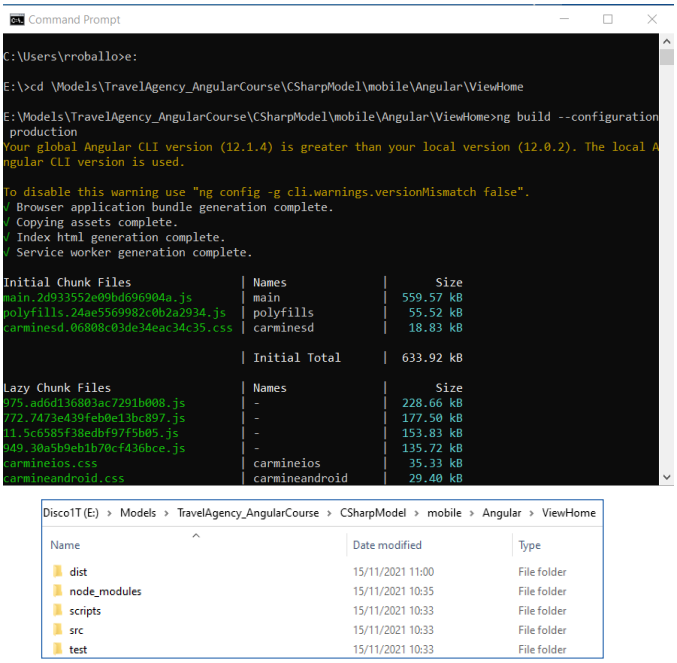

Deploying an app in Angular implies three stages:

- Distribution of the front-end objects to be deployed on the HTTP server.
- Distribution of the REST services to be deployed on the app server.
- Distribution of the database structures.

# Deploying the Front-end

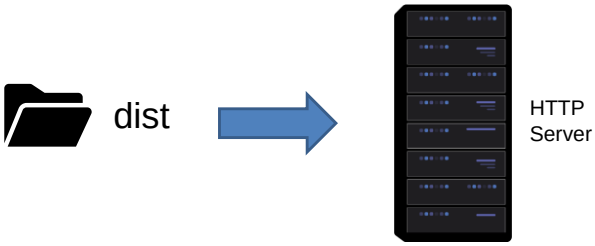
Start by the front-end

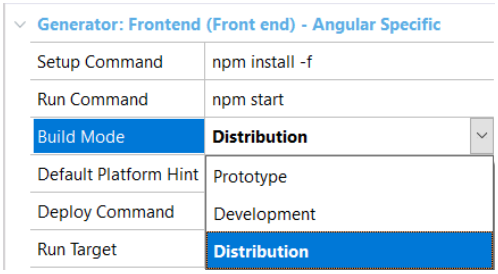
## Deploying the front-end










Deployment of the front-end objects is done through the Angular Command Line Interface.

Do it by opening a command window. Go to the KB folder that contains your main object (in this case, ViewHome) and write the following command: ng build –-configuration production. With this process you are creating, under the folder of the main object, a “dist” folder that has everything needed for the front-end.

Instead of writing a command line you can do this directly from the GeneXus IDE. If you go to the properties of the Angular generator you will see there the Build Mode property that allows the developer to set the mode in which the Angular app is to be executed.

The default value is Prototype, where the minimum dependencies required are imported into your project for executing the Angular app.

In the case of the Development mode, packages used in testing and debugging are also imported to aid in the development of the app.

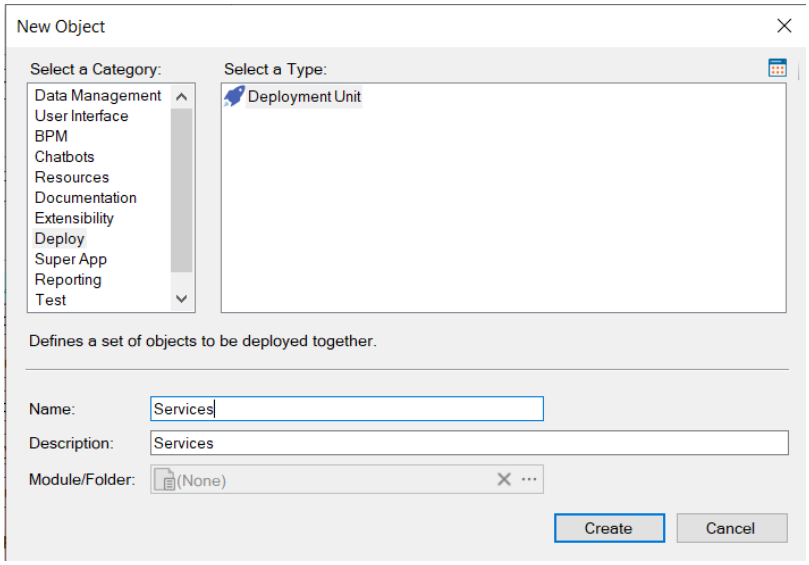
Use the Distribution mode if you want to generate the packages for deploying the app. Selecting this mode implies the automatic execution of the command ng build –-configuration production that you saw before.

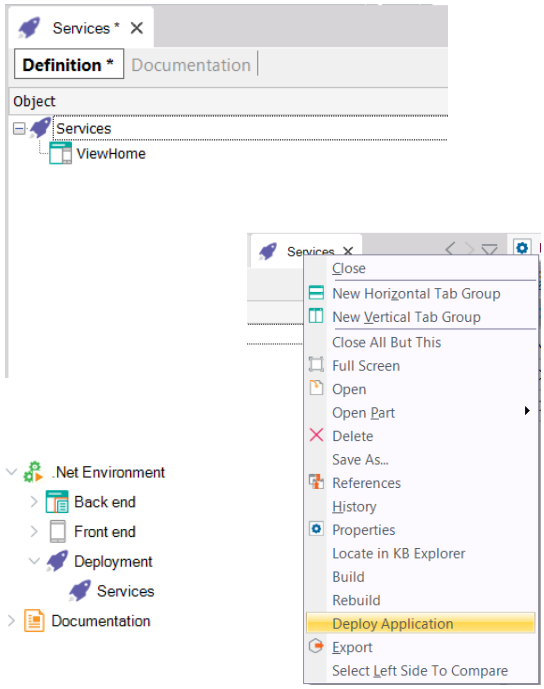
Following the distribution process and the generation of the “dist” file, the full file is copied to the client’s HTTP server (for example, Internet Information Services or Apache Tomcat).

# Deployment of REST Services

Now go to the deployment of the REST services through the main object.

# Deployment of REST services



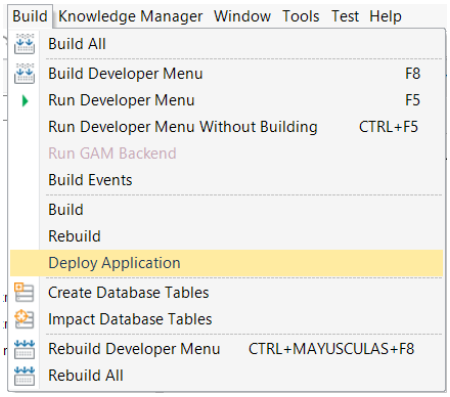



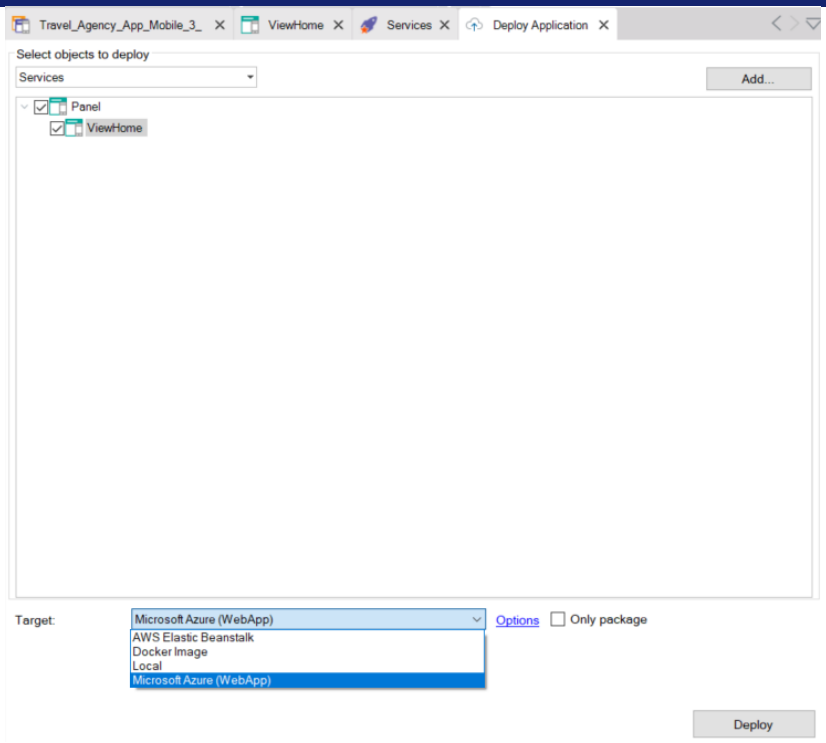

Deployment of the REST services is done by creating a Deployment Unit object that you will call, for example, Services, and then dragging the main object over the Services node.

In the KB Explorer you will see that the deployment unit you created is shown under the environment Deployment node.

To deploy right click on the deployment unit and select Deploy application.

## Deployment of REST services



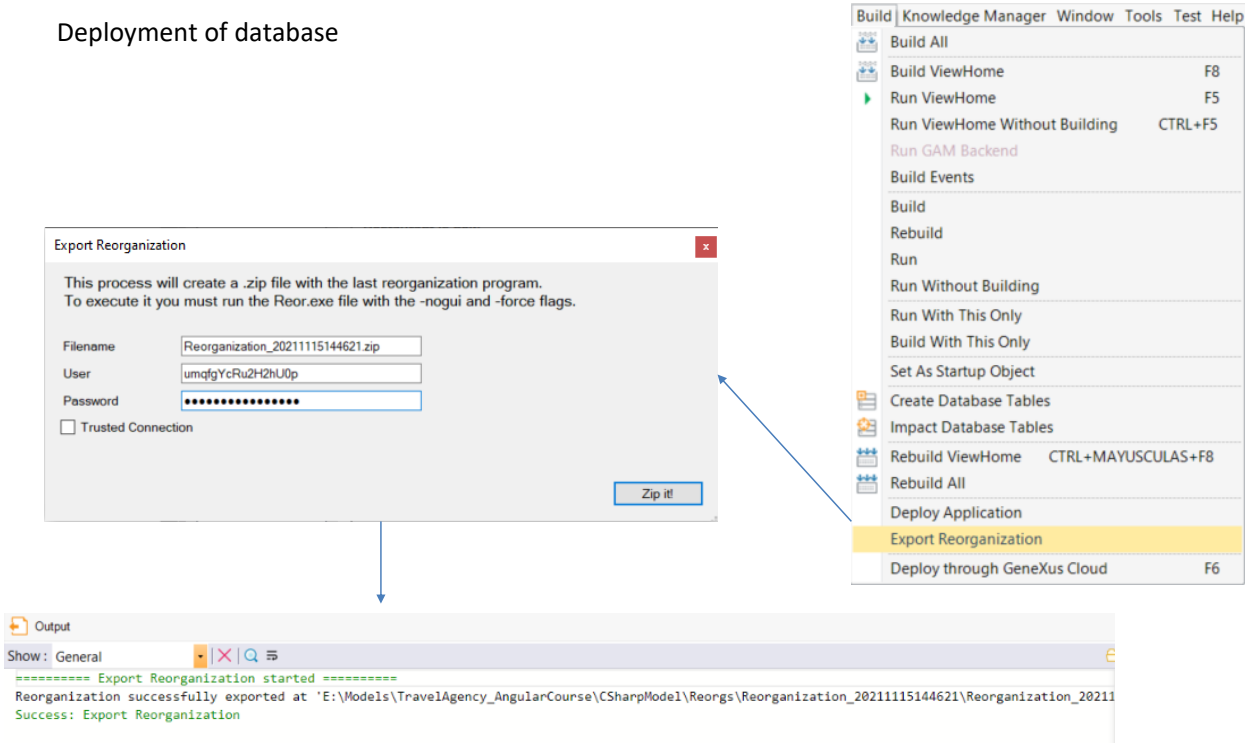

Another option is using the Application Deployment Tool that is accessed from the Build menu and selecting Deploy Application. You will see that the option deployment unit Services –that already contains your main object– is automatically selected.

Then, with the Target combo box, select the type of deployment that you will do (local, to a Docker image, or to an external server, such as Microsoft Windows Azure or AWS Elastic Beanstock), and then press the Deploy button.

# Deployment of database

The database is distributed by exporting the reorg to the production database server.

## Deployment of database



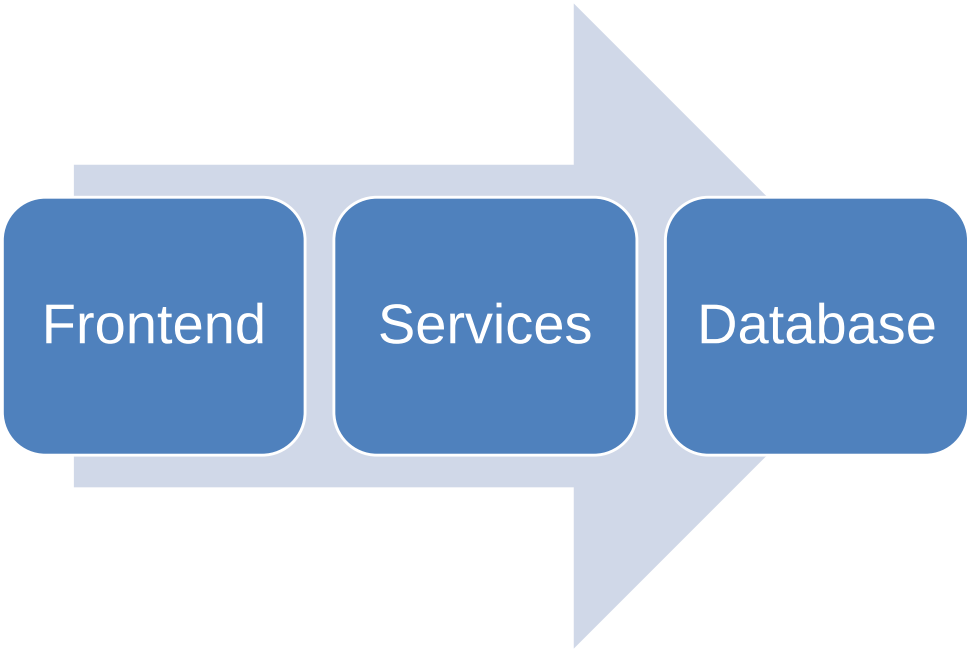

During the Build process, GeneXus creates (and by default also executes) the reorganization programs.

To export the reorg you may go to Build / Export reorganization to create a package (zip file for .NET, like the one in the example, and .jar for Java) with the files required to execute the reorg programs that have been created in the last compilation process.

The package may be sent to the production database server to decompress and execute it there, to create or reorganize the structures of the corresponding database.

Search the Wiki to find additional data on how to export reorganizations in the article entitled “Export Reorganization”
https://wiki.genexus.com/commwiki/servlet/wiki?34476

Summary




This video showed you how to deploy an app generated in Angular, in its three components: front-end, REST services and database reorganizations.

Deployment is part of the development process, particularly in a DevOps development cycle where it is possible to automate this process along with other processes that allow the app's new version to function in production in a brief period and in an agile and continuous manner.

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