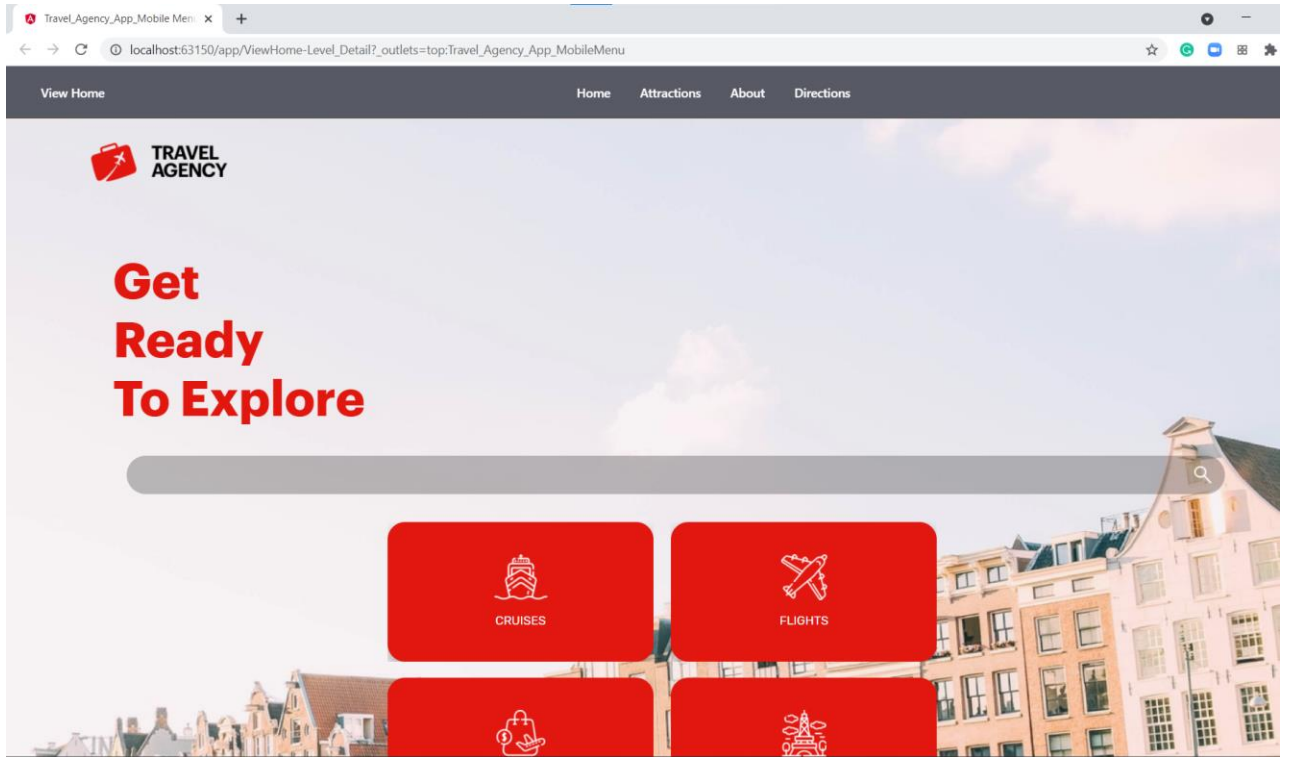


# Getting Started with Angular

*GeneXus™*

Throughout this course, we will show how we implemented an application for the customers of a Travel Agency. Also, all the examples provided will deal with its development. But first let's take a look at the application already in operation.



We open GeneXus, press F5 and see that the web browser opens with the initial screen of the application.

In the upper section, there is the agency's logo and in the gray section a bar with quick access buttons to the attractions' data. The center of the page has large buttons that give access to the different parts of the application to view information on cruises, flights, tour packages, and attractions.

Travel\_Agency\_App\_Mobile Menu x +

localhost:63150/app/ViewAttractions-Level\_Detail?\_outlets=top:Travel\_Agency\_App\_MobileMenu

View Attractions Home Attractions About Directions

The new age of  
EXPLORATION

CRUISES FLIGHTS PACKAGES **ATTRACTIONS**

POPULAR ATTRACTIONS →

**Louvre Experience**  
PARIS, FRANCE  
BOOK NOW 4 ★

**Glenfinnan Viaduct**  
UNITED KINGDOM  
BOOK NOW 4 ★

**Eiffel Tower**  
PARIS, FRANCE  
BOOK NOW 5 ★

**Glenfinnan Viaduct**  
UNITED KINGDOM  
BOOK NOW 4 ★

**Louvre Experience**  
PARIS, FRANCE  
BOOK NOW 4 ★


If we click on this last button, we access a page where we can navigate horizontally among the most popular attractions. For each attraction we see its image, location, rating and a button to schedule a visit.

Travel\_Agency\_App\_Mobile Men x +

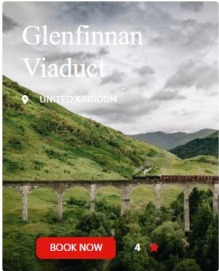
localhost:63150/app/ViewAttractions-Level\_Detail?\_outlets=top:Travel\_Agency\_App\_MobileMenu

View Attractions Home Attractions About Directions


POPULAR ATTRACTIONS →



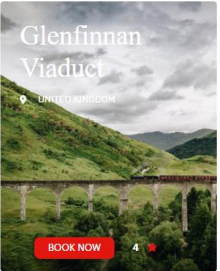
Louvre Experience  
PARIS, FRANCE  
BOOK NOW 4 ★




Glenfinnan Viaduct  
UNITED KINGDOM  
BOOK NOW 4 ★




Eiffel Tower  
PARIS, FRANCE  
BOOK NOW 5 ★




Glenfinnan Viaduct  
UNITED KINGDOM  
BOOK NOW 4 ★




Louvre Experience  
PARIS, FRANCE  
BOOK NOW 4 ★




Louvre Experience  
PARIS, FRANCE  
BOOK NOW 4 ★




Rifugio Nuvolau  
ITALY  
BOOK NOW 5 ★




London Towers  
UNITED KINGDOM  
BOOK NOW 4 ★




Cinque Terre  
ITALY  
BOOK NOW 4 ★




Sacre Coeur  
FRANCE  
BOOK NOW 5 ★



Statue of Liberty  
USA  
BOOK NOW 4 ★

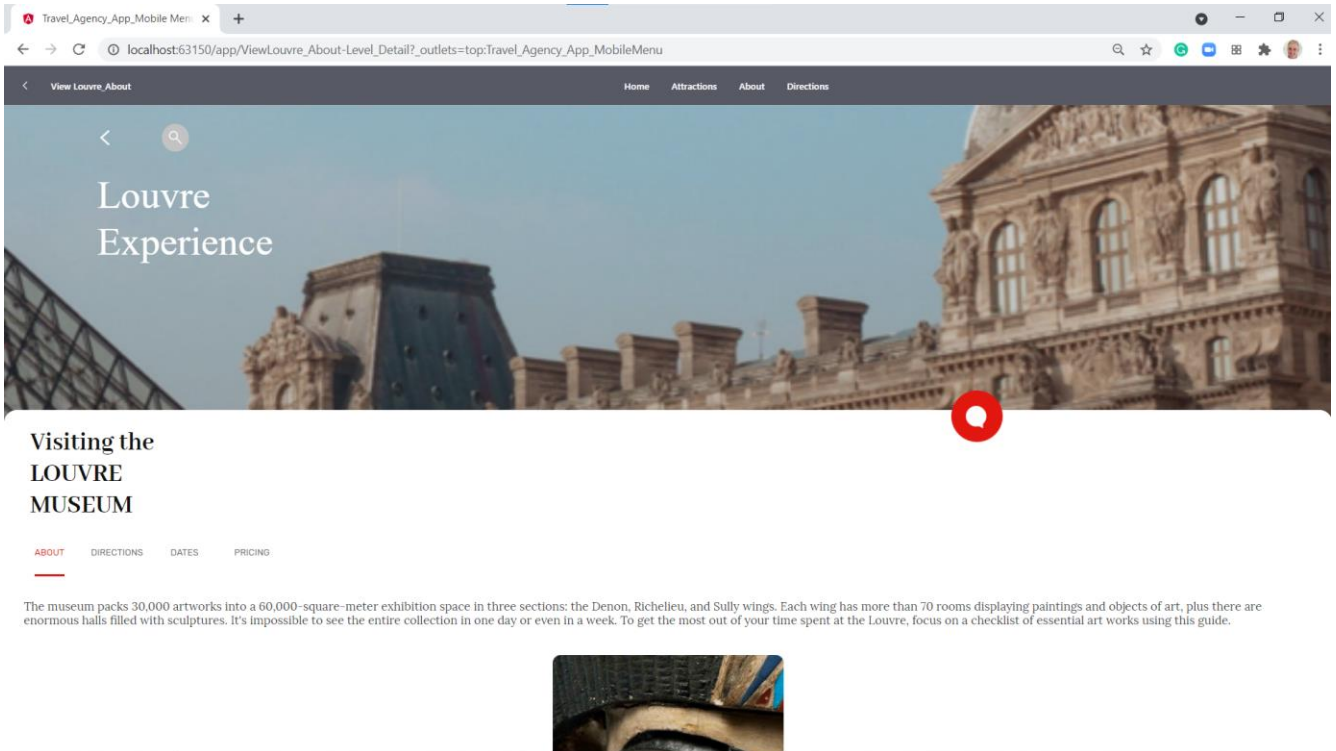


Taj Mahal  
INDIA  
BOOK NOW 4 ★



Chichen Itza  
MEXICO  
BOOK NOW

Further down in the page, we see another list with all the attractions to visit. Here we can also move horizontally, and for each attraction we see its country, rating and buttons to book a visit.



If we click on a popular attraction, such as the Louvre, a page opens with the details of the attraction containing a brief description with representative images.

There is also a submenu, which in addition to this general data (marked with About) gives access to other attraction data such as addresses, dates of visits and prices.

Travel\_Agency\_App\_Mobile Men: x +

localhost:63150/app/ViewLouvre\_Directions-Level\_Detail?\_outlets=top:Travel\_Agency\_App\_MobileMenu

View Louvre\_Directions Home Attractions About Directions

## Visiting the LOUVRE MUSEUM

ABOUT DIRECTIONS DATES PRICING

**DIRECTIONS**

**Louvre Museum**  
Rue de Rivoli,  
75001 Paris, France

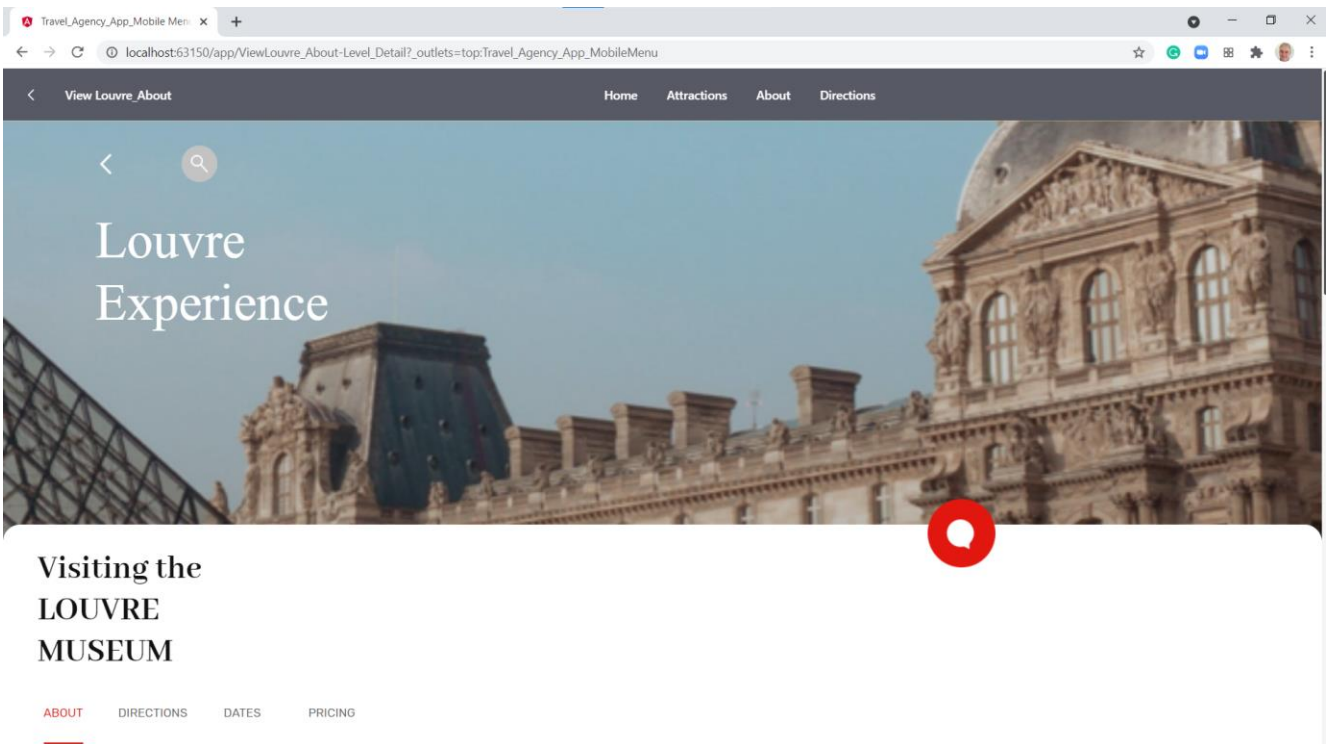
**Metro**  
Palais-Royal Musée du Louvre  
(lines 1 and 7) and Pyramides (line 14)

**Bus**  
no. 21, 24, 27, 39, 48, 68, 69, 72, 81, 95

Place du Ca  
Société des Amis du Louvre  
Musée du Louvre - Département des...  
Louvre Museum  
Landmark art museum  
Louvre - Rivoli M

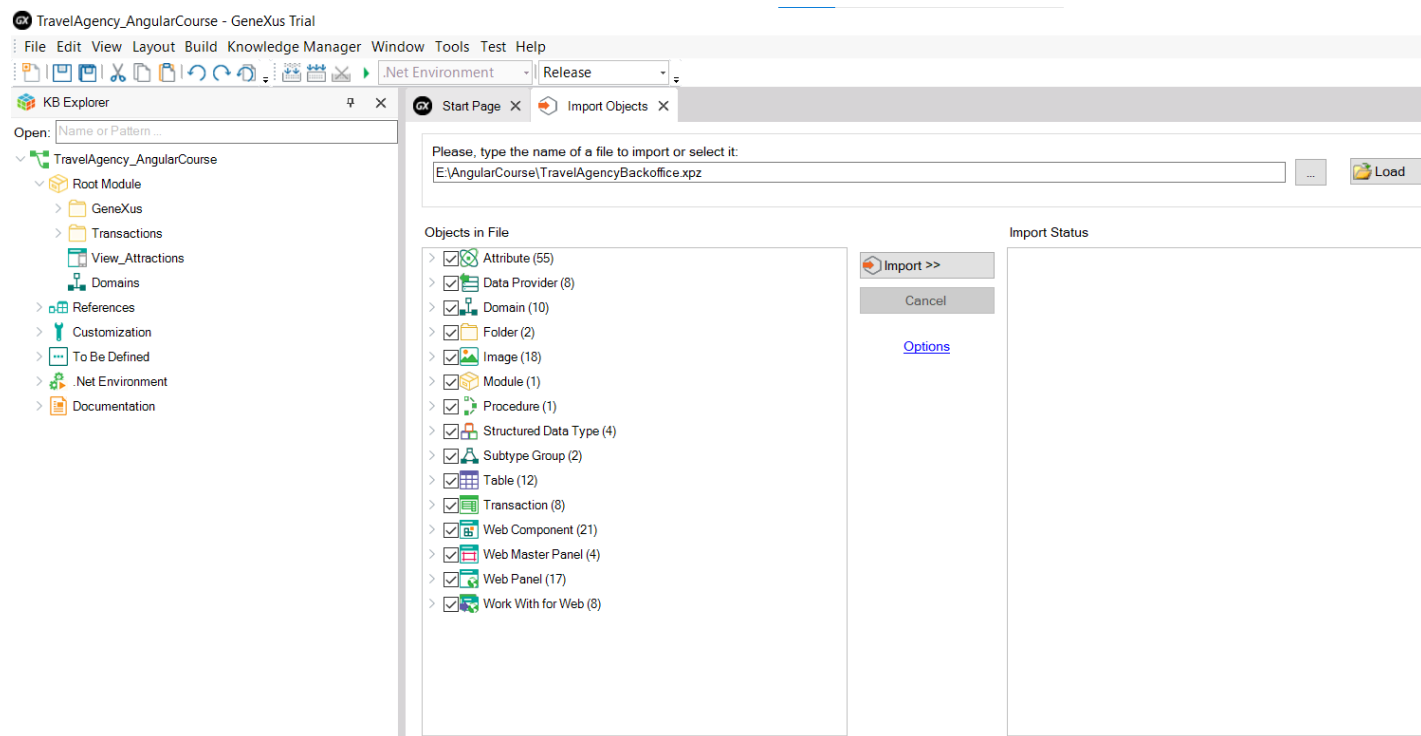
Selecting addresses shows the street where the Louvre museum is located, its ZIP code, city and country. It also shows the subway lines and stops and the bus lines to access the museum.

Further down, there is a map showing the precise location of the entrance. Dates and visiting hours, as well as prices, have not been implemented yet.



Clicking on the Attractions button on the top bar will take us back to the list of attractions. And by clicking on home, we return to the initial screen.

In the following videos, we will see how to complete these functionalities. At first, we won't pay much attention to appearance aspects such as colors, button shapes and font styles because we will focus on the development of what the application has to do and its logic; once this is done, we will see how to import into our application the appearance definitions made by a design professional who will help us achieve the desired look and feel.



Let's go back to GeneXus and start developing the application we saw. First, we create a KB called `TravelAgency_AngularCourse`.

Now we import the data model with all the transactions that we will use, to which we have already applied the `WorkWith for Web` pattern, so we will have the back office of the travel agency application available.

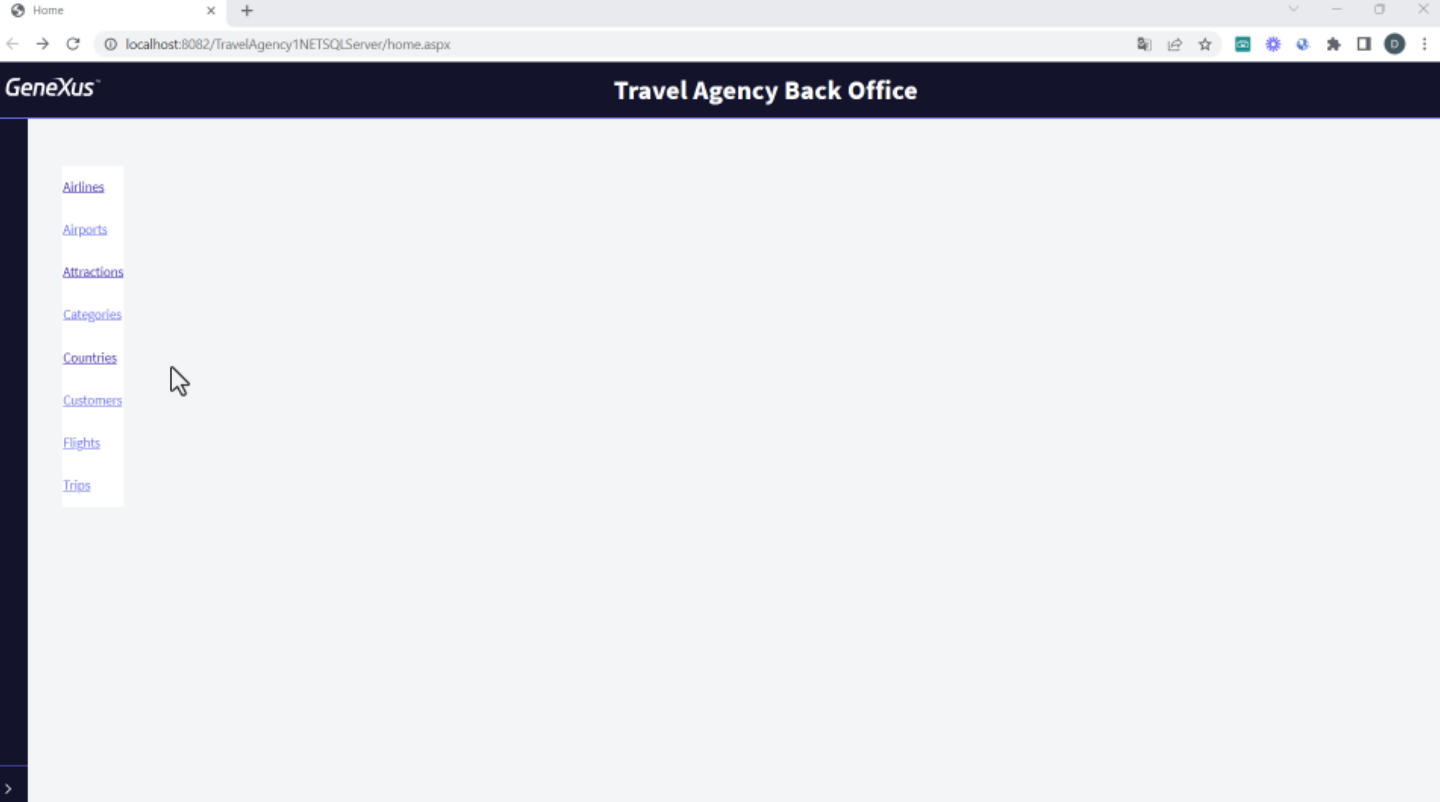
This web backoffice, previously made, was imported to be used as is, for its implementation is not part of the course's objectives. In this course we will be developing the web frontend oriented at customer-facing, in Angular.

Let's go to Knowledge Manager/Import and select the file `TravelAgencyBackoffice.xpz`

This XPZ file also contains data providers for data loading, so when we run it, the database will be created and its tables will already contain data.

First, let's select the Home web panel as the startup object and press F5 to become familiar with the travel agency entities while running the application's back office.





In this Home web panel, we see a link to each entity. If we go to the countries, we see that several countries are loaded, each one with its cities. If we go to the tourist attractions, we see which attraction belongs to a country and a city, including detailed information about the attraction.











If we go back to the Home, we can browse the other travel agency data. If you have any doubts about transaction objects, the Work With pattern or data providers, we suggest you review these concepts in the GeneXus Core course.

Attractions

localhost:8082/TravelAgency1NETSQLServer/wwattraction.aspx

INSERT

Q Name

Name	Country Name	Category Name	Photo	City Name	UPDATE	DELETE
<a href="#">Christ the Redeemer</a>	<a href="#">Brazil</a>	<a href="#">Monument</a>		Rio de Janeiro	UPDATE	DELETE
<a href="#">Cinque Terre</a>	<a href="#">Italy</a>	<a href="#">Tourist site</a>		Maranola	UPDATE	DELETE
<a href="#">Eiffel Tower</a>	<a href="#">France</a>	<a href="#">Monument</a>		Paris	UPDATE	DELETE
<a href="#">Forbidden city</a>	<a href="#">China</a>	<a href="#">Tourist site</a>		Beijing	UPDATE	DELETE
<a href="#">Glenfinnan Viaduct</a>	<a href="#">Scotland</a>	<a href="#">Tourist site</a>		Glenfinnan	UPDATE	DELETE
<a href="#">London Towers</a>	<a href="#">England</a>	<a href="#">Monument</a>		London	UPDATE	DELETE
<a href="#">Long Bridge</a>	<a href="#">United States</a>	<a href="#">Tourist site</a>		San Francisco	UPDATE	DELETE
<a href="#">Louvre</a>	<a href="#">France</a>	<a href="#">Museum</a>		Paris	UPDATE	DELETE
<a href="#">Matisse Museum</a>	<a href="#">France</a>	<a href="#">Museum</a>		Nice	UPDATE	DELETE
<a href="#">Meet the Emperor</a>	<a href="#">China</a>	<a href="#">Monument</a>		Beijing	UPDATE	DELETE

< > >>

If we go to the tourist attractions, we see which attraction belongs to a country and a city, including detailed information about the attraction.

If we go back to the Home, we can browse the other travel agency data. If you have any doubts about transaction objects, the Work With pattern or data providers, we suggest you review these concepts in the GeneXus Core course.

The screenshot displays the GeneXus IDE interface. On the left, the 'View\_Attractions' panel is shown with a grid containing five columns: 'AttractionId', 'AttractionName', 'AttractionPhoto' (with a landscape image icon), 'CityName', and 'CountryName'. The 'Properties' window on the right shows the configuration for the 'Panel: View\_Attractions', including its name, description, module, and main program settings. The 'KB Explorer' on the right shows the project structure, highlighting the '.NET Environment' and 'Front end' folders. The 'Properties' window for the 'Generator: Frontend (Front end)' is also visible, showing settings for generating the application for different platforms.

Panel: View_Attractions	
Name	View_Attractions
Description	View_Attractions
Module/Folder	Root Module
Qualified Name	View_Attractions
Object Visibility	Public
Main program	True
Master Panel	(none)
Caption	View_Attractions
Miscellaneous	
Network	
Connectivity Support	Online
Deep Link Name	
Main object properties	
Application Title	

Generator: Frontend (Front end)	
Name	Frontend
Generate Android	False
Generate Apple	False
Main Platform	Angular
Dynamic Services URL	False
Services URL	https://trialapp
SSL Pinning Pin Set	
Smart Devices Cache	On
Generate Angular	True
Angular Specific	
Setup Command	npm install
Run Command	npm start
Build Mode	Prototype
Default Platform	Best fit

Let's return to GeneXus and start implementing the first screen of the application.

The travel agency has asked us to build a panel that shows a horizontal list of the available attractions. When we select one, the details of the attraction should be displayed.

For that we create a Panel object and name it View\_Attractions.

We drag a grid to the form and select the attributes AttractionId, AttractionName, AttractionPhoto, CityName and CountryName.

Let's go to the panel properties and set it as main object.

Now in the Knowledge Base Navigator we double click on .NET environment, select Front end and go to its properties. We set Generate Angular to True, and then Generate Android and Generate Apple to False. Note that the Main Platform property is now set to Angular.

GeneXus Community Wiki

MENU - PAGE INFO - PAGE TOOLS -

Try a Search

Recents Search Wiki Home

ANGULAR APPLICATIONS DEVELOPMENT

- Getting started
  - Angular Overview
- Requirements & execution
  - Prerequisites**
    - How to execute the app
    - How to deploy the app
- Modeling
  - Roadmap
  - Troubleshooting
- Media

## < Angular Generator prerequisites for development environment

This documentation is valid for:

[GeneXus Beta](#)

For [Angular applications development](#), your development environment should include the following:

- **Node.js**

Get Node.js from [nodejs.org](#) (download the LTS version). After the installation of Node.js, to check your version run the following command in a terminal/console window:

```
node -v
```

Version has to be v12.17 or higher

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Node.js® is an open-source, cross-platform JavaScript runtime environment.

### Download for Windows (x64)

**18.17.1 LTS**  
Recommended For Most Users

**20.6.1 Current**  
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)

For information about supported releases, see the [release schedule](#).

Before generating our application in Angular, we must install some applications. We can see this if we go to the wiki, and search for: "Angular requirements".

We see that we have to install Node.js from the nodejs.org page. Here we download and install the LTS version. I've already done it on my machine.

For [Angular Applications Development](#), your development environment should include the following:

- **Node.js**

Get Node.js from [nodejs.org](https://nodejs.org) (download the LTS version). After the installation of Node.js, to check your version run the following command in a terminal/console window:

```
node -v
```

- **npm package manager**

For the environment setup, npm packages will be installed (automatically) using the [npm client](#) command line interface, so you must have an npm package manager. Node.js includes it by default. So, after installing Node.js, check that you have the npm client installed, by running the following command in a terminal/console window (npm version should be 7 or higher):

```
npm -v
```

Once installed, to check the version we open a command window and type: **node -v**

Then we have to install an npm package manager client, but since we already have Node.js installed, it is already included.

To confirm that it has been correctly installed, we write “npm -v”, and we obtain the version.

Now we have everything ready to build and run our application in Angular.

```
Administrator: Windows PowerShell
.69 kB
node_modules_genexus_web-controls-library_dist_esm_gx-map-line_entry_js.js | - | 2
.61 kB
node_modules_genexus_web-controls-library_dist_esm_gx-gauge-range_entry_js.js | - | 2
.30 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-parameter_entry_js.js | - | 1
.63 kB
node_modules_genexus_web-controls-library_dist_esm_gx-audio_entry_js.js | - | 1
.59 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-element-format_entry_js.js | - | 1
.46 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-format-style_entry_js.js | - | 1
.45 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-element_entry_js.js | - | 1
.41 kB

Build at: 2021-08-03T13:15:59.371Z - Hash: 9d0ed77ac7a19ddd9fa8 - Time: 17217ms

Warning: E:\Models\TravelAgency_AngularCourse\CSharpModel\mobile\Angular\View_Attractions\node_modules\@genexus\web-standard-functions\dist\lib-esm\types\guid.js depends on 'uuid'. CommonJS or AMD dependencies can cause optimization bailout
s.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies














** Angular Live Development Server is listening on localhost:54960, open your browser on http://localhost:54960/ **

√ Compiled successfully.
- Generating browser application bundles...
```

We go back to GeneXus, open the View\_Attractions panel, right-click and select Run.

A command window opens and several lines written by the Angular generator are executed.

When finished, the web browser opens automatically and we see the list of tourist attractions.

View_Attractions					
1	Eiffel Tower		France	Paris	
2	Glenfinnan Viaduct		Scotland	Glenfinnan	
3	Meet the Emperor		China	Beijing	
4	Christ the Redemmer		Brazil	Rio de Janeiro	
5	Rifugio Nuvolau		Italy	Belluno	
6	London Towers		England	London	
7	Louvre		France	Paris	
8	Forbidden city		China	Beijing	
9	Cinque Terre		Italy	Maranola	
10	Smithsonian Institute		United States	Washington	
11	Matisse Museum		France	Nice	
12	Long Bridge		United States	San Francisco	
13	The Great Wall		China	Beijing	

We already have our first Angular application running.

Looking at the URL we see that it was run on localhost, a colon and a dynamically assigned port. This corresponds to the web server instantiated to run the Angular application. The application executed is View\_Attractions-Level\_Detail, which corresponds to the attractions grid included in the panel.

**Data Provider View\_Attractions\_Level\_Detail\_Grid1 Navigation Report**

**Name:** View\_Attractions\_Level\_Detail\_Grid1  
**Description:** View\_Attractions\_Level\_Detail\_Grid1  
**Status:** No Generation Required  
**Output Devices:** None

**Environment:** C# Default (C#)  
**Spec. Version:** 17\_0\_4-151606  
**Form Class:** HTML  
**Program Name:** View\_Attractions\_Level\_Detail\_Grid1  
**Parameters:** in: &start, in: &count, in: &gxid, out: View\_Attractions\_Level\_Detail\_Grid1Sdt

**LEVELS**

**For Each Attraction (Line: 2)**

**Order:** [AttractionId](#)  
**Index:** IATTRACTION  
**Navigation filters:** Start from: FirstRecord  
 Loop while: NotEndOfTable  
**Join location:** Server  
**Optimizations:** Server Paging

[Attraction \( AttractionId \)](#) INTO [AttractionPhoto.Uri](#)  
[AttractionId](#) [AttractionName](#) [AttractionPhoto](#) [CountryId](#) [CityId](#)  
[Country \( CountryId \)](#) INTO [CountryName](#)  
[CountryCity \( CountryId, CityId \)](#) INTO [CityName](#)

If we go back to GeneXus and look at the navigation list, we see that the View\_Attractions panel contains a node called Level\_Detail\_Grid1 and has the symbol of a data provider.

By clicking on it, we see it is the navigation list of the data provider View\_Attractions\_Level\_Detail\_Grid1, which is responsible for accessing the database and retrieving the information of the tourist attractions.

Where it says Levels, we find a For Each Attraction because as our grid had attributes, GeneXus found a base table and created an implicit For Each to access the data in the database.

The Attraction table was effectively navigated to retrieve the attraction data, and then the Country table was accessed to retrieve the country name, and the CountryCity table to get the city name.

Later we will see what GeneXus takes into account to determine the base table of the grid and how the information is loaded through the data provider whose navigation list we are looking at.



Disco1T (E) > Models > TravelAgency\_AngularCourse > CSharpModel > mobile > Angular > View\_Attractions > src >

Name	Date modified	Type	Size
app	2/8/2021 14:26	File folder	
assets	2/8/2021 14:26	File folder	
environments	2/8/2021 14:26	File folder	
images	2/8/2021 14:26	File folder	
sass	2/8/2021 14:26	File folder	
translations	2/8/2021 14:26	File folder	
index.html	2/8/2021 14:26	File	
main.ts	2/8/2021 14:26	TypeScript Source File	
manifest.webmanifest	2/8/2021 14:26	File	
polyfills.ts	16/6/2021 14:31	TypeScript Source File	
test.ts	16/6/2021 14:31	TypeScript Source File	
tsconfig.app.json	16/6/2021 14:31	File	
tsconfig.spec.json	16/6/2021 14:31	File	

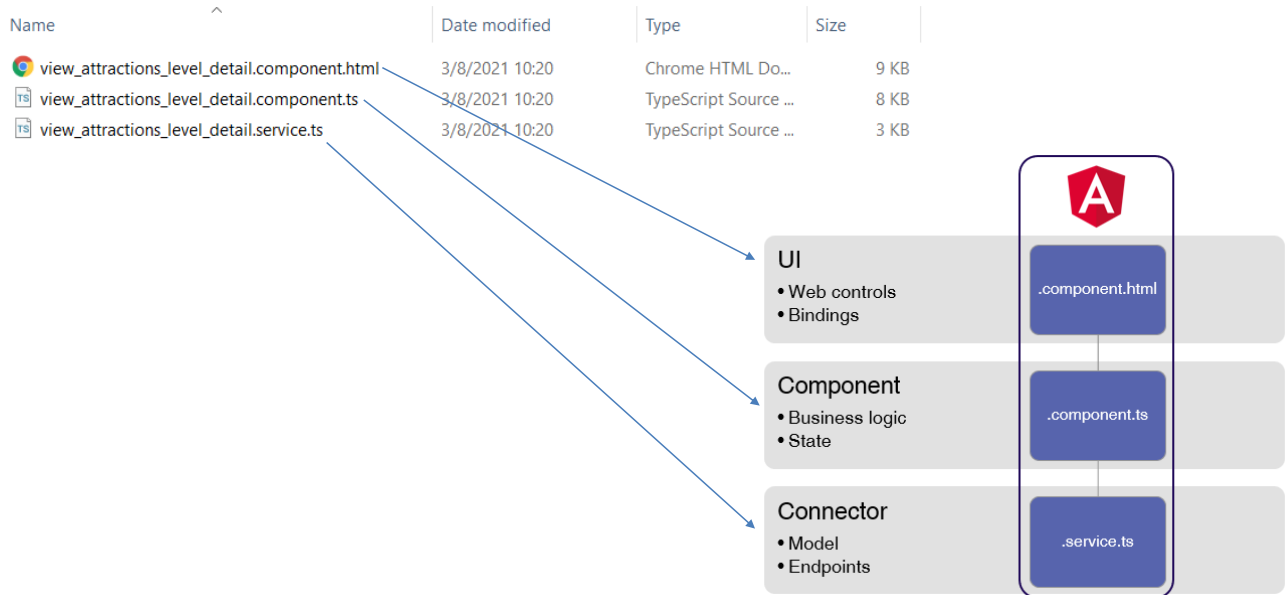
Name	Date modified	Type	Size
gx	2/8/2021 14:26	File folder	
View_Attractions	2/8/2021 14:26	File folder	
app.component.html	2/8/2021 14:26	Chrome HTML Document	4 KB
app.component.ts	2/8/2021 14:26	TypeScript Source File	12 KB
app.module.ts	2/8/2021 14:26	TypeScript Source File	3 KB
app.routing.ts	2/8/2021 14:26	TypeScript Source File	1 KB
app.settings.ts	2/8/2021 14:26	TypeScript Source File	3 KB
app-home.component.ts	2/8/2021 14:26	TypeScript Source File	1 KB
common.module.ts	16/6/2021 14:31	TypeScript Source File	4 KB
main.module.ts	2/8/2021 14:26	TypeScript Source File	1 KB
shared.module.ts	2/8/2021 14:26	TypeScript Source File	1 KB
shared-routing.module.ts	2/8/2021 14:26	TypeScript Source File	1 KB
styles.css	16/6/2021 14:31	Cascading Style Sheet	0 KB

Now let's analyze what GeneXus generates when the Angular generator is executed.

If we go to Tools/Explore Target Environment and then go to mobile/Angular, we find a View\_Attractions folder. This is because it is the main object that we executed. If we now select src/app, we see the parts that make up the application.

By clicking on View\_Attractions, we see the parts that include the Angular component, which are 3 files.














Disco1T (E:) &gt; Models &gt; TravelAgency\_AngularCourse &gt; CSharpModel &gt; mobile &gt; Angular &gt; View\_Attractions &gt; src &gt; app &gt; View\_Attractions



The file ending in **component.html** contains the definition of the application's user interface and is an Angular template that represents the panel layout. Also included here are elements that Angular interprets to associate on-screen controls with data in the database.

Then we have the file ending in **component.ts**, which is a TypeScript file where the logic of the panel we implemented is defined. This is where the events and the status of the screen controls are defined.

Lastly, we have the file ending in **service.ts** which is responsible for interacting with the web server services. Here is where communications with REST services take place, data is obtained from the database and the data structures to be used by the panel are generated.

View_Attractions			
1	Eiffel Tower		France Paris
2	Glenfinnan Viaduct		Scotland Glenfinnan
3	Meet the Emperor		China Beijing
4	Christ the Redemmer		Brazil Rio de Janeiro
5	Rifugio Nuvolau		Italy Belluno
6	London Towers		England London
7	Louvre		France Paris
8	Forbidden city		China Beijing
9	Cinque Terre		Italy Maranola
10	Smithsonian Institute		United States Washington
11	Matisse Museum		France Nice
12	Long Bridge		United States San Francisco
13	The Great Wall		China Beijing



If we go back to the attractions screen, we see a grid showing the attraction data in a top down list, which is the default mode.

Next, we will see how to change this display so as to page to the right, and other screen controls that will allow us to build the application with a more attractive user interface.

# GeneXus™

[training.genexus.com](http://training.genexus.com)

[wiki.genexus.com](http://wiki.genexus.com)

[training.genexus.com/certifications](http://training.genexus.com/certifications)