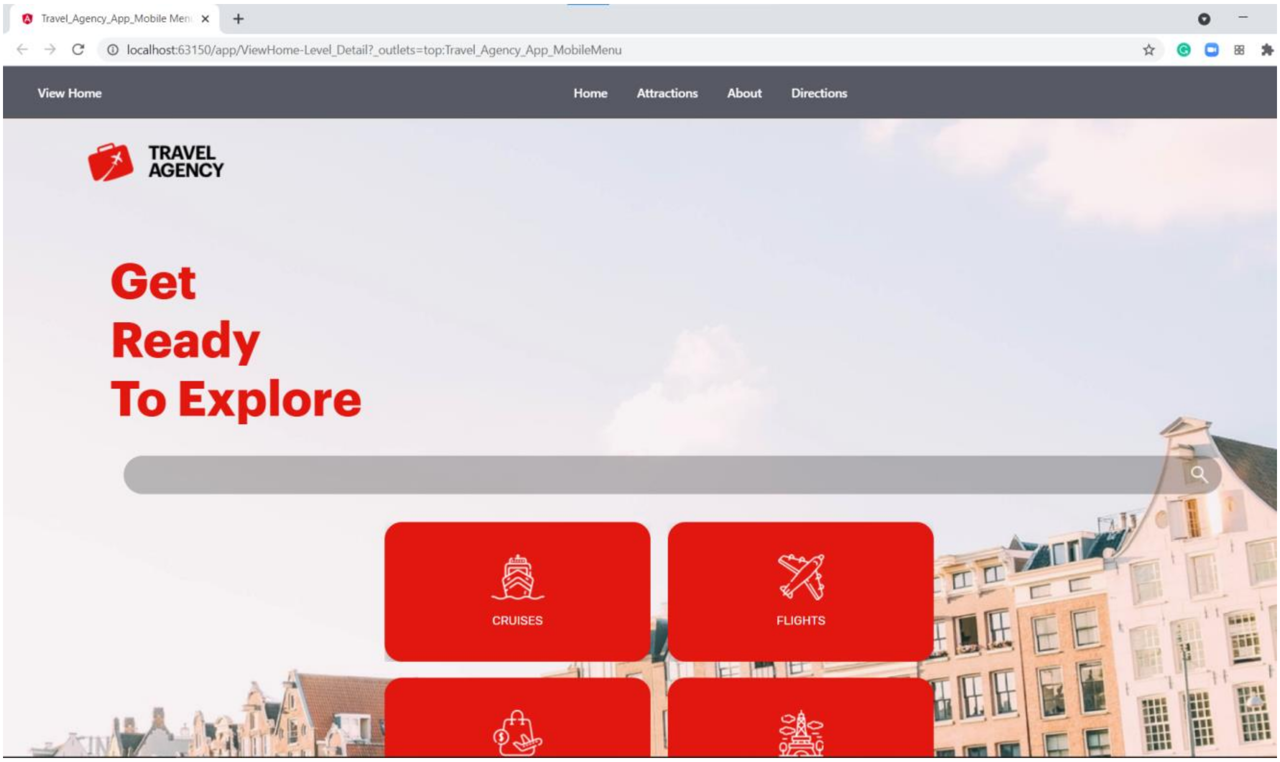
Getting Started with Angular

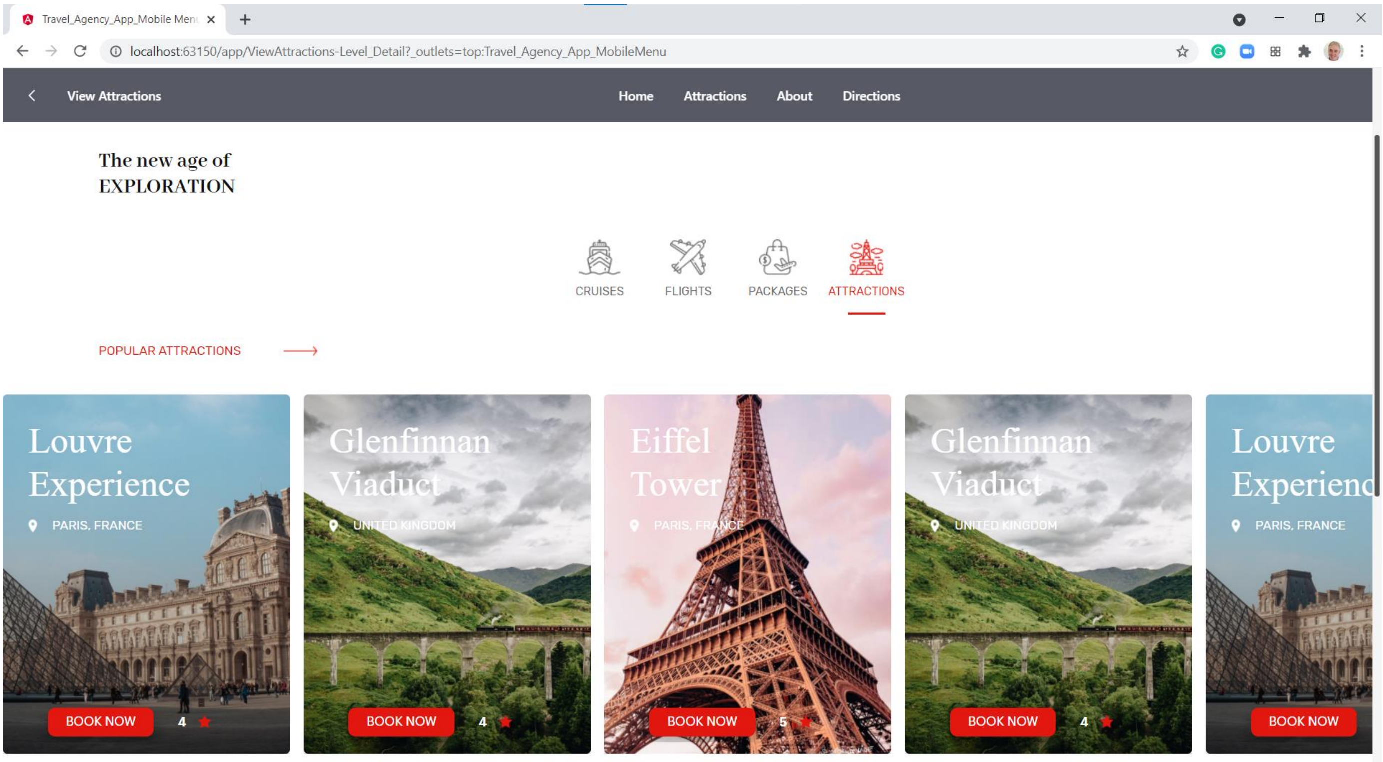
GeneXus™

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.

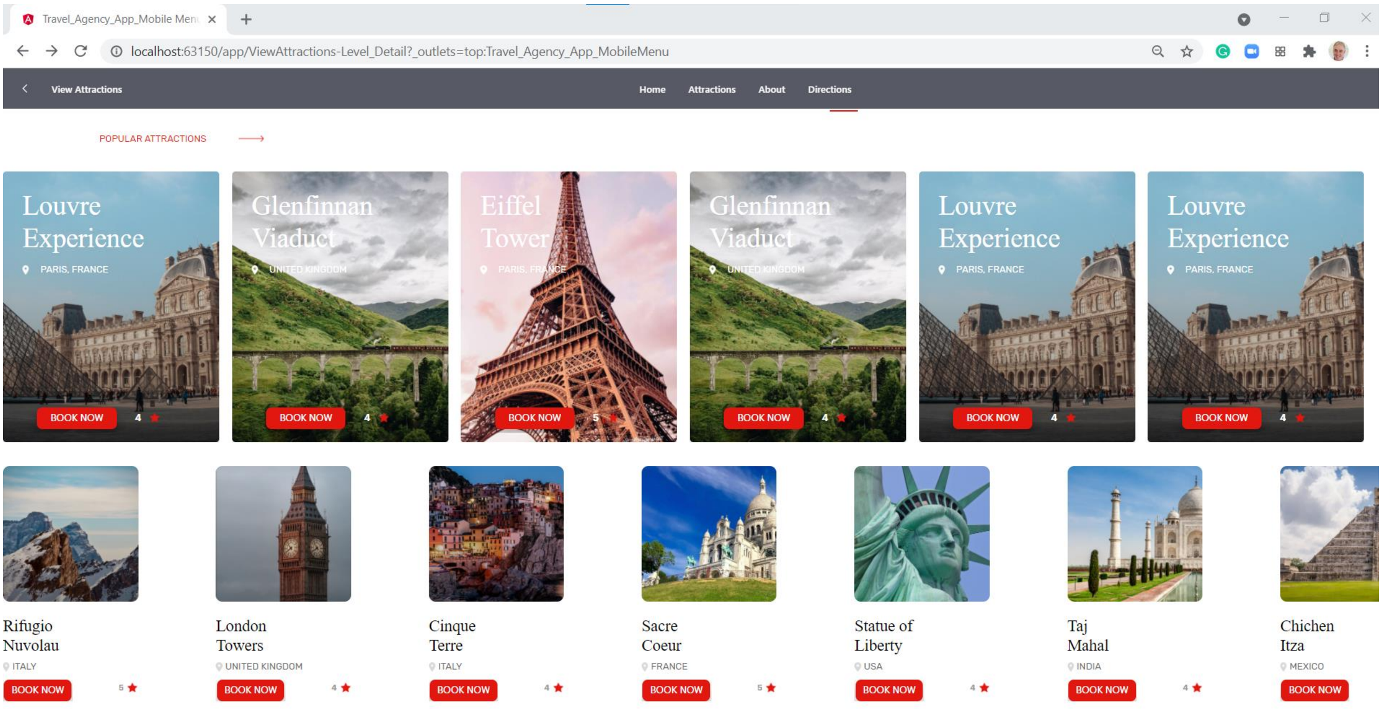

Travel_Agency_App_Mobile Menu
localhost:63150/app/ViewHome-Level_Detail?_outlets=top:Travel_Agency_App_MobileMenu
View Home
Home
Attractions
About
Directions
TRAVEL AGENCY
Get Ready To Explore
CRUISES
FLIGHTS

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

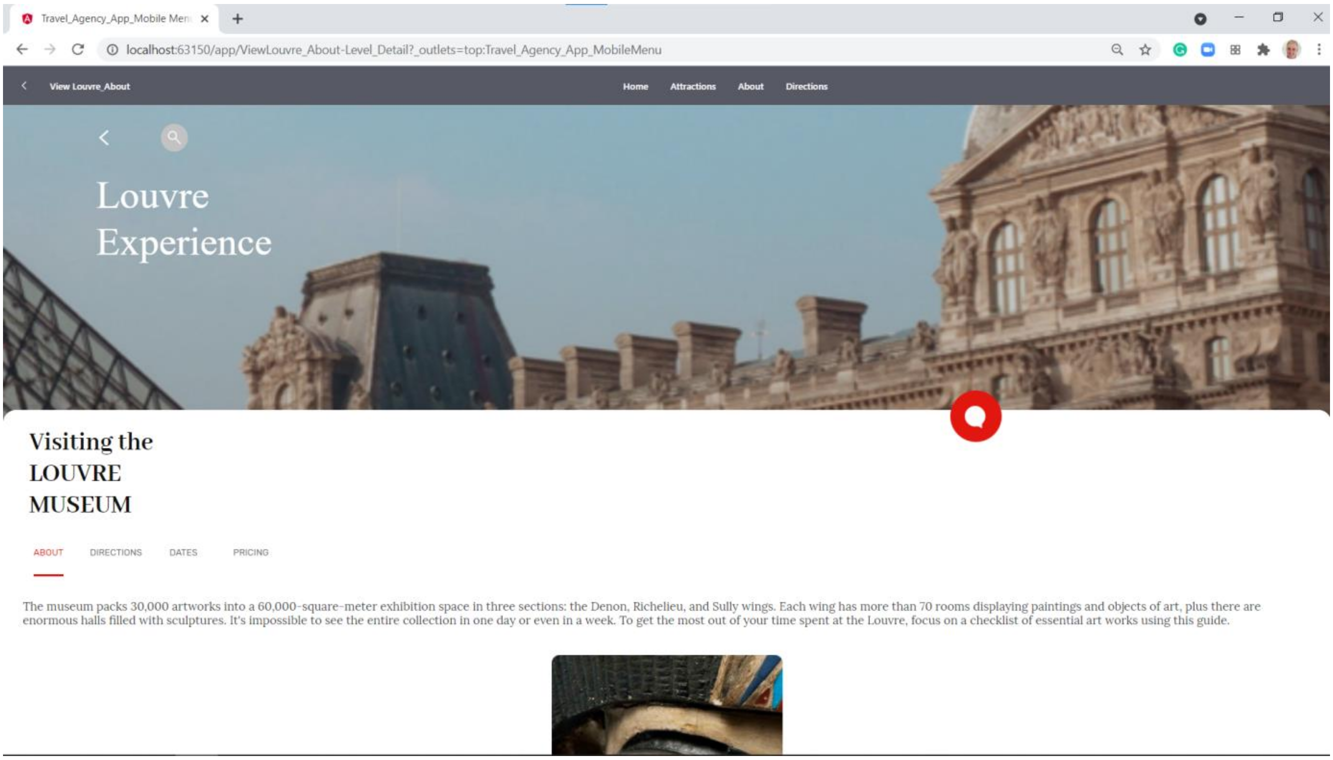
In the upper section, there is the agency's logo and in the gray section a bar with quick access buttons to the attractions' data. The center of the page has large buttons that give access to the different parts of the application to view information on cruises, flights, tour packages, and attractions.



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.



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.



Travel_Agency_App_Mobile Menu
localhost:63150/app/ViewLouvre_About-Level_Detail?_outlets=top:Travel_Agency_App_MobileMenu
View Louvre_About
Home
Attractions
About
Directions
Louvre Experience

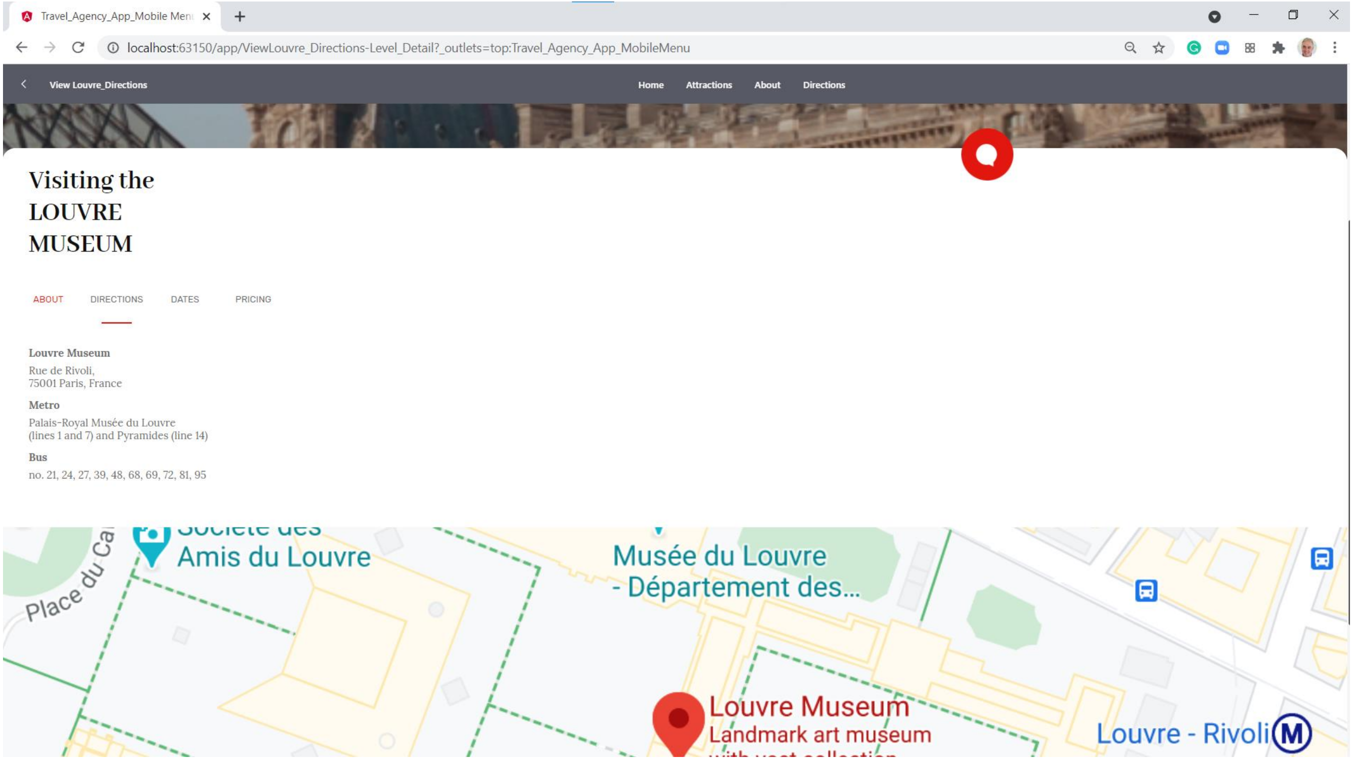
Visiting the LOUVRE MUSEUM

ABOUT
DIRECTIONS
DATES
PRICING

The museum packs 30,000 artworks into a 60,000-square-meter exhibition space in three sections: the Denon, Richelieu, and Sully wings. Each wing has more than 70 rooms displaying paintings and objects of art, plus there are enormous halls filled with sculptures. It's impossible to see the entire collection in one day or even in a week. To get the most out of your time spent at the Louvre, focus on a checklist of essential art works using this guide.

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.



Visiting the
LOUVRE
MUSEUM

ABOUT DIRECTIONS DATES PRICING

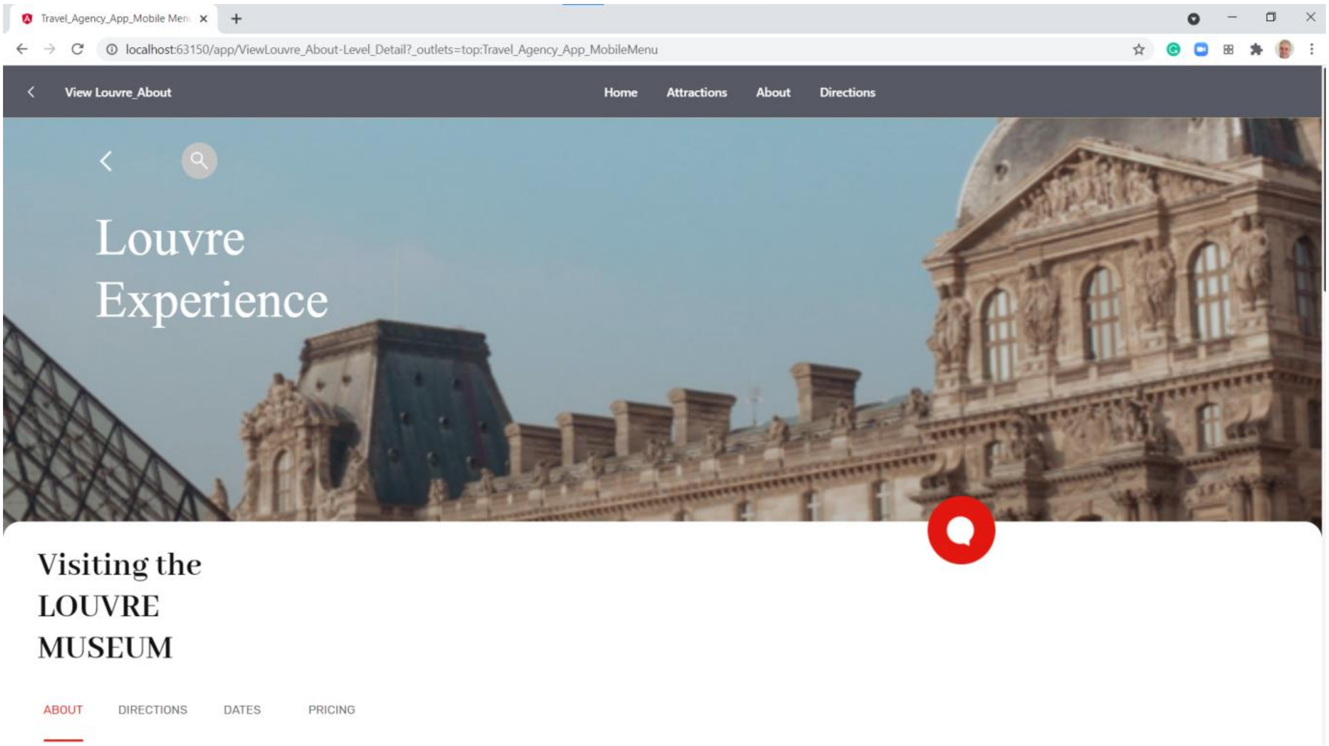
Louvre Museum
Rue de Rivoli,
75001 Paris, France

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

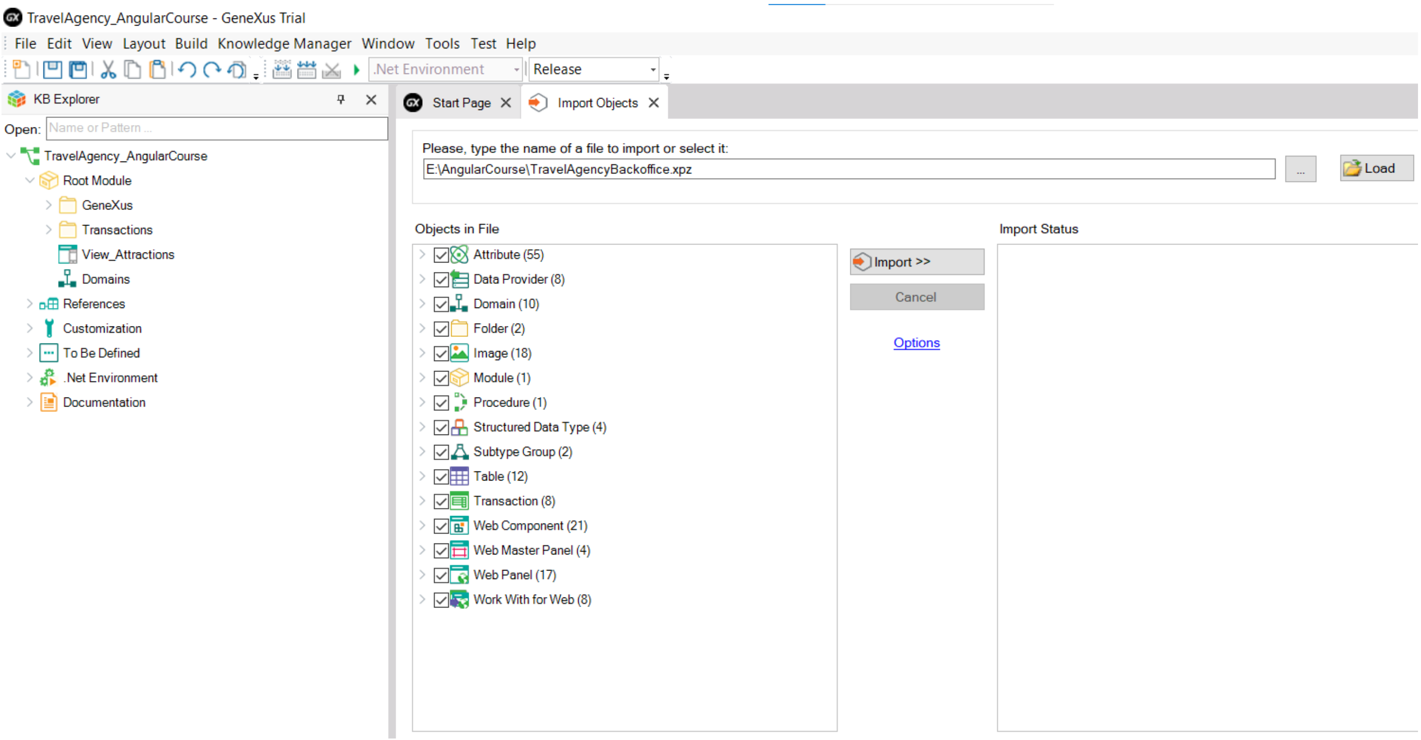
Selecting addresses shows the street where the Louvre museum is located, its ZIP code, city and country. It also shows the subway lines and stops and the bus lines to access the museum.

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.


Travel_Agency_App_Mobile Menu
localhost:63150/app/ViewLouvre_About-Level_Detail?_outlets=top:Travel_Agency_App_MobileMenu
View Louvre_About
Home
Attractions
About
Directions
Louvre
Experience
Visiting the
LOUVRE
MUSEUM
ABOUT
DIRECTIONS
DATES
PRICING

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.



TravelAgency_AngularCourse - GeneXus Trial

Please, type the name of a file to import or select it:
E:\AngularCourse\TravelAgencyBackoffice.xpz

Objects in File

- ☑ Attribute (55)
- ☑ Data Provider (8)
- ☑ Domain (10)
- ☑ Folder (2)
- ☑ Image (18)
- ☑ Module (1)
- ☑ Procedure (1)
- ☑ Structured Data Type (4)
- ☑ Subtype Group (2)
- ☑ Table (12)
- ☑ Transaction (8)
- ☑ Web Component (21)
- ☑ Web Master Panel (4)
- ☑ Web Panel (17)
- ☑ Work With for Web (8)

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

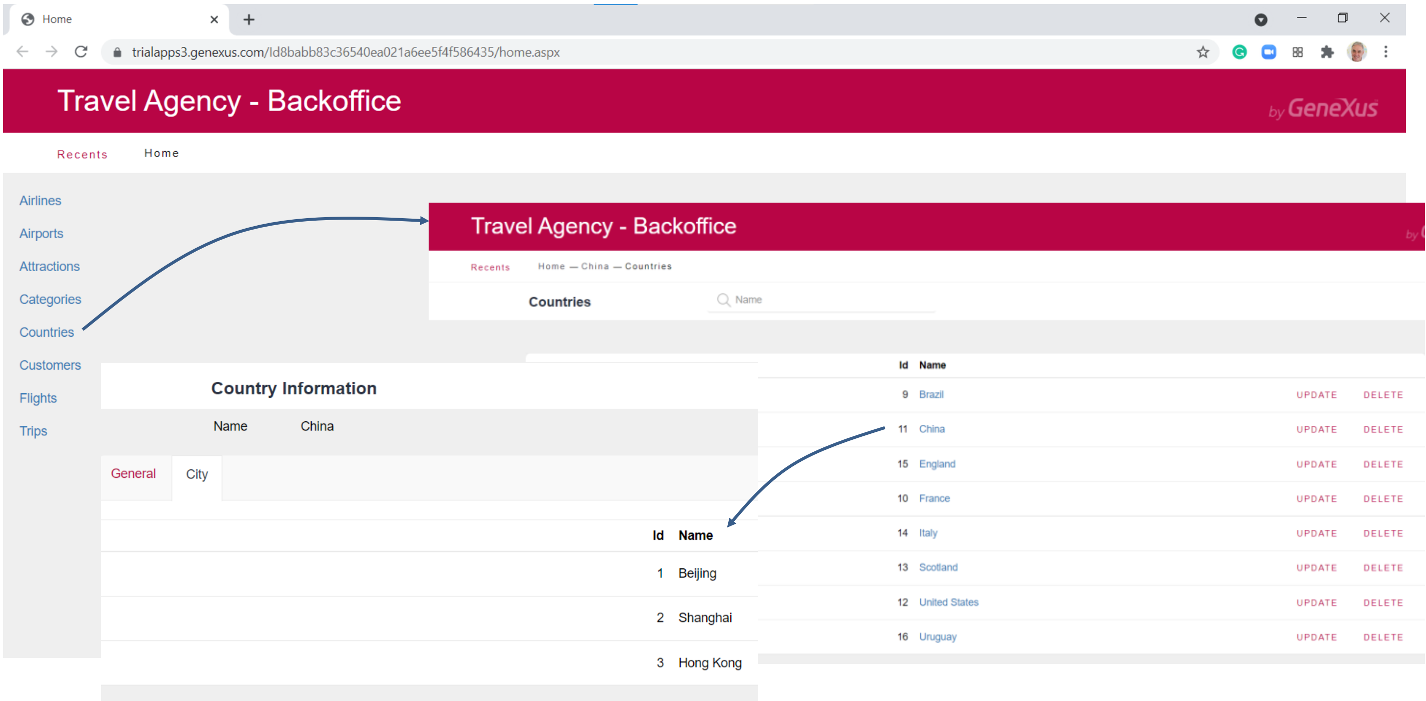
Now we import the data model with all the transactions that we will use, to which we have already applied the WorkWith for Web pattern, so we will have the back office of the travel agency application available.

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.



Home
trialapps3.genexus.com/Id8babb83c36540ea021a6ee5f4f586435/home.aspx

Travel Agency - Backoffice

Recents Home

- Airlines
- Airports
- Attractions
- Categories
- Countries
- Customers
- Flights
- Trips

Travel Agency - Backoffice

Recents Home — China — Countries

Countries

Id	Name		
9	Brazil	UPDATE	DELETE
11	China	UPDATE	DELETE
15	England	UPDATE	DELETE
10	France	UPDATE	DELETE
14	Italy	UPDATE	DELETE
13	Scotland	UPDATE	DELETE
12	United States	UPDATE	DELETE
16	Uruguay	UPDATE	DELETE

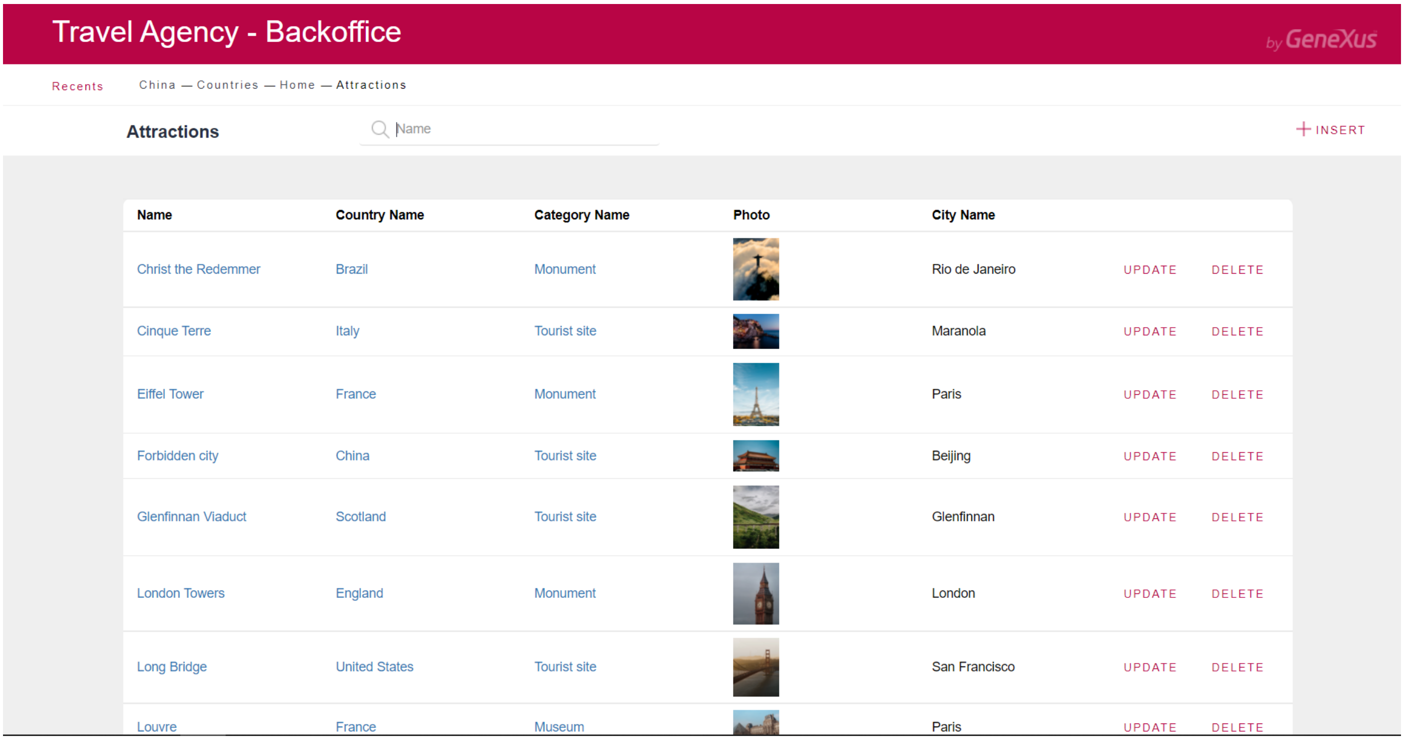
Country Information

Name China

Id	Name
1	Beijing
2	Shanghai
3	Hong Kong

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.



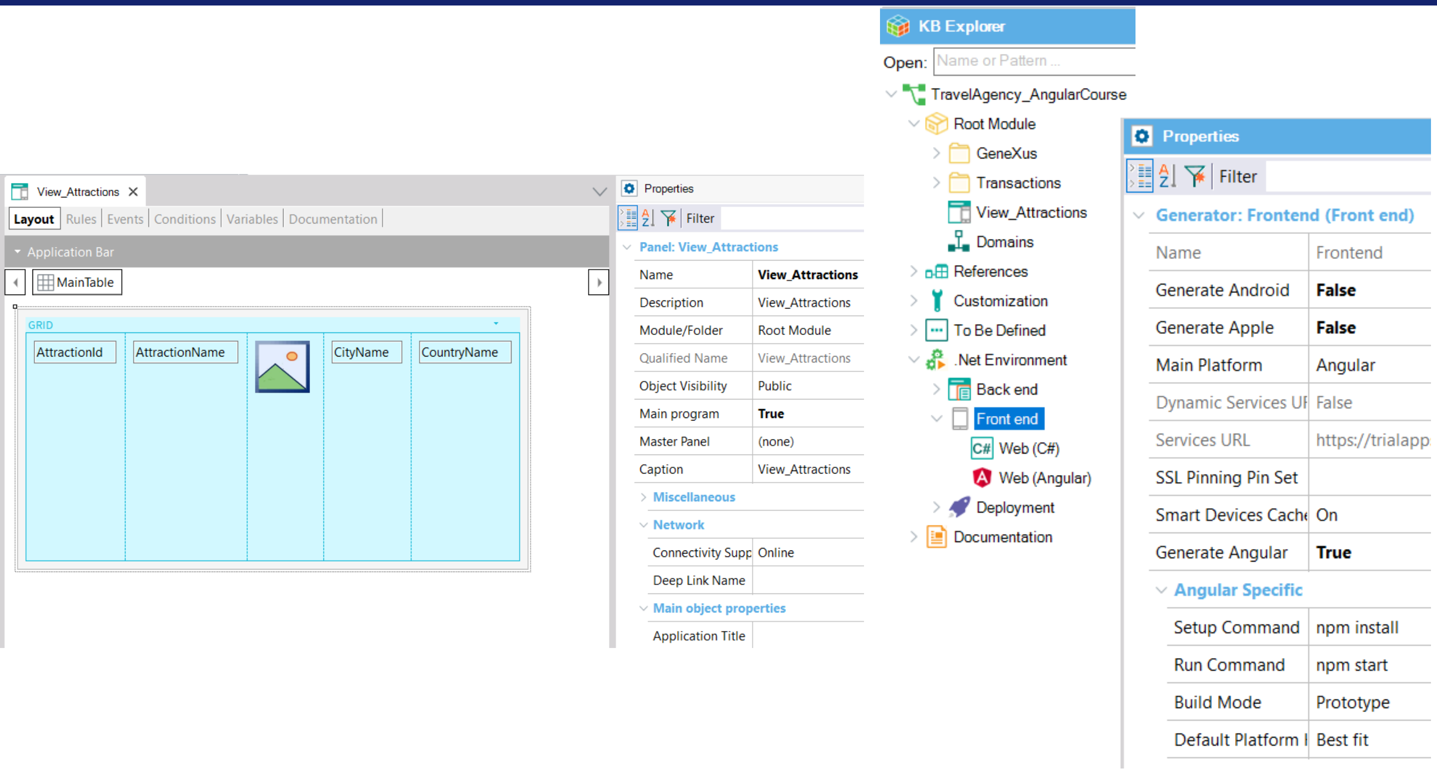
Travel Agency - Backoffice

Attractions

Name	Country Name	Category Name	Photo	City Name		
Christ the Redemmer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE
Cinque Terre	Italy	Tourist site		Maranola	UPDATE	DELETE
Eiffel Tower	France	Monument		Paris	UPDATE	DELETE
Forbidden city	China	Tourist site		Beijing	UPDATE	DELETE
Glenfinnan Viaduct	Scotland	Tourist site		Glenfinnan	UPDATE	DELETE
London Towers	England	Monument		London	UPDATE	DELETE
Long Bridge	United States	Tourist site		San Francisco	UPDATE	DELETE
Louvre	France	Museum		Paris	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.



View_Attractions

Layout | Rules | Events | Conditions | Variables | Documentation

Application Bar

MainTable

GRID

AttractionId	AttractionName		CityName	CountryName

Properties

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 Supp	Online
Deep Link Name	
Main object properties	
Application Title	

KB Explorer

Open: Name or Pattern ...

- TravelAgency_AngularCourse
  - Root Module
    - GeneXus
    - Transactions
    - View_Attractions
    - Domains
  - References
  - Customization
  - To Be Defined
  - .Net Environment
    - Back end
    - Front end
      - Web (C#)
      - Web (Angular)
    - Deployment
  - Documentation

Properties

Generator: Frontend (Front end)

Name	Frontend
Generate Android	**False**
Generate Apple	**False**
Main Platform	Angular
Dynamic Services UF	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 I	Best fit

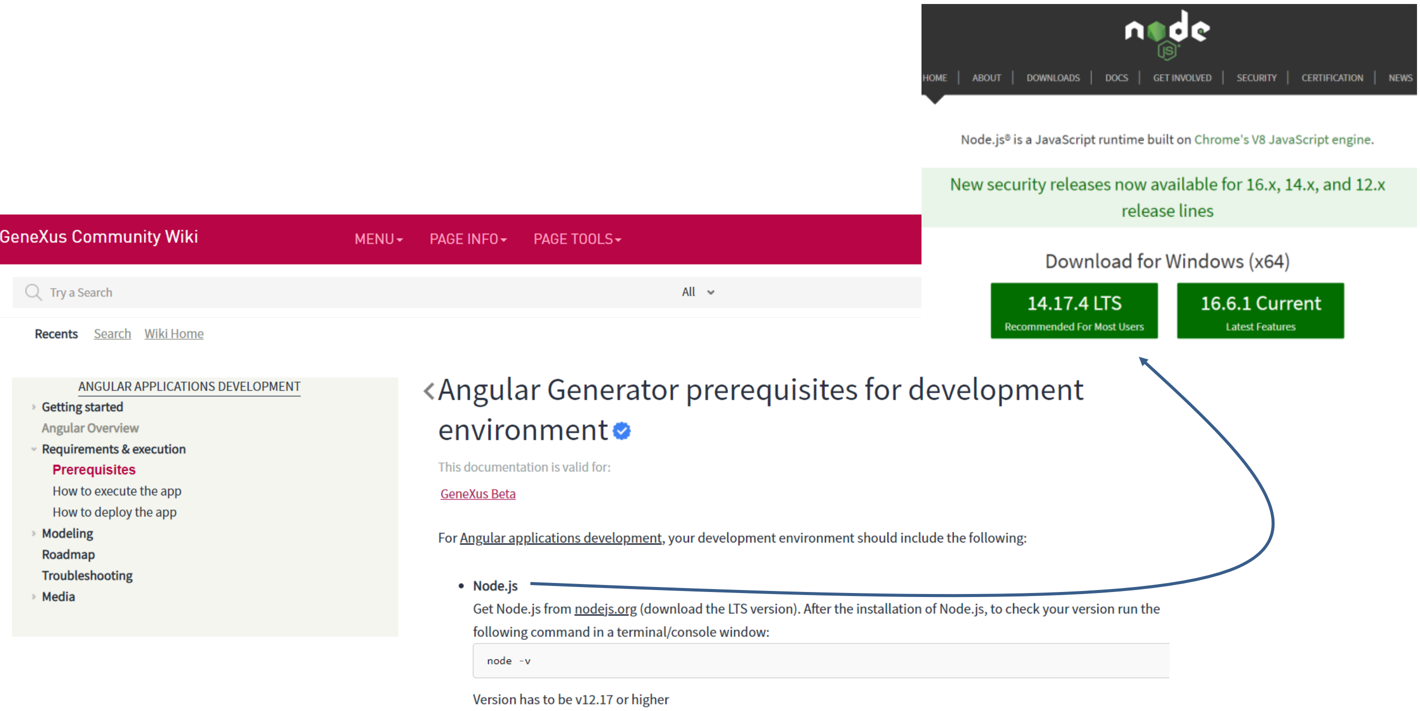
Let’s return to GeneXus and start implementing the first screen of the application.

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.



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

New security releases now available for 16.x, 14.x, and 12.x release lines

Download for Windows (x64)

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**

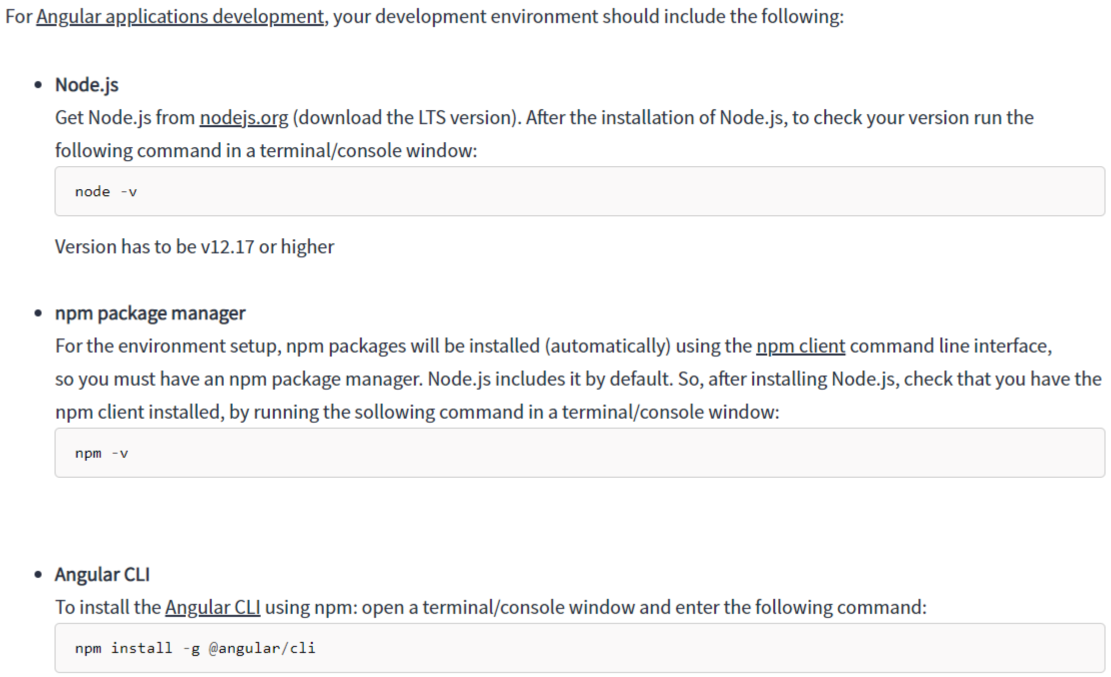
  Get Node.js from nodejs.org (download the LTS version). After the installation of Node.js, to check your version run the following command in a terminal/console window:

  ```
  node -v
  ```

  Version has to be v12.17 or higher

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 prerequisites.

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 (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

- **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 sollowing command in a terminal/console window:

  ```
  npm -v
  ```

- **Angular CLI**
  To install the Angular CLI using npm: open a terminal/console window and enter the following command:

  ```
  npm install -g @angular/cli
  ```

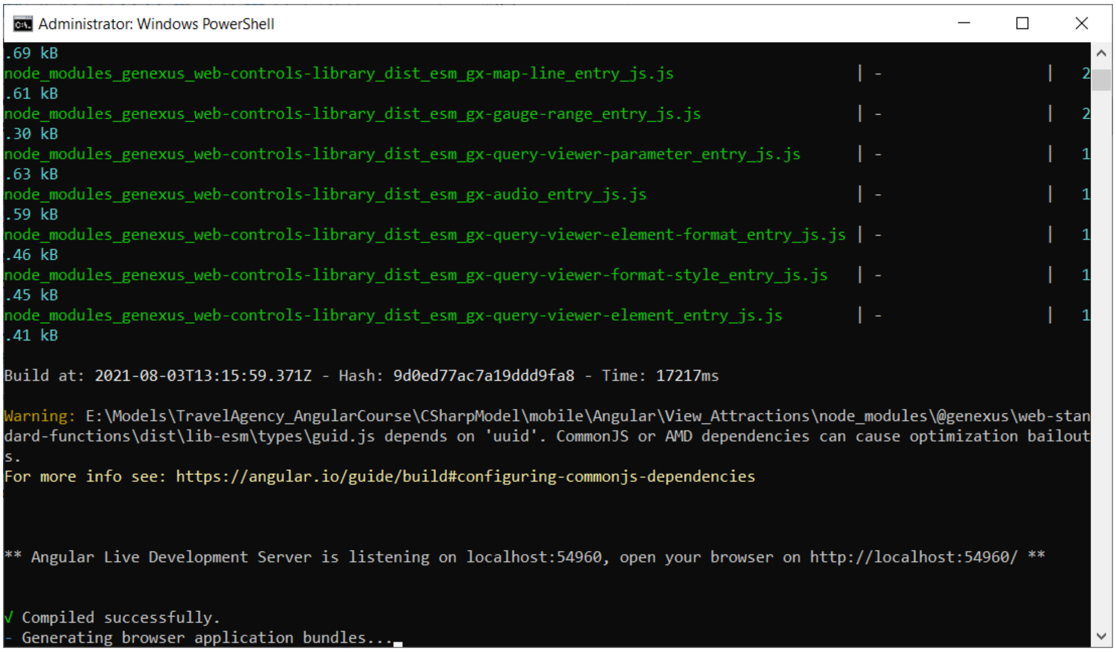
Once installed, to check the version we open a command window and type: node –v

We see that the version I have is 14.16.1. To generate Angular with GeneXus, the version must be 12.17 or higher.

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, type: npm -v and we obtain the version, in my case 6.14.12.

Lastly, we have to install the Angular command-line interface tool (Angular CLI) and for this we use npm, by typing…

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



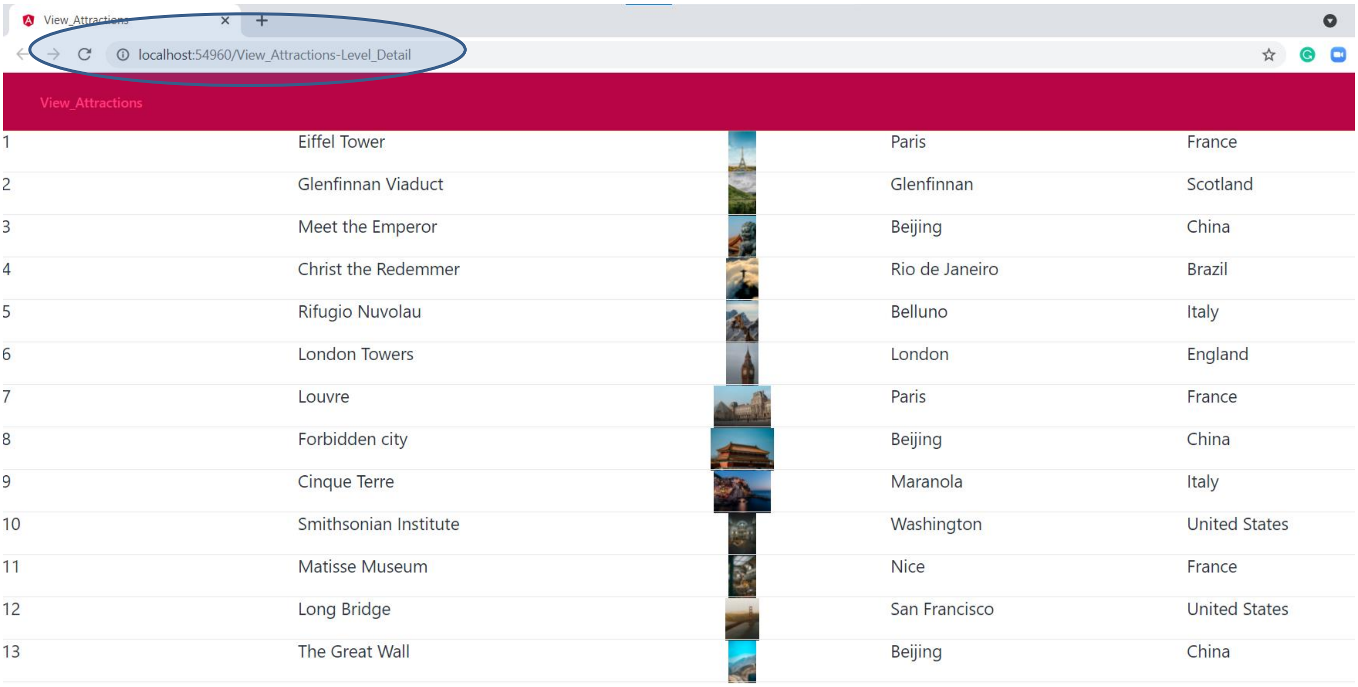
```
.69 kB
node_modules_genexus_web-controls-library_dist_esm_gx-map-line_entry_js.js                      | -                    |   2
.61 kB
node_modules_genexus_web-controls-library_dist_esm_gx-gauge-range_entry_js.js                   | -                    |   2
.30 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-parameter_entry_js.js        | -                    |   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-stan
dard-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.

After a few minutes, that window closes and another command window opens showing the result of the compilation.

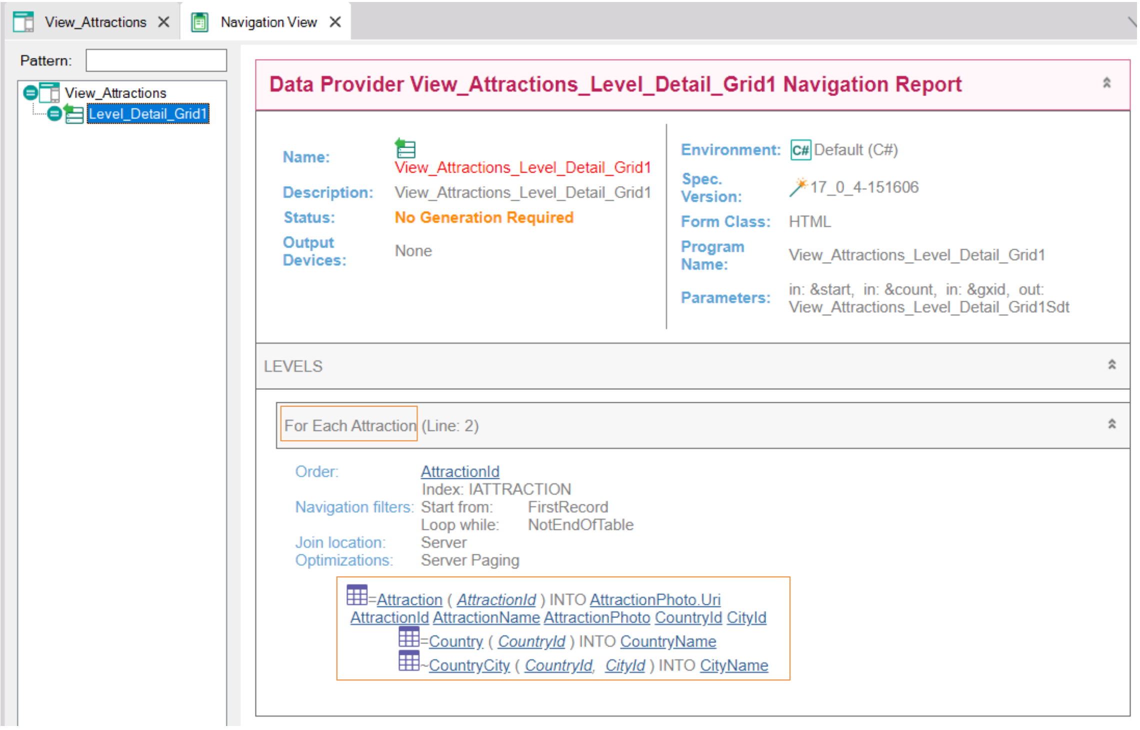
When finished, the web browser opens automatically and we see the list of tourist attractions.



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

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	Environment:	C# Default (C#)
Description:	View_Attractions_Level_Detail_Grid1	Spec. Version:	17_0_4-151606
Status:	No Generation Required	Form Class:	HTML
Output Devices:	None	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
```

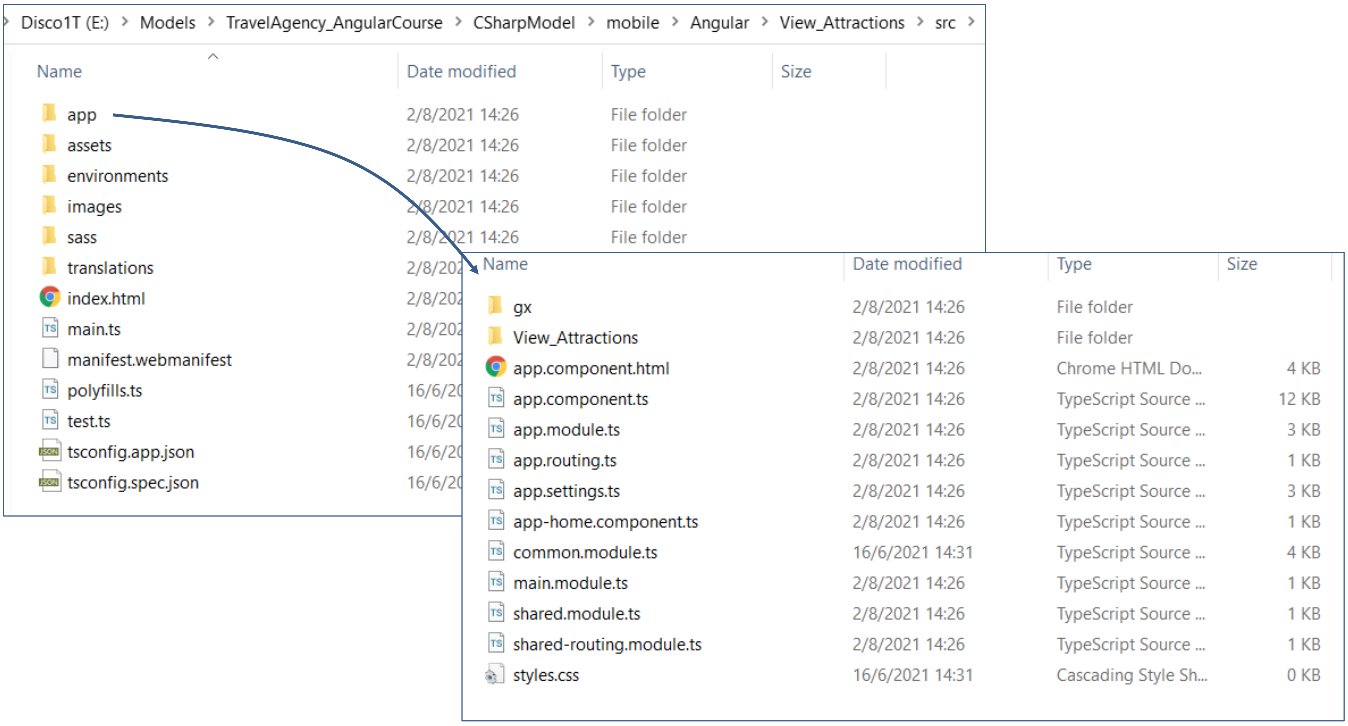
If we go back to GeneXus and look at the navigation list, we see that the View_Attractions panel contains a node called Level_Detail_Grid1 and has the symbol of a data provider.

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 ›

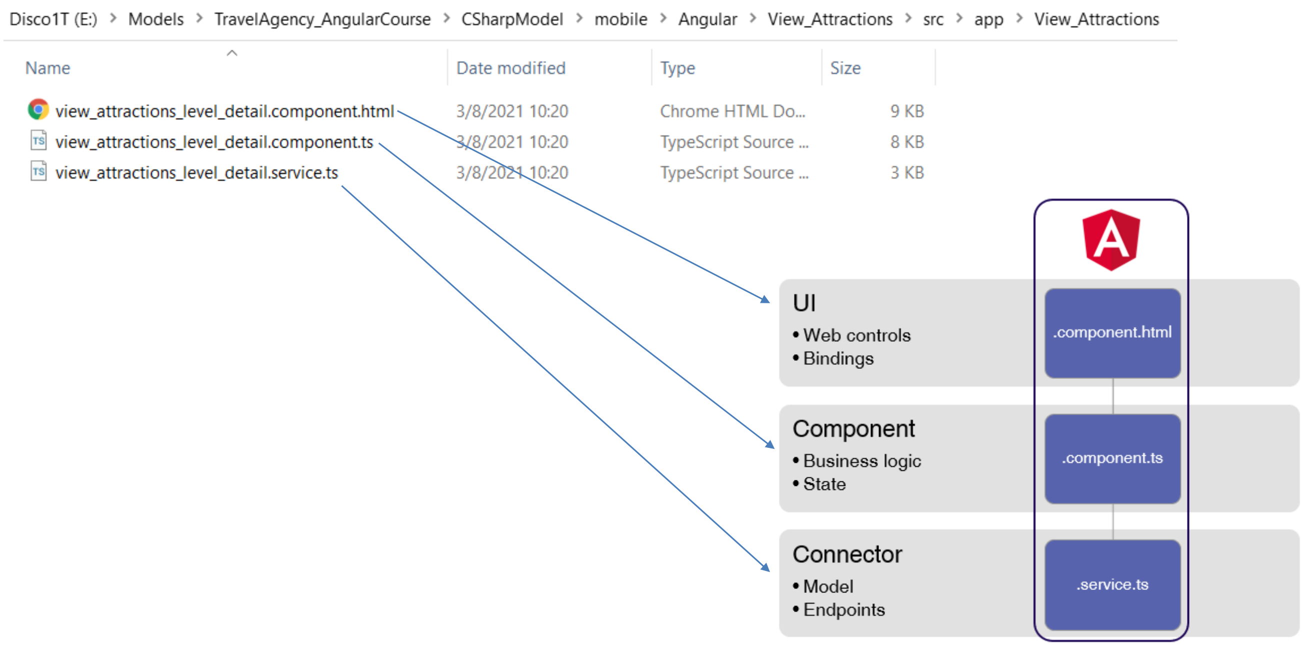
Name	Date modified	Type	Size
app	2/8/2021 14:26	File folder	
assets	2/8/2021 14:26	File folder	
environments	2/8/2021 14:26	File folder	
images	2/8/2021 14:26	File folder	
sass	2/8/2021 14:26	File folder	
translations	2/8/202		
index.html	2/8/202		
main.ts	2/8/202		
manifest.webmanifest	2/8/202		
polyfills.ts	16/6/20		
test.ts	16/6/20		
tsconfig.app.json	16/6/20		
tsconfig.spec.json	16/6/20		

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 Do...	4 KB
app.component.ts	2/8/2021 14:26	TypeScript Source ...	12 KB
app.module.ts	2/8/2021 14:26	TypeScript Source ...	3 KB
app.routing.ts	2/8/2021 14:26	TypeScript Source ...	1 KB
app.settings.ts	2/8/2021 14:26	TypeScript Source ...	3 KB
app-home.component.ts	2/8/2021 14:26	TypeScript Source ...	1 KB
common.module.ts	16/6/2021 14:31	TypeScript Source ...	4 KB
main.module.ts	2/8/2021 14:26	TypeScript Source ...	1 KB
shared.module.ts	2/8/2021 14:26	TypeScript Source ...	1 KB
shared-routing.module.ts	2/8/2021 14:26	TypeScript Source ...	1 KB
styles.css	16/6/2021 14:31	Cascading Style Sh...	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:) > Models > TravelAgency_AngularCourse > CSharpModel > mobile > Angular > View_Attractions > src > app > View_Attractions

Name	Date modified	Type	Size
view_attractions_level_detail.component.html	3/8/2021 10:20	Chrome HTML Do...	9 KB
view_attractions_level_detail.component.ts	3/8/2021 10:20	TypeScript Source ...	8 KB
view_attractions_level_detail.service.ts	3/8/2021 10:20	TypeScript Source ...	3 KB

UI
- Web controls
- Bindings

.component.html

Component
- Business logic
- State

.component.ts

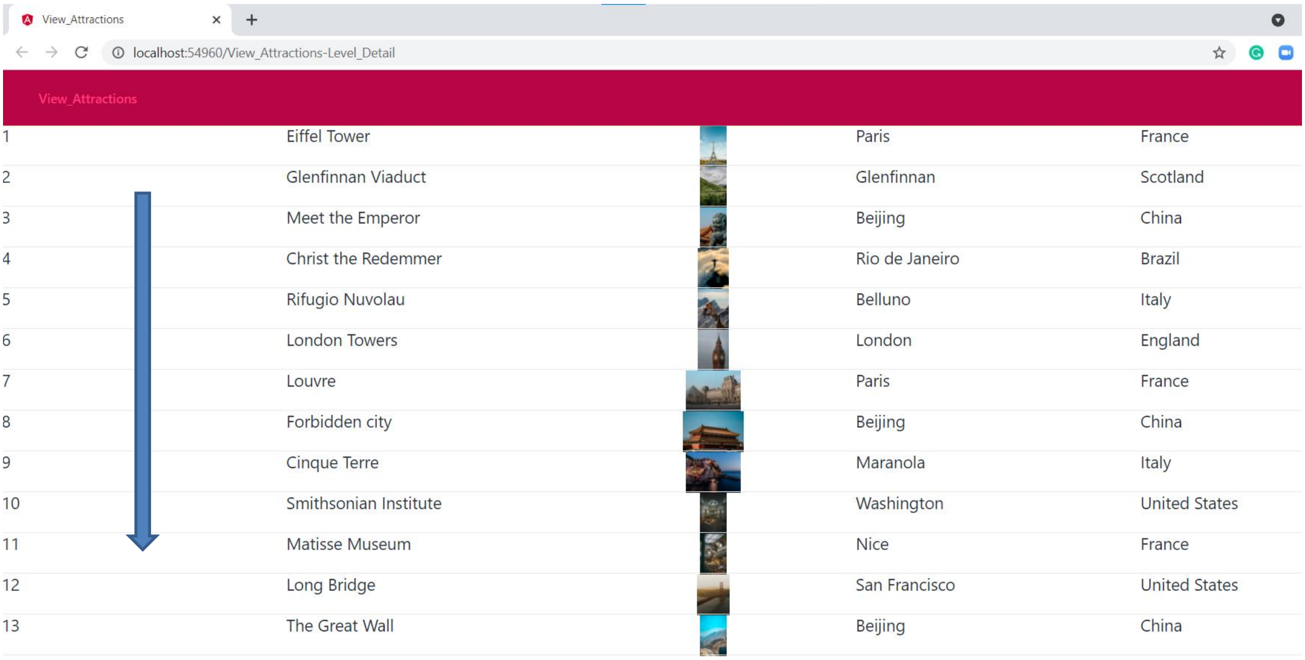
Connector
- Model
- Endpoints

.service.ts

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 components.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.



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

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