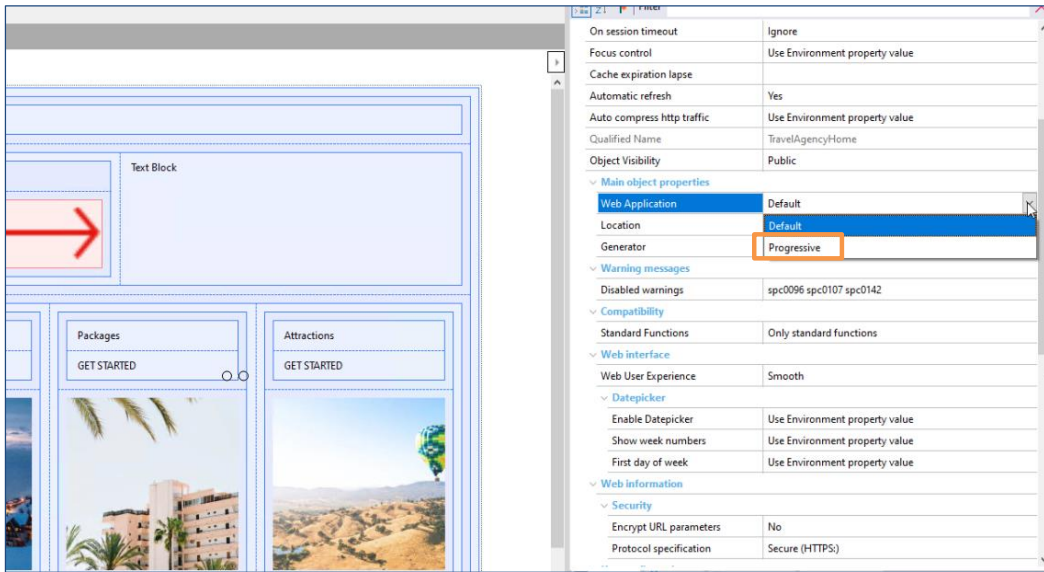


# Progressive Web Apps in GeneXus

How to Create Progressive Web Applications in GeneXus

*GeneXus*<sup>™</sup>

## Progressive Web App in GX

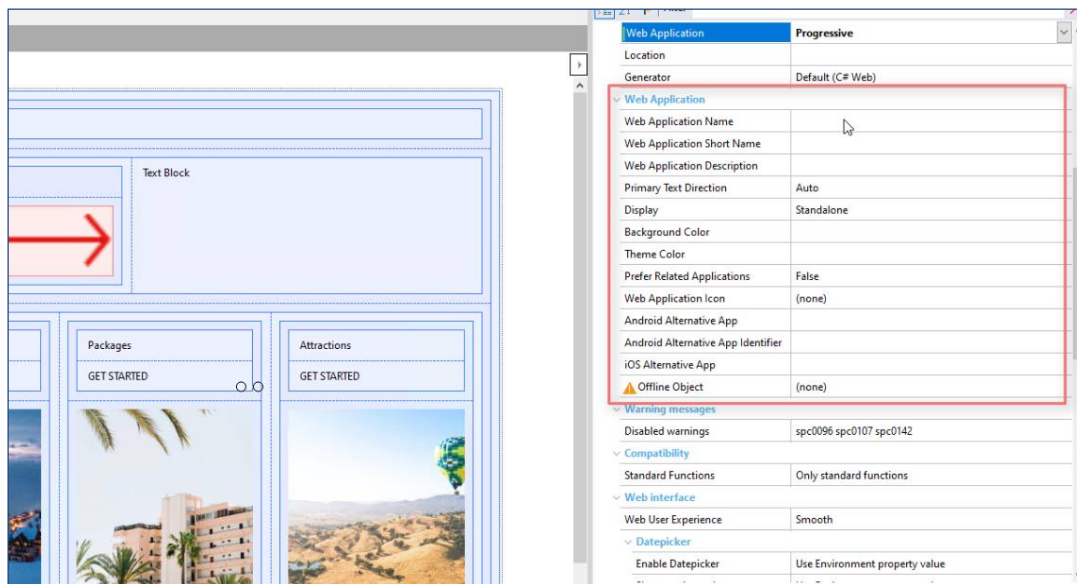


In this video, we will see how easy it is to create a Progressive Web Application with GeneXus.

For this example, we will use the Travel Agency application we have been working on. We select the TravelAgencyHome web panel. This web panel is a main object, because it has the Main program property set to true; that is, it can be executed independently, without the need to access it from the Developer Menu. This type of web panel can be configured as a progressive web application. (At the moment, only one main object per KB can be configured as a PWA.)

In the properties of this panel, we have a selection named Main object properties, and inside it the Web Application property. By default, it is displayed with the "Default" value, which means that a standard web application will be generated. If we change its value to "Progressive," we will indicate that we want a progressive web application to be generated. By changing this value, a new selection called Web Application will appear, from which we will be able to configure the necessary properties to create a progressive web application. Let's review these properties one by one.

## Progressive Web App in GX



**Web Application Name.** This property will contain the name assigned to the application; it will be the name we see in the list of applications or in the label of the access icon.

**Web Application Short Name.** This is the short name that will be assigned to the application. This is the name that will be displayed when there isn't enough space to display the full name; that is, the one we assigned to the previous property (Web Application Name).

**Web Application Description.** Here you can enter a brief overview of your web application.

**Primary Text Direction.** It determines the text direction for the Name, Short Name and Description properties. The options available are as follows:

- Auto: This is the default value. The browser will use the bi-directional Unicode algorithm to determine the direction of the text. For example, in English or Spanish it will be from left to right, while in Arabic or Hebrew it will be from right to left.
- From left to right: The text will be displayed from left to right.
- From right to left: The text will be displayed from right to left.

**Display.** In this property, we will define the display mode we want for our application. The options available are as follows:

**Standalone:** In this mode, the application will run in its own window, independent of the browser, simulating the experience of a native application and hiding elements of the user interface.

**FullScreen:** It will be displayed without any user interface, and will take the entire available area. These two options are the most used for this type of application. In addition, we have:

**Minimal:** This option will have minimal user interface elements. This mode will only be supported in Chrome with mobile devices. **Browser:** The display of the navigation window will be that of a standard browser. Note that if you choose this option, it will not allow you to install the application on the device. For these reasons, this option is not recommended unless there is a particular reason to use it.

**Background Color.** Here we can assign the background color of the application when it is starting. It is important to assign a color related to the site, as this helps create a smooth transition between the start of the web application and the loading of the content.

**Theme Color.** The color selected or entered in this property will be the color of the elements of the application's user interface and its toolbar, such as the address bar.

**Prefer Related Applications.** If we set this property to true, we will indicate that downloading the native application is recommended over the web application we are running. As we access the site, a dialog box will appear at the bottom of the screen, suggesting that the native application we indicate should be installed. We will see where to enter this path.

**Web Application Icon.** In this property, we will have to select the access icon that will be shown in the device start screen.

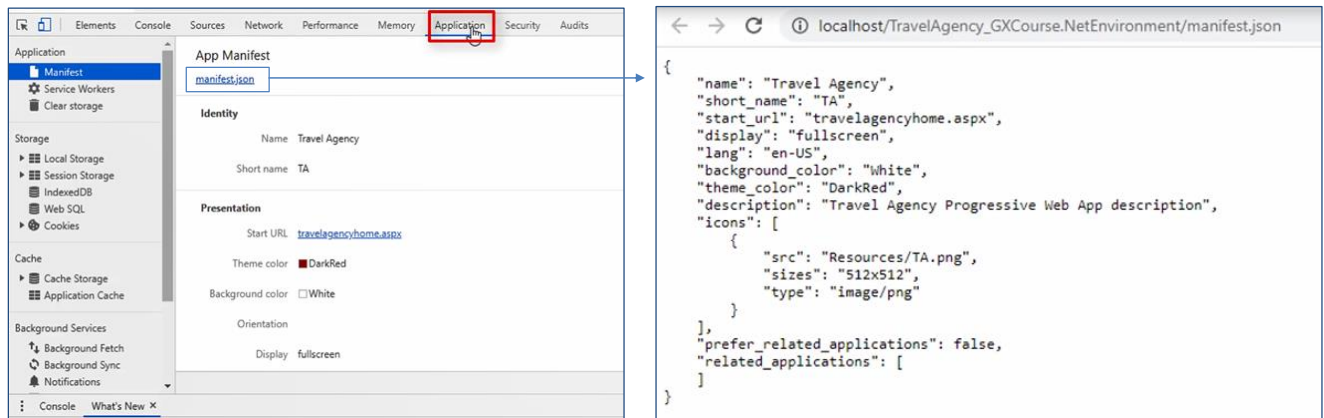
**Android Alternative App.** If the property "Prefer Related Applications = true," we must indicate the URL to download the native Android application. It can be a Google Play URL or a custom one where the application is hosted. Assuming that we have already published the native application in the Play Store, to find out the URL we simply access the application and copy the URL from there.

**Android Alternative App Identifier.** In this property, we can indicate the ID of the native Android application to download. If we look at the URL of the previous property, we see that at the end there is an ID; this is the value we should enter here.

**iOS Alternative App.** If the property "Prefer Related Applications = true," here we can enter the URL to download the iOS application. For this example, we leave the property Prefer Related Applications set to false, so we will leave these last three fields empty.

**Offline Object.** Finally, here we must choose a web object, which will be displayed if we run the application without an Internet connection. It can be a web panel that we have already created or we can generate one specifically for it. In this case, we select the "About" panel.

## Progressive Web App in GX



With these simple steps, we have already configured our application to turn it into a progressive web application.

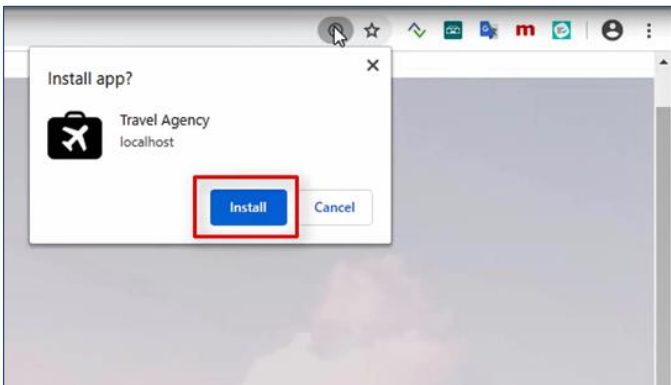
Let's try running the application with these changes. To do this, we save them and run the web panel directly by right-clicking on the TravelAgencyHome object and selecting the option "Run With This Only." In this case, we're using the Chrome browser.

Based on these settings we've made, GeneXus automatically creates a file called Manifest.Json. This manifest is one of the most important elements of the application. It is a file in JSON format, which contains the properties used to describe and generate the progressive web application.

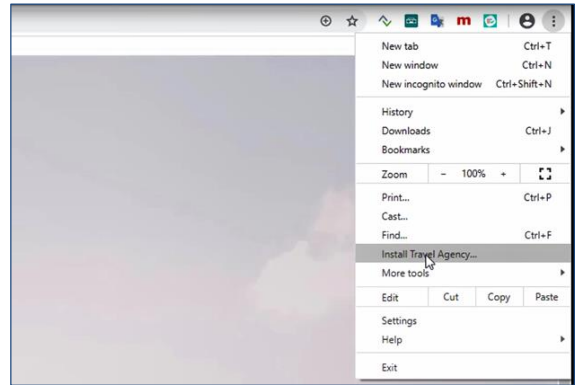
We can view this file. To do so, we access the Chrome Developer Tool (Ctrl.+Shift+i), and then select the Application tab. If we select this manifest, we see that the properties that we've just configured in GeneXus appear in JSON format: Name, short name, background color, theme color, etc. This can help us to make sure that everything is configured correctly.

## Install PWA

### Option 1



### Option 2



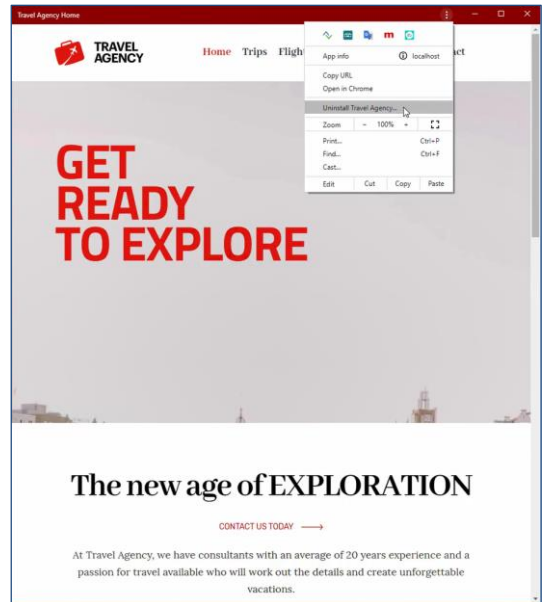
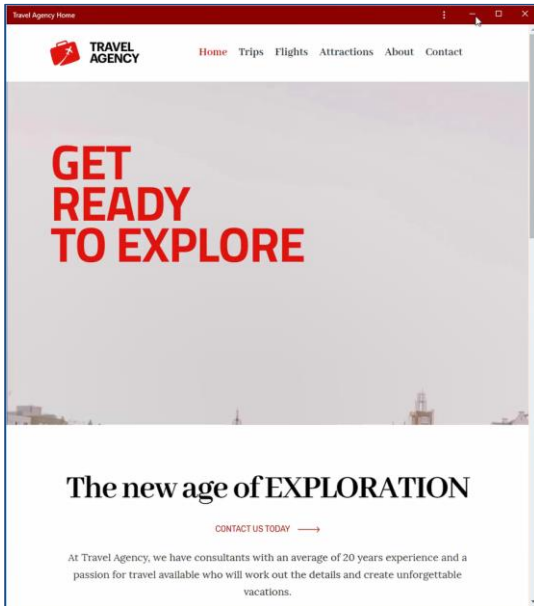
Returning to the site, we see that we have the option to install the application in the navigation bar.

The other option available to install the application is to use the browser menu and then install it.

Once installed, it generates a login icon on the device screen, which as we can see will have the name we assigned it in the “Web Application Name” property and the icon will be the one we entered in the “Web Application Icon” property.

The application opens automatically in a separate browser window and you can start using it.

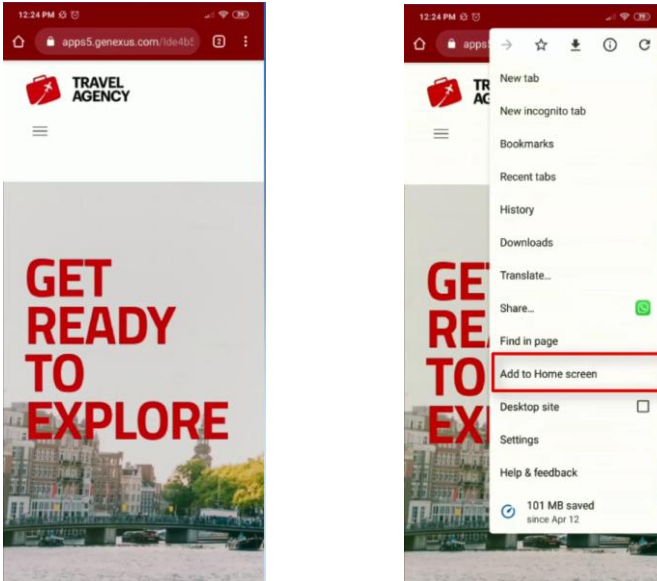
## Uninstall PWA



Note that this new window has a much cleaner, simpler design with no navigation bar, menus, or icons. It simulates the experience of a native application. Here we can clearly see the impact of the color of the theme we assign in the “Theme Color” property. This color should be related to the design of the site, to ensure overall consistency. Perhaps in this case a better option would have been to use a white color, but that will be left to each one's choice.

Within this same window we will have the option to uninstall the application.

## Install on smartphone



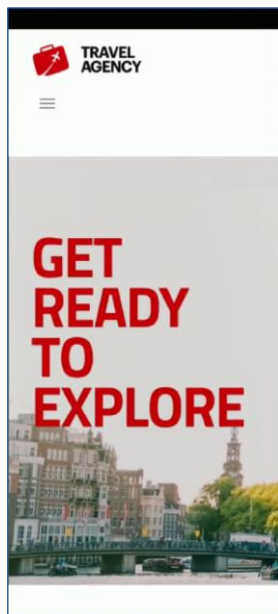
Now let's try running it from a mobile device; in this example, we will use an Android device with a Chrome browser.

We access the URL where we published the site, and then from the options menu we select Add to home screen. From that moment on, the application will be installed on our device.



## Running on smartphone

With  
Internet



Without  
Internet



Once the installation is finished, we can open the application and start using its functions. On the application loading screen we can already see several configurations we made from GeneXus: background color, theme color, icon image entered, and name.

Once we enter, we will be able to navigate and use its functions in the same way as any other application. On the main screen we see that a shortcut icon was generated with the name and image that we configured. From there we will access the application.

Let's now try one of the important features of progressive web applications, which is the ability to access content without a network connection. To do this, we disconnect the device from the Internet, leaving it in airplane mode. When we open the application, we see that it doesn't start in the home screen; instead, it shows the object we selected in the offline object property from GeneXus. In this case, the "About" web panel.

To delete this type of application, we will do it in the same way as any other native application. For example, we can do it directly from the home screen.

So far, we have seen how easy it is to create a progressive web application with GeneXus, simply by configuring some properties from a web panel of Main type. To learn more about this topic, you can visit our wiki.

# GeneXus™

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

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

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