

Web QuickStart



Welcome to Xojo, the easiest tool for creating your own apps. Xojo is made up of a rich set of graphical user interface objects, a modern object-oriented language, an integrated debugger, and a multi-platform compiler. Together they make up the Xojo Integrated Development Environment or IDE.

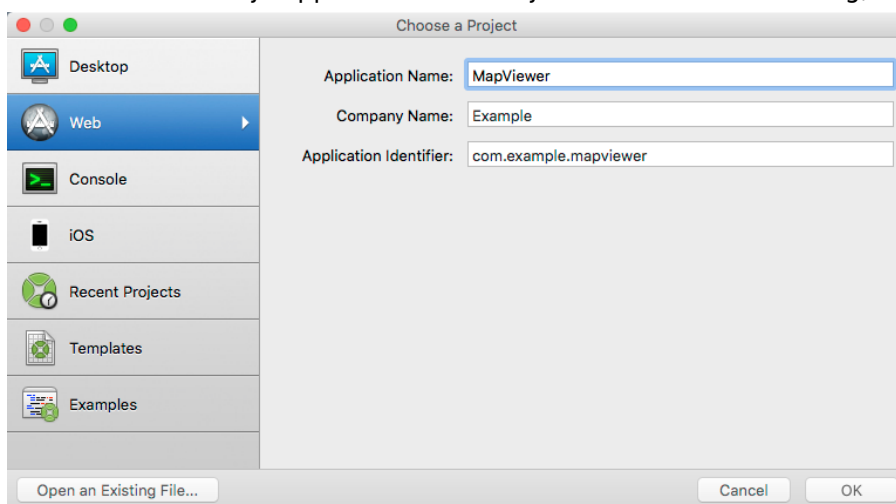
With the IDE, you can build your app's interface simply by dragging and dropping interface objects onto the app's web pages and dialogs. In this QuickStart, you will see how easy it is. Xojo provides you with all the tools you need to build virtually any app you can imagine.

This QuickStart is for people who are new to programming or new to Xojo. It will give you a gentle introduction to the Xojo development environment and lead you through the development of a working web app (a simple map viewer). It may take you up to 15 minutes to complete this QuickStart.

Getting Started

If you haven't done so already, now is the time to start Xojo.

1. Double-click the Xojo app icon to start Xojo. After it finishes loading, the Project Chooser window appears.



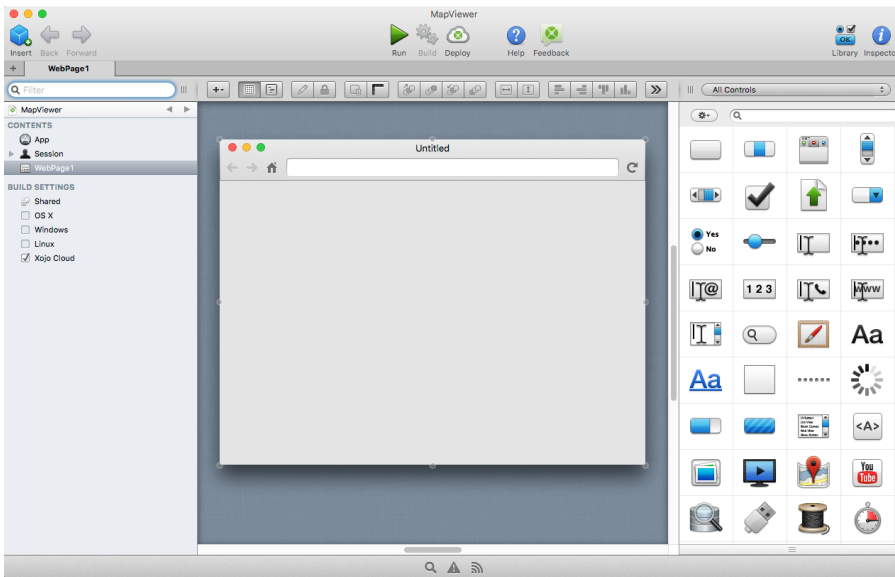
2. Xojo lets you build several different types of apps (Desktop, Web, Console and iOS). For this QuickStart, you are building a Web app, so click on Web.
3. You should now see three fields that need values: Application Name, Company Name and Application Identifier.
 - **Application Name** is the name of your app. This will be the filename of the actual app file that gets created.
 - **Company Name** is the name of your company. You may choose to leave this blank.
 - **Application Identifier** is a unique identifier for this app. It will automatically populate using what you enter

for the Application and Company Names, but you can also change it to whatever you want.

4. Enter "MapView" as the Application Name. You can leave Company Name as it is or change it.
5. Click OK to open the main Xojo window (called the Workspace), where you will begin designing your app.

Workspace

Xojo opens the Workspace with the default web page for your app selected in the Navigator and displayed in the Layout Editor.



Navigator: The area on the top left shows you all the items in your project. By default you can see WebPage1 (which is selected), the App object and the Session object. You use the Navigator to navigate within your project.

Layout Editor: The center area is the Layout Editor. You use the Layout Editor to design the user interface for the windows in your app. It shows the window and previews how it looks when the app runs. In this illustration, the window is blank because you haven't yet added any user interface controls from the Library.

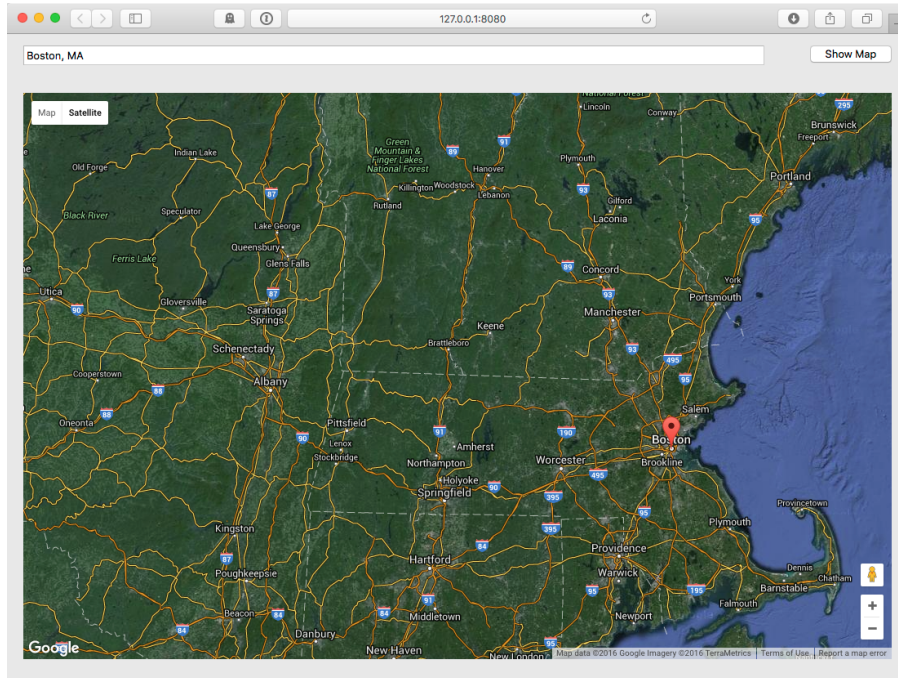
Library: The area on the right is the Library and shows the controls and interface elements that you can add to a window or to the project. You design the window by dragging controls from the Library to the window. You can also add a control to the window by double-clicking it. You can change how the controls display in the Library by clicking the small gear icon and choosing a different setting.

 If the Library is not visible, click the Library button on the toolbar to show it.

Inspector: Not shown in the above illustration is the Inspector, which allows you to see and change the properties for the selected control. This area of the Main window is shared with the Library. You can show the Inspector by clicking the Inspector button on the toolbar. The Inspector shows information about the selected item in the Navigator or Editor. The contents of the Inspector changes as you click on different items. You change an Inspector value by entering a new value in the field to the right of the field label.

Making the Map Viewer App

The best way to quickly learn how to use Xojo is to create a simple app. For this QuickStart, you will create a web app that shows you locations on a map.



A Xojo application consists of a collection of objects, called classes. Nearly everything in Xojo is a class, including your web pages and the controls on the web page. In the MapViewer project, you will use the default WebPage class to create your web page and you will add controls (user interface classes) to the web page to create the design.

MapViewer uses these controls:

- **Text Field:** A Text Field control is used to enter text. In this project, the location to display is typed into a Text Field at the top of the window.
- **Button:** A Web Button is used to trigger an action. The user clicks the button to load the location from the Text Field into the Map Viewer.
- **Map Viewer:** A Map Viewer is used to display a location on a map (it uses Google Maps behind the scenes). In this project, it is what displays the location entered in the Text Field.

The next sections walk you through creating the user interface and adding the necessary code to make the app work.

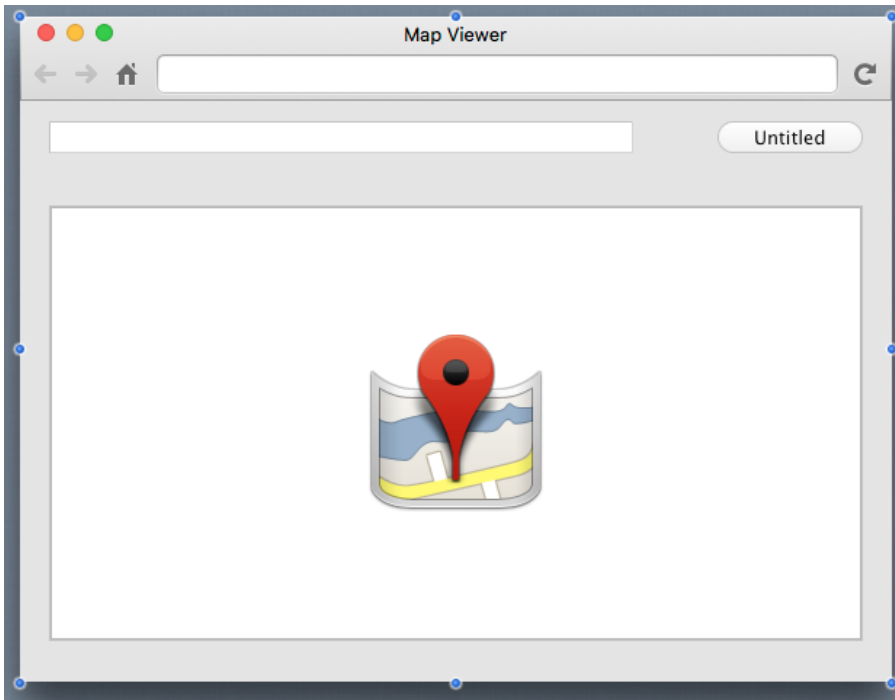


Building the User Interface

You should have Xojo running and WebPage1 open in the Layout Editor. Now you are ready to start adding controls to the web page.

1. Starting with the Text Field: In the Library, click on the Text Field and drag it to the top-left corner of the web page in the Layout Editor. As you get close to the edges of the page, you will see alignment indicators that help you position the control.
2. Next is the Button: In the Library, click on the Button and drag it to the top-right corner of the web page.
3. The final control is the Map Viewer: Drag the Map Viewer to the remaining empty area on the page.
4. The final step to do is to resize the Text Field so that it is larger. Click on it to show the selection handles. Click the center-right handle and drag it to the right until the alignment guides tell you it is close enough to the Button.

Your finished web page layout should look like this:



Setting the Properties

What is a Property?

A property is a value of a class. Changing property values allows you to change the behavior of the class. For this project, you want to change various properties for the view and the controls you added. Some of the things you need to do are:

- Rename all controls (and the window) so that they describe their behavior and are easy to refer to in code.
- Add a Caption to the Button.
- Set Locking properties so that the controls resize properly when the window is resized.

Inspector

The Inspector is used to change view and control properties. It shares the same area on the right of the Workspace as the Library. In order to show the Inspector, click the Inspector button on the toolbar or press $\text{⌘}+I$ (Ctrl+I on Windows and Linux).

Setting the Properties for the Web Page

If you haven't already, display the Inspector by clicking the Inspector button on the toolbar.

You need to change the Name and Title properties:

1. First, in the web page editing area, click on the title bar of the web page to select it. The Inspector now shows the properties for the web page.
2. In the Name field (located in the ID group), change the name from "WebPage1" to "MapLocationPage". Press Return to see the name change in the Navigator. Changing the Name makes it easier to identify project items

in the Navigator and is a good practice to start doing.

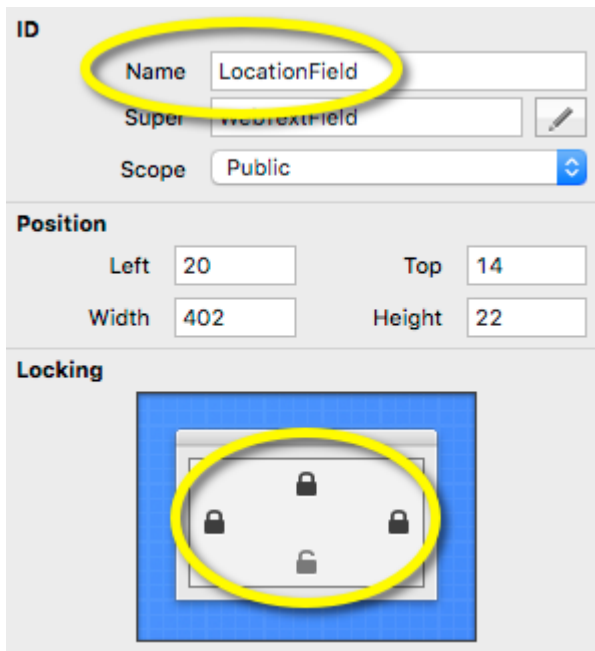
3. In the Title field (located in the Frame group), change the name from “Untitled” to “Map Viewer”. Press Return and you will see the name change in the title bar of the web page.

The screenshot shows the Xcode Inspector for a web page. It is divided into three sections: ID, Size, and Frame. In the ID section, the Name field is set to 'MapLocationPage' and the Super field is set to 'WebPage'. In the Size section, the Width is 600, Height is 400, Min Width is 600, and Min Height is 400. In the Frame section, the Title field is set to 'Map Viewer'. The Name and Title fields are circled in yellow.

Setting the Properties for the Text Field

The Text Field is where your user will enter the location to display. You want to change the following properties: Name and Locking

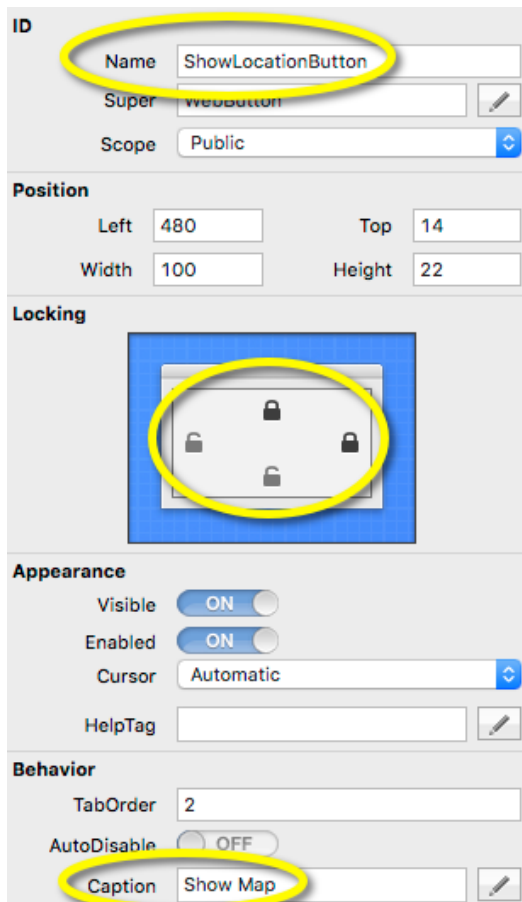
1. On the Web Page, select the TextField1 control. The Inspector changes to show the Text Field properties.
2. In the Name field, change the name from “TextField1” to “LocationField”. Press Return to see the name change in the Navigator. Changing the Name makes it easier to identify controls in the Navigator and when you are writing code. It is a good practice to start doing.
3. Now you need to make changes to the locking so that the Text Field gets larger or smaller as the web page size changes. In the Locking group look at the image that shows the web page with small locked icons for the top and left and small unlocked icons for bottom and right. Click the locks so that top, left and right are locked and bottom is unlocked.



Setting the Properties for the Button

Your users click the button to display the location. You need to change these properties: Name, Locking and Caption

1. On the Web Page, select the Button1 control. The Inspector changes to show the WebButton properties.
2. In the Name field, change the name from "Button1" to "ShowLocationButton". Press Return to see the name change in the Project List. Changing the Name makes it easier to identify controls in the Navigator and when you are writing code. It is a good practice to start doing.
3. Now you need to make changes to the locking so that the Button stays attached to the right side of the web page as the web page size changes. In the Locking group look at the image that shows the web page with small locked icons for the top and left and small unlocked icons for bottom and right. Click the locks so that top and right are locked and left and bottom are unlocked.
4. Lastly, you want to give your button a Caption. In the Caption field (located in the Behavior group), change the caption from "Untitled" to "Show Map".



ID

Name:

Super:


Scope:

Position

Left: Top:

Width: Height:

Locking



Appearance

Visible: ☒ ON

Enabled: ☒ ON

Cursor:

HelpTag:

Behavior

TabOrder:

AutoDisable: ☐ OFF

Caption:

Setting the Properties for the Map Viewer

The last user interface change you need to make is for the Map Viewer. Here you need to change these properties: Name and Locking

1. On the Web Page, select the MapViewer1 control. The Inspector changes to show the WebMapView properties.
2. In the Name field, change the name from "MapViewer1" to "LocationMap". Press Return to see the name change in the Navigator. Changing the Name makes it easier to identify controls in the Navigator and when you are writing code. It is a good practice to start doing.
3. Finally, you need to make changes to the locking so that the Map continues to fill the web page as its size changes. In the Locking group look at the image that shows the web page with small locked icons for the top and left and small unlocked icons for bottom and right. Click the locks so that top, bottom, left and right are locked.

The screenshot shows the Xojo IDE's Properties window for a 'LocationMap' object. The 'ID' section has 'Name' set to 'LocationMap', 'Super' set to 'WebMapView', and 'Scope' set to 'Public'. The 'Position' section shows 'Left' as 20, 'Top' as 72, 'Width' as 560, and 'Height' as 300. The 'Locking' section shows a preview of a map with four lock icons, one of which is circled in yellow.

Adding Code

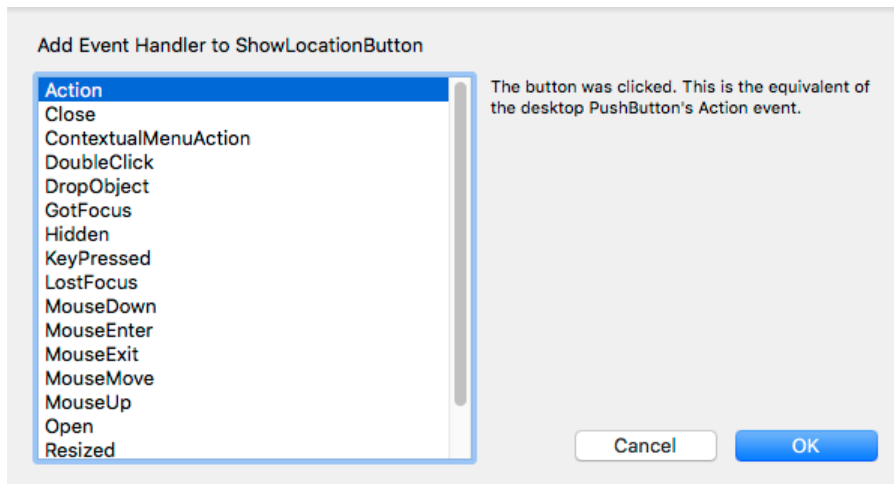
Your application is almost complete. Now it is time to add the code that will tell the MapViewer the location to display. Xojo uses an object-oriented programming language that is quite easy to learn. You need only four lines of code to finish your project!

The steps you need to do are:

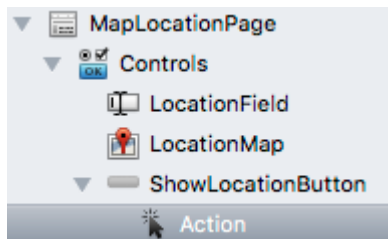
1. Figure out when your user clicks the ShowLocationButton, called “Show Map” on the web page.
2. Get the location that your user entered in LocationField.
3. Have LocationMap display the location.

Follow these steps to add the code:

1. On the web page, double-click the ShowLocationButton control, labelled “Show Map”. The Add Event Handler window appears. When a user clicks on a WebButton, the code in its Action event handler is run. This means you want to add your code to the Action event handler, so select Action from the Event Handler list and click OK.



Notice the Navigator updates to show the Action event underneath the ShowLocationButton control and the Code Editor displays.



This step solves your first problem of when to know when the user clicks on the ShowLocationButton.

2. Now you need to get the location and there are two parts to this:

1. The location that the user typed is stored in a property of the Text Field class called Text. You access property values by specifying the name of the class, followed by a "." and then the name of the property. In this case the Text Field is called LocationField so you write: LocationField.Text
2. You need a convert this text into an actual location using the WebMapLocation class. These two steps are accomplished with this code:

```
Dim location As New WebMapLocation location.Address = LocationField.Text
```

3. The last step is to have LocationMap display the location. This is done by calling two methods on the class: GoToLocation and AddLocation GoToLocation moves the map to a location and AddLocation drops a pin on the location. Your code looks like this:

```
LocationMap.GoToLocation(location)
LocationMap.AddLocation(location)
```

4. Now you can go ahead and add this code to the Code Editor. Start by clicking in the white space below the Action() event name and then type this code (as practice, type the code rather than copy and pasting it):

```
Dim location As New WebMapLocation
location.Address = LocationField.Text
LocationMap.GoToLocation(location)
```

```
LocationMap.AddLocation(location)</code>
```

That's it! Your first web application is complete.

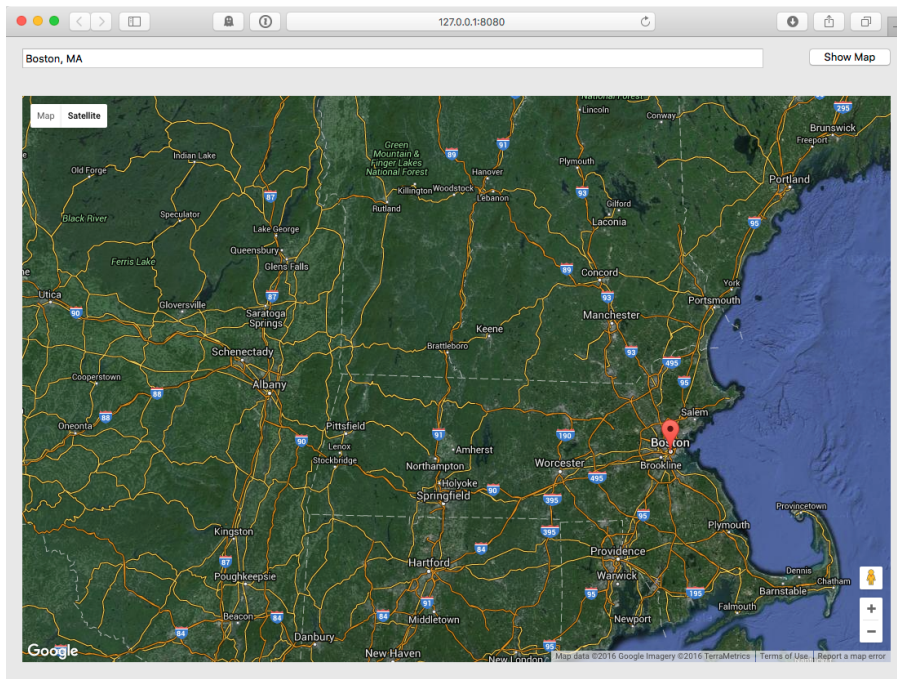
Running the App

You should save your work periodically and always before running your project. To do so:

1. Save the project by choosing File → Save.
2. Name the project “MapViewer” and click Save.

Now you can test your finished application:

1. Click the Run button in the toolbar to run your project. The web application opens in the default web browser.
2. Type a location of your choice, perhaps your home city, and click the “Show Map” button.
3. You will see a map of the location with a pin dropped at the precise location.
4. When you are finished experimenting with Map Viewer application, you can close the browser (tab or window depending on how it launched) to return to Xojo.



What's Next

The QuickStart has introduced you to Xojo. You've learned how to make your first app by designing a window, adding controls and writing code. You should next work through the Web Tutorial and then explore the [User Guide](#) and [Reference Guide](#) to continue learning how to create great apps using Xojo.