2
Enhanced Prototyping Recipes

In this chapter, you will cover several advanced Axure prototype recipes. You will learn the following:

- Adding external HTML and CSS files
- Incorporating inline frames
- Embedding external videos
- Including WordPress blog interactions
- Adding Google photo spheres
- Product visualization with flyout zoom
- Using Google Maps with Geolocation
- Leveraging social media logos – Facebook, Twitter, and Pinterest
- Adding app store badges – Apple iTunes and Google Play

Introduction

As your confidence and comfort level with Axure increases, you may want to add even more robust interactions to your prototypes. You can accomplish this by leveraging advanced concepts within Axure as well as utilizing external libraries and frameworks. In this chapter, you will first see how to use external HTML and CSS files to enhance an Axure prototype. You will explore recipes that detail how to interact with YouTube, WordPress blogs, and visualize products with flyout zoom. We'll also leverage social media logos and badges.

Adding external HTML and CSS files

Let's say that you started working on a design using HTML and CSS. You would like to include these in your Axure prototype. There are several ways to accomplish this task. In this recipe, you will explore a couple of those methods.

Getting ready

To complete this recipe, you will need to have external HTML and CSS files.

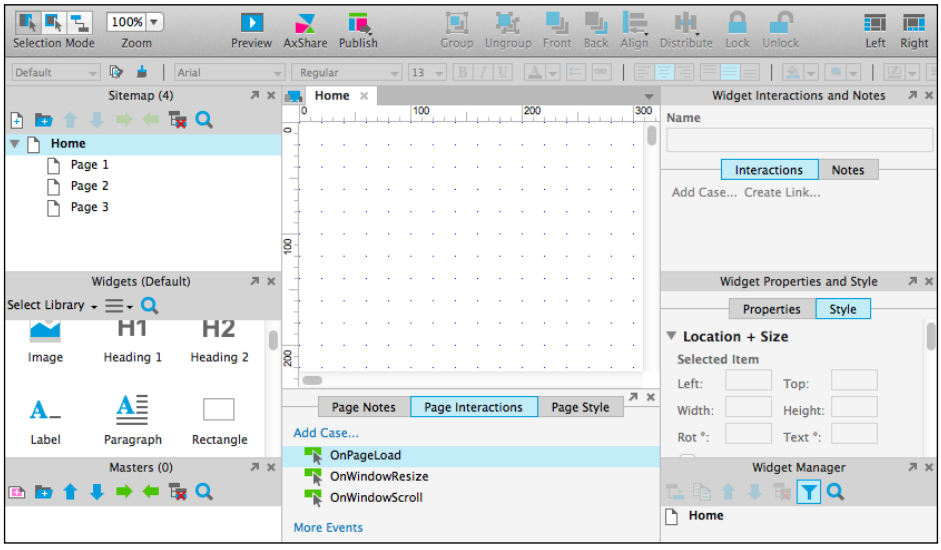
How to do it...

You will now link a prototype to the external HTML and CSS files.

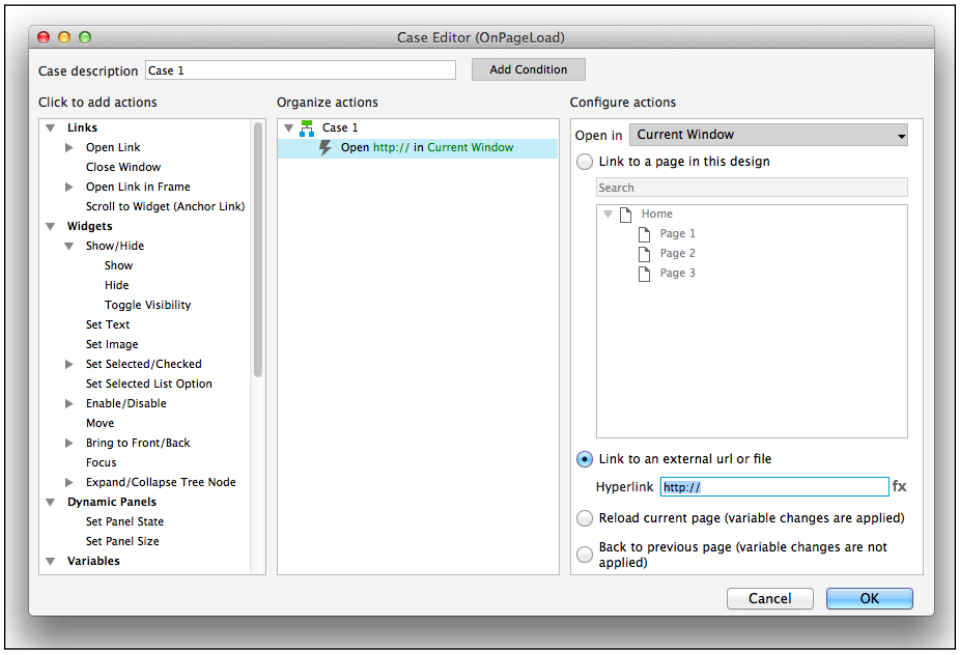
1. Start Axure and under **Create New** select **RP File**.

> 
>
> If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

2. Double-click on **Home** in the **Sitemap** window to select the home page. The blank home page will be displayed in the wireframe pane.
3. Next, click on the **Page Interactions** tab in the **Page Properties** pane. Double-click on **OnPageLoad**.



4. Expand the **Links** button under **Click to add actions** on the **Case Editor (OnPageLoad)** pop up. Click on **Open Link**.
5. You will see, under **Organize actions**, that **Case 1** has been updated to have the action **Open http:// in Current Window**.
6. Under **Configure actions**, click on the radio button next to **Link to an external URL or file**.

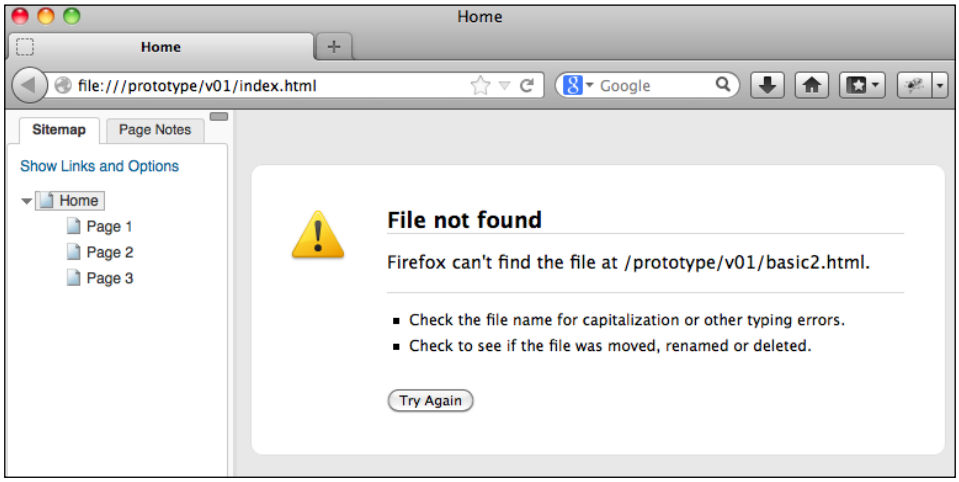


7. In the **Hyperlink** field, enter the path to the external `HTML` file.
8. Click on **OK**.
9. You can now choose to preview or save a copy of the prototype. To preview the prototype, click on the **Preview** button in the toolbar. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish** and clicking on **Generate HTML Files...**.

How it works...

When you view the Axure prototype, the link that is shown in the site map for our given page links to our external HTML file. That external HTML file references the external CSS file.

Also, make sure to have your HTML and CSS files in the same folder as your prototype. If the prototype cannot find the HTML and CSS files when it loads, you will see an error message similar to the one in the following screenshot:



There's more...

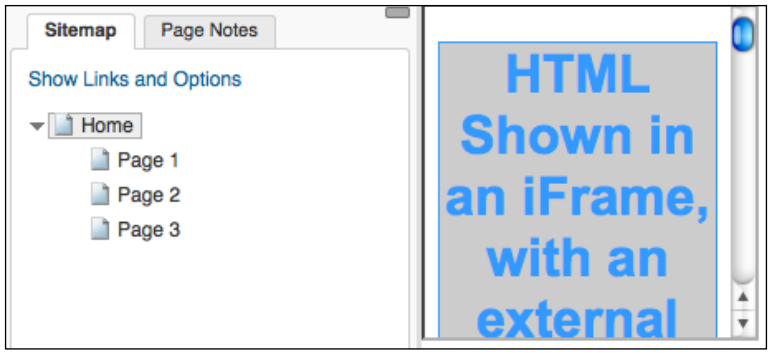
In this recipe, you learned how to link your prototype to external HTML and CSS files. There is another approach to accomplishing this task if you just want to link external HTML and CSS files. This approach involves modifying the Axure-generated HTML page. You would reference the external CSS file in the `head` tag and add your additional HTML in the `body` tag.

Incorporating inline frames

Leveraging the inline frame widget in your prototypes opens a world of possibilities. The inline frame widget allows you to link external HTML pages in your prototypes.

Getting ready

In this recipe, you will use an inline frame widget to open an external HTML page.



How to do it...

1. Start Axure and under **Create New** select **RP File**.

If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

2. While holding down the mouse button, drag the **Inline Frame** widget and place it at these coordinates on the wireframe: (0,0).
3. Click on the **Inline Frame** widget on the wireframe. In the top right corner of the wireframe, you will see two fields marked **w:** and **h:**. These are for the width and height of the **Inline Frame** widget. Adjust the width and height as needed.
4. Double-click on the **Inline Frame** widget on the wireframe.
5. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.
6. Click in the **Hyperlink** field and enter the filename of your HTML document.
7. Click on **OK**.
8. Click on the **Publish** button on the toolbar and then on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish** and clicking on **Generate HTML Files...**.
9. Note the directory where the prototype is saved.
10. Copy your HTML document referenced in step 6 and save it in the same directory as your prototype.
11. In your prototype directory, you will see a file labeled `start.html`. Axure created this file when the prototype was generated.
12. Open `start.html` in a browser to view the prototype.

How it works...

In this recipe, you use the inline frame widget and set the default target to link to an external file. When the prototype is loaded into the browser, the default HTML file is shown in the inline frame widget.

Embedding external videos

Embedding external video can make your prototype more vibrant and interactive. This recipe will show you how to embed YouTube and Vimeo videos into your prototype.

Getting ready

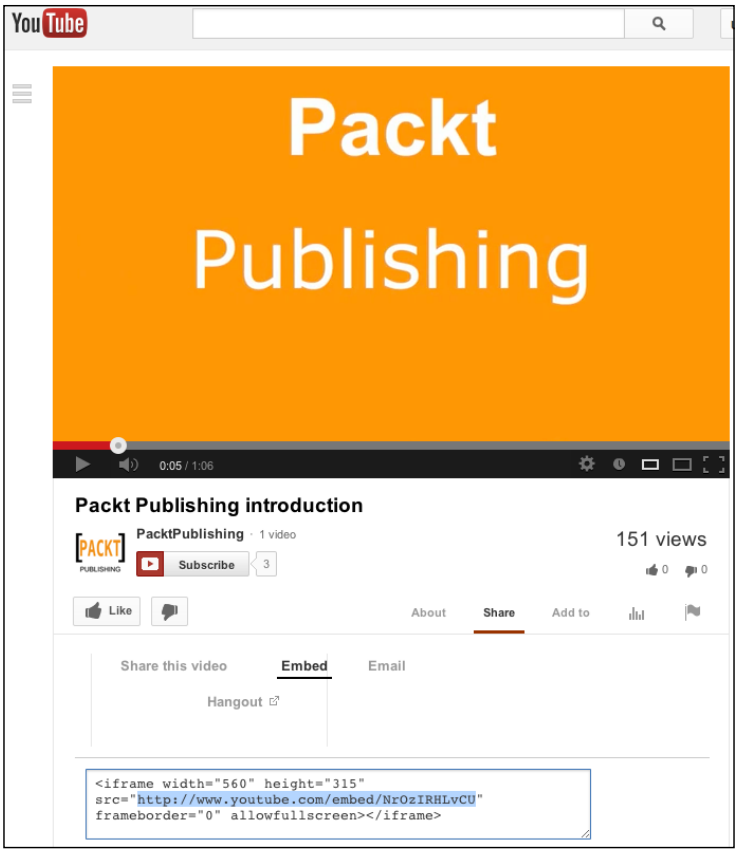
For this recipe, all you will need is a YouTube URL and a Vimeo URL for the videos you wish to link to. If you do not have a YouTube or Vimeo URL, don't worry; we will walk you though the process of getting what you need in this recipe.

How to do it...

In this recipe, you will first use an inline frame widget to open an external YouTube video. Next, you will view the YouTube video in your prototype. Then, you will add a second inline frame widget on another page in the prototype to open an external Vimeo video. Here's how you do it:

1. Open a browser and go to `http://www.youtube.com`.
2. Search for and select the video you would like to include in your prototype.
3. On the YouTube page for your selected video, click on the **Share** menu item under the video.
4. Next, click on the **Embed** menu item.
5. Copy the URL shown in the `embed` HTML tag provided. It will typically be shown on the second line starting with `src=`.

The `src` attribute is used to specify the location or URL of a video file. Please make sure to add `http:` at the beginning of the URL if it is not given in the `embed` HTML tag.

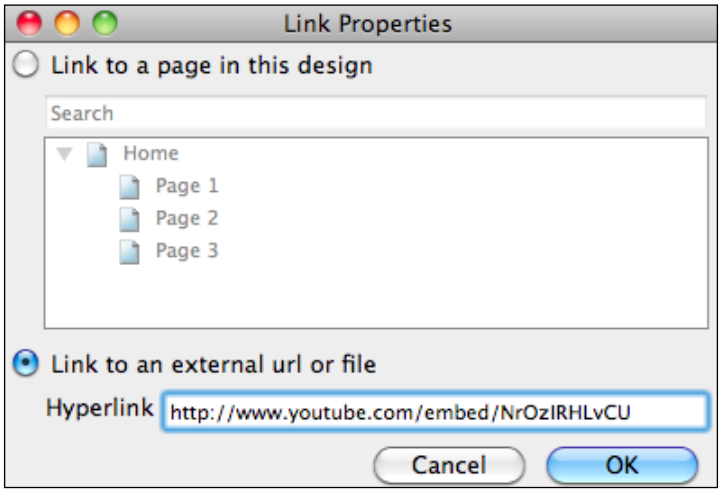


6. Start Axure and click on **Create New RP Document**.

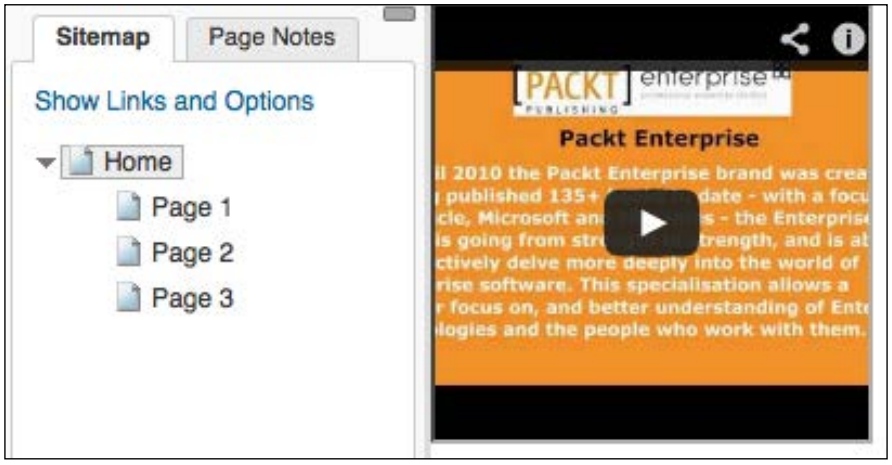
If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

7. While holding down the mouse button, drag the **Inline Frame** widget and place it at the coordinates (0,0) on the wireframe.
8. Click on the **Inline Frame** widget on the wireframe.
9. Double-click on the **Inline Frame** widget on the wireframe.
10. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.

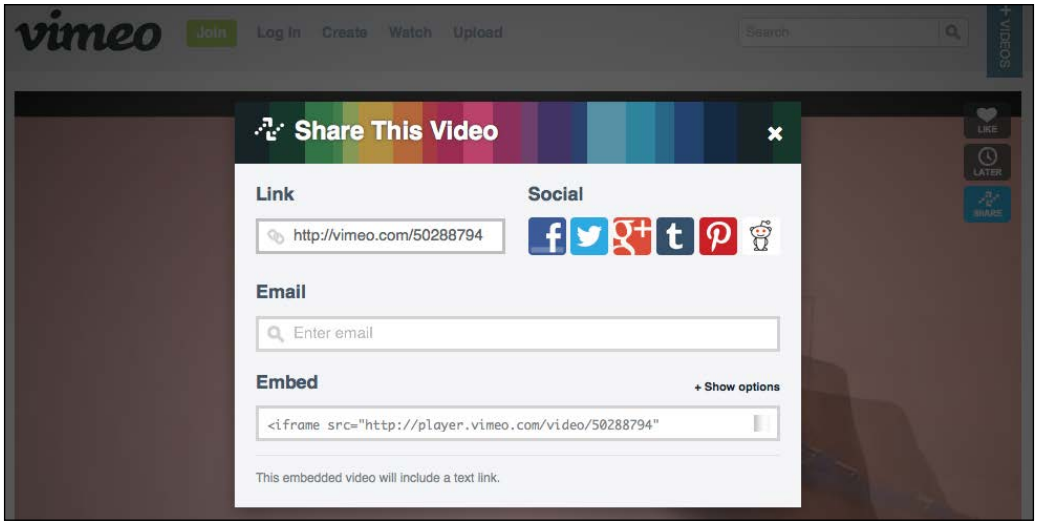
11. Click in the **Hyperlink** field and enter the URL for the YouTube video you copied in step 5.



12. Click on **OK**.
13. You can now choose to preview or save a copy of the prototype. To preview the prototype, click on the **Preview** button in the toolbar. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu, and clicking on **Publish** and clicking on **Generate HTML Files...**. Now both videos will be viewable in the prototype.



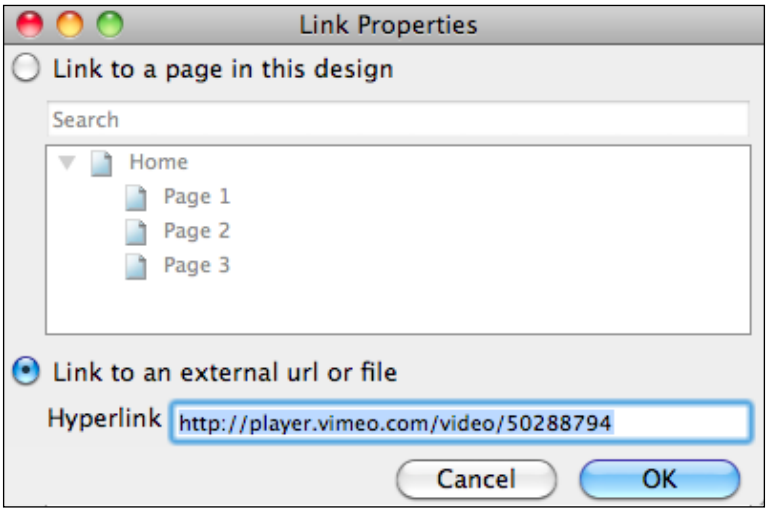
14. Next, you will add an **Inline Frame** widget to **Page 1** and link to a Vimeo video.
15. Open a browser and go to `http://vimeo.com/`.
16. Search for and select the video you would like to include in your prototype.
17. On the Vimeo page for your selected video, click on the **Share** menu item to the right of the video.
18. An overlay titled **Share This Video** will appear.
19. Copy the URL shown in the `embed` tag provided. It will typically be shown in the line starting with `src=`.



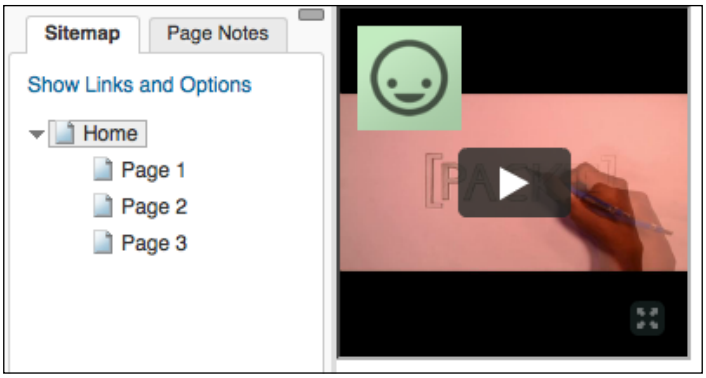
20. In Axure, click on **Page 1** in the **Sitemap** window.
21. While holding down the mouse button, drag the **Inline Frame** widget and place it at coordinates (0,0) on the wireframe.
22. Click on the **Inline Frame** widget on the wireframe.
23. Double-click on the **Inline Frame** widget on the wireframe.
24. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.

25. Click in the **Hyperlink** field and enter the URL for the Vimeo video you copied in step 5.

Please include `http://` with your URL.



26. Click on **OK**.
27. You can now choose to preview or save a copy of the prototype. To preview the prototype, click on the **Preview** button in the toolbar. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish**, and clicking on **Generate HTML Files...**. Now both videos will be viewable in the prototype.
28. You can view the YouTube video by clicking on the **Home** page in the prototype. To view the Vimeo video, click on **Page 1** in the prototype.



How it works...

In this recipe, you use the inline frame widget and set the default target to link to the embedded URL you copied from the selected YouTube or Vimeo video. When the prototype is loaded into the browser, the YouTube video is shown in the inline frame widget on the **Home** page. Clicking on the **Page 1** page shows the Vimeo video. The YouTube and Vimeo links provide all of the media controls for the videos.

Including WordPress blog interactions

Adding live feeds to WordPress blogs is a nice way of providing realistic user interactions for flows that include blogs. This recipe will show you how to add a working WordPress blog feed to your prototypes.

How to do it...

In this recipe, you will perform the following steps to link your Axure prototype to the WordPress feed:

1. Start Axure and under **Create New** select **RP File**.

If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

2. While holding down the mouse button, drag the **Inline Frame** widget and place at this coordinates on the wireframe: (0,0).
3. With the **Inline Frame** widget selected, you will see two fields marked **w:** and **h:** in the top right-hand of the wireframe. These are for the width and height of the **Inline Frame** widget. Enter `800` in the width field and `500` in the height field.

4. Click on the **Inline Frame** widget on the wireframe.
5. Double-click on the **Inline Frame** widget on the wireframe.
6. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.
7. Click in the **Hyperlink** field and enter the link to the official WordPress feed at `http://en.blog.wordpress.com/`.
8. You can now choose to preview or save a copy of the prototype. To preview the prototype, click on the **Preview** button in the toolbar. To save a copy of the prototype, click on the **Publish** button on the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish** and clicking on **Generate HTML Files...**.

How it works...

The key to this recipe is using the inline frame widget with a default target set to the WordPress feed link at `http://en.blog.wordpress.com/`. When the prototype is running, the inline frame widget displays the link to the WordPress feed.

There's more...

You can learn more about WordPress feeds at `http://en.support.wordpress.com/feeds/`. Another popular method of using feeds on websites today is to use **Google FeedBurner**. More information about Google FeedBurner can be found at `https://support.google.com/feedburner/answer/79408`.

Adding Google photo spheres

An update to Google's mobile operating system Jelly Bean (that is, Android 4.2) provides the ability to capture photo spheres. Photo spheres are 360-degree panoramic images. Google recently shared with developers a viewer that allows everyone to enjoy photo spheres on their website. This recipe will show you how to add Google photo spheres to your prototypes. See `https://developers.google.com/photo-sphere/web/` for more information.

You may have to register as an Android developer to access the preceding URL. To register as an Android developer, visit `https://support.google.com/googleplay/android-developer/answer/113468?hl=en`.

Getting ready

To complete this recipe, you will need a photo sphere image. You may use the Hello World example provided by Google. If you have a photo sphere image, we'll show you where to reference it in your external HTML.

How to do it...

In this recipe, you will use an inline frame widget to display a Google photo sphere image in an Axure prototype. Perform the following steps:

1. If you have a photo sphere image, copy the file to your Axure prototype directory.
2. Using your favorite text editor, create a blank `.html` file.
3. Place the following HTML code into the blank `.html` file. Change the reference in `imageURL` to point to the local file you copied to your Axure prototype directory in step 1.

```
<html>
  <head>
    <title>Photo Sphere Viewer: Hello World</title>
    <script type="text/javascript"
      src="https://apis.google.com/js/plusone.js"></script>
    <style> body {font-family: Arial,Verdana}</style>
  </head>
  <body>
    <h1>Photo Sphere Viewer: Hello World</h1>
    <g:panoembed imageURL="https://lh5.googleusercontent.com/-
  kr97Eucg6sM/UKGEuvo_eBI/AAAAAAAAi0s/adq8uqyhm_k/
    photo.jpg" fullsize="4096,2048" croppedsize="4096,1380"
      offset="0,480" displaysize="600,400"/>
    <script>
      gapi.panoembed.go();
    </script>
  </body>
</html>
```

4. Start Axure and under **Create New** select **RP File**.

If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

5. While holding down the mouse button, drag the **Inline Frame** widget and place it at coordinates (0,0) on the wireframe.
6. With the **Inline Frame** widget selected, you will see two fields marked **w:** and **h:** in the top right-hand of the wireframe. These are for the width and height of the **Inline Frame** widget. Enter `800` in the width field and `500` in the height field.

7. Click on the **Inline Frame** widget on the wireframe.
8. Double-click on the **Inline Frame** widget on the wireframe.
9. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.
10. Click in the **Hyperlink** field and enter the filename of the HTML document you created in step 2.
11. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish** and clicking on **Generate HTML Files...**.
12. Note the directory where the prototype is saved.
13. Copy the HTML document you created in step 3 and save it in the same directory as your prototype.
14. In your prototype directory, open `start.html` in a browser to view the prototype.

How it works...

In this recipe, you used the inline frame widget with an external HTML file set as the default target. The external HTML file contains the HTML code that references the JavaScript file for the Google photo sphere viewer. You place the external HTML file and photo sphere file in the same folder as the root of the prototype output. When the prototype is run, the inline frame widget runs the external HTML file that calls the photo sphere JavaScript that in turn renders the photo sphere image. When the user holds down the left mouse button and moves the cursor, the photo sphere image rotates in the viewer shown in the inline frame widget.

Product visualization with flyout zoom

Product visualization with flyout zoom is popular on many websites. Here is how it works: a product image is displayed in the browser window, and when the user hovers their mouse pointer over a portion of the product image, the portion of the image under the mouse is magnified in a flyout window.

There are several commercial solutions available (for example, Adobe Scene 7) that provide this effect. There also are free open source solutions as well. In this recipe, you will use an open source flyout zoom solution.

Getting ready

To complete this recipe, you will need the **MojoZoom** CSS and JavaScript files available at `http://www.nihilogic.dk/labs/mojozoom/`, an image saved as two different files, and Axure. For this recipe, you are using a resolution of 400 x 300 pixels for the small image file and a resolution of 1200 x 900 pixels for the large image file. You can learn more about MojoZoom at `http://www.nihilogic.dk/labs/mojozoom/`.

How to do it...

In this recipe, you are going to explore using **MojoZoom Javascript Image Zoom**, which is free and licensed under the MPL license.

1. Start Axure and under **Create New** select **RP File**.

If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

2. While holding down the mouse button, drag the **Inline Frame** widget and place it at coordinates (0,0) on the wireframe.
3. With the **Inline Frame** widget selected, you will see two fields marked **w:** and **h:** in the top right-hand of the wireframe. These are for the width and height of the **Inline Frame** widget. Enter `800` in the width field and `500` in the height field.
4. Click on the **Inline Frame** widget on the wireframe.
5. Double-click on the **Inline Frame** widget on the wireframe.
6. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.
7. Click in the **Hyperlink** field and enter `mojozoom.html`.

8. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish** and clicking on **Generate HTML Files...**.
9. Note the directory where the prototype is saved.
10. Download the MojoZoom ZIP file from `http://www.nihilogic.dk/labs/mojozoom/`. Unzip the `MojoZoom.zip` file and copy the `mojozoom.css` and `mojozoom.js` files to your prototype directory.
11. Create two image files of the same JPG image. A small image file with a resolution of 400 x 300 pixels and a large image file with a resolution of 1200 x 900 pixels. Name the files `small_image1_400x300.jpg` and `large_image1_1200x900.jpg`, respectively. Copy your small and large image files to your prototype directory.
12. Using your favorite text editor, create an HTML file named `mojozoom.html` with the following HTML code and save in your prototype directory:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Axure RP Prototyping Cookbook
      Zoom Demo</title>
    <script src="mojozoom.js"
      type="text/javascript" ></script>
    <link rel="stylesheet" href="mojozoom.css"
      type="text/css"></link>
  </head>

  <body>

    <img id="myimage" style="padding:0px;border:0px;"
      data-zoomsrc="large_image1_1200x900.jpg"
        src="small_image1_400x300.jpg">
    <div style="position: absolute; left: 0px; top: 0px;
      width: 400px; height: 299px; overflow: hidden;
        display: none;">
    <div class="mojozoom_marker" style="width: 90px;
      height: 90px; left: 308px; top: 152px;">
    <div class="mojozoom_fill"></div>
    <div class="mojozoom_border"></div>
    </div>
    </div>

  </body>

</html>
```

13. In your prototype directory, open `start.html` in a browser to view the prototype.

How it works...

In this recipe, you used the inline frame widget with an external HTML file set as the default target. The external HTML file contains the HTML code that references the CSS and JavaScript files for the flyout zoom solution. You place the external HTML file, image files, and flyout zoom CSS and JavaScript files in the same folder as the root of the prototype output. When the prototype is run, the inline frame widget runs the external HTML file that shows the small image, and the flyout zoom JavaScript renders the zoom window when the mouse pointer is over the image shown in the inline frame widget.

There's more...

There are many open source flyout zoom solutions available. This recipe has shown you an approach to integrate flyout zoom into your prototype using the MojoZoom solution. Feel free to explore other open source solutions as well.

Using Google Maps with Geolocation

You can also use Google Maps with Geolocation to create dynamic prototypes. You will leverage the latest Google Maps APIs to make our prototypes come to life. You will use the Google Maps Geolocation example found at `https://developers.google.com/maps/documentation/javascript/examples/map-geolocation`.

How to do it...

In this recipe, you will use an inline frame widget to utilize Google Maps Geolocation in an Axure prototype. Perform the following steps:

1. Using your favorite text editor, create a `.html` file containing the following `HTML` code and place it in your Axure prototype directory:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Geolocation</title>
    <meta name="viewport" content="initial-scale=1.0,
      user-scalable=no">
    <meta charset="utf-8">
    <link href="http://code.google.com/
      apis/maps/documentation/javascript/examples/
        default.css" rel="stylesheet">
    <!--
    Include the maps javascript with sensor=true because this code
is using a
```

```
    sensor (a GPS locator) to determine the user's location.
    See: https://developers.google.com/apis/maps/documentation/
javascript/basics#SpecifyingSensor
    -->
    <script src="https://maps.googleapis.com/maps
      /api/js?v=3.exp&sensor=true"></script>

    <script>
var map;

function initialize() {
  var mapOptions = {
    zoom: 6,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  map = new google.maps.Map(document.getElementById
    ('map-canvas'),
      mapOptions);

  // Try HTML5 geolocation
  if(navigator.geolocation) {
    navigator.geolocation.getCurrentPosition
      (function(position) {
      var pos = new google.maps.LatLng
        (position.coords.latitude,
          position.coords.longitude);

      var infowindow = new google.maps.InfoWindow({
        map: map,
        position: pos,
        content: 'Location found using HTML5.'
      });

      map.setCenter(pos);
    }, function() {
      handleNoGeolocation(true);
    });
  } else {
    // Browser doesn't support Geolocation
      handleNoGeolocation(false);
  }
}

function handleNoGeolocation(errorFlag) {
  if (errorFlag) {
```

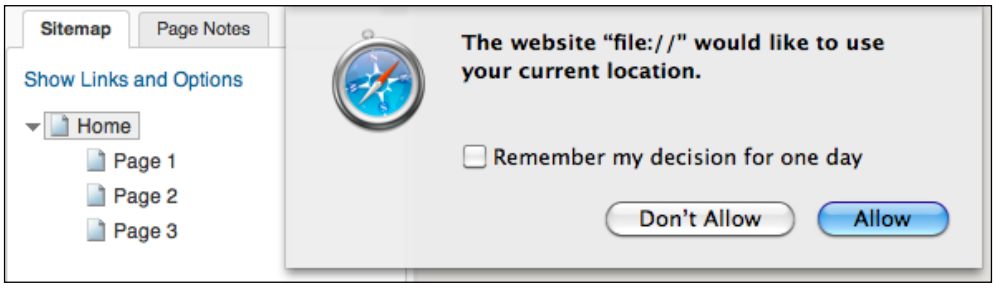
```
      var content = 'Error: The Geolocation service failed.';
    } else {
      var content = 'Error: Your browser doesn\
        't support geolocation.';
    }

    var options = {
      map: map,position:
        new google.maps.LatLng(60, 105),
      content: content
    };

    var infowindow = new google.maps.InfoWindow(options);
    map.setCenter(options.position);
  }

  google.maps.event.addDomListener
    (window, 'load', initialize);

      </script>
    </head>
    <body>
      <div id="map-canvas"></div>
    </body>
  </html>
```

2. Start Axure and under **Create New** select **RP File**.

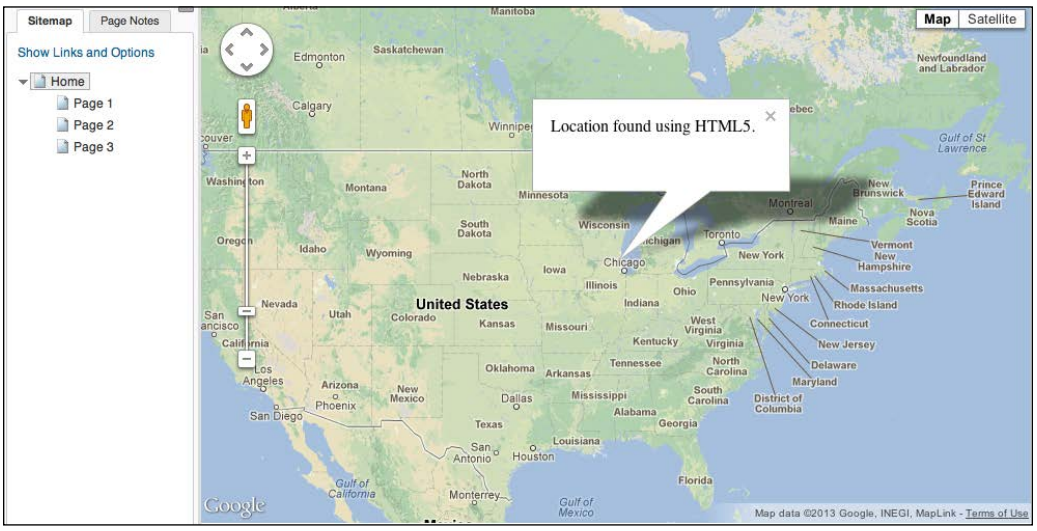
If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

3. While holding down the mouse button, drag the **Inline Frame** widget and place it at coordinates (0,0) on the wireframe.
4. With the **Inline Frame** widget selected, you will see two fields marked **w:** and **h:** in the top right-hand of the wireframe. These are for the width and height of the **Inline Frame** widget. Enter `800` in the width field and `500` in the height field.
5. Click on the **Inline Frame** widget on the wireframe.
6. Double-click on the **Inline Frame** widget on the wireframe.
7. In the **Link Properties** pop up click on the radio button next to **Link to an external URL or file**.

8. Click in the **Hyperlink** field and enter the filename of your HTML document created in step 1.
9. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and click on **Publish**, then click on **Generate HTML Files...**.
10. Note the directory where the prototype is saved.
11. Copy your HTML document created in step 1 and save it in the same directory as your prototype.
12. In your prototype directory, open `start.html` in a browser to view the prototype.
13. Click on **Allow** to enable the browser to share your current location with Google to update the Google Maps.



14. The map will now be displayed and you can move and zoom in on the map.



How it works...

In this recipe, you used the inline frame widget with an external HTML file set as the default target. The external HTML file contains the HTML code that references the Google Maps API. You placed the external HTML file in the same folder as the root of the prototype output. When the prototype is run, the inline frame widget runs the external HTML file that requests location data from the browser. After the user allows the browser to share this data, the Google Maps location is shown in the inline frame widget.

Leveraging social media logos – Facebook, Twitter, and Pinterest

Allowing clients to interact with familiar social media properties can help to make your prototypes feel more realistic. This recipe will show you how to open external Facebook, Twitter, and Pinterest pages using social media logos.

Getting ready

For this recipe, you will need the social media logos and the URL links you would like to reference. Logos can be found from the following links:

- Facebook: `www.facebook.com/brandpermissions/logos.php`
- Twitter: `https://twitter.com/logo`
- Pinterest: `http://business.pinterest.com/logos-and-marketing-guidelines/`

How to do it...

For this recipe, you will create three image widgets linked to Facebook, Twitter, and Pinterest.

1. Start Axure and under **Create New** select **RP File**.

If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

2. While holding down the mouse button, drag the **Image** widget and place it at the coordinates (0,50) on the wireframe.
3. Double-click on the **Image** widget on the wireframe and click on the file with the Facebook icon.

Do not autosize the image.

4. With the **Image** widget selected, perform the following steps:
    1. In the **Widget Interactions and Notes** pane, click in the **Image Name** text field and type `FacebookLogo`.
    2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on the **OnClick** interaction.
    3. In the **Case Editor (OnClick)** pop up, in **Case description**, change the case description to `FacebookLogoClicked`.
    4. In **Click to add actions**, click on **Open Link**.
    5. In **Organize actions**, you will see the interaction description change to **Open http:// in Current Window**.
    6. In **Configure actions**, click on the radio button next to **Link to an external URL or file** and type `https://www.facebook.com/PacktPub` in the **Hyperlink** field.
    7. You will see the interaction description under **Organize Actions** change to **Open https://www.facebook.com/PacktPub in Parent Window**.
    8. Click on **OK**.
5. While holding down the mouse button, drag the **Image** widget and place it at the coordinates (100,50) on the wireframe.
6. Double-click on the **Image** widget on the wireframe and click on the file with the Twitter icon.

Do not autosize the image.

7. With the **Image** widget selected, perform the following steps:
    1. In the **Widget Interactions and Notes** pane, click in the **Label** text field and type `TwitterLogo`.
    2. In the **Widget Interactions and Notes** pane, click on the **Interactions** tab and double-click on the **OnClick** interaction.
    3. In the **Case Editor (OnClick)** pop up, in **Case Description**, change the case description to `TwitterLogoClicked`.
    4. Under **Click to add actions**, click on **Open Link in New Window/Tab**.
    5. Under **Organize actions**, you will see the interaction description change to **Open Link in New Window/Tab**.
    6. Under **Configure actions**, click on the radio button next to **Link to an external URL or file** and type `https://twitter.com/packtpub` in the **Hyperlink** field.
    7. You will see the interaction description under **Organize actions** changes to **Open https://twitter.com/packtpub in Parent Window**.
    8. Click on **OK**.
8. While holding down the mouse button, drag the **Image** widget and place at the coordinates (200,50) on the wireframe.
9. Double-click on the **Image** widget on the wireframe and click on the file with the Pinterest icon.

Do not autosize the image.

10. With the **Image** widget selected, perform the following steps:
    1. In the **Widget Properties and Notes** pane, click in the **Label** text field and type `PinterestLogo`.
    2. In the **Widget Properties and Notes** pane, click on the **Interactions** tab and double-click on the **OnClick** interaction.
    3. In the **Case Editor (OnClick)** pop up, in **Case description**, change the case description to `PinterestLogoClicked`.
    4. In **Click to add actions**, click on **Open Link in New Window/Tab**.
    5. In **Organize actions** you will see the interaction description change to **Open Link in New Window/Tab**.
    6. In **Configure actions**, click on the radio button next to **Link to an external URL or file** and type `http://pinterest.com/search/boards/?q=Axure+RP+Prototyping+Cookbook` in the **Hyperlink** field.

7. You will see the interaction description under **Organize actions** change to **Open http://pinterest.com/search/boards/?q=Axure+RP+Prototyping+Cookbook in Parent Window**.
8. Click on **OK**.

11. You can now choose to preview or save a copy of the prototype. To preview the prototype, click on the **Preview** button in the toolbar. To save a copy of the prototype, click on the **Publish** button in the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and click on **Publish**, then click on **Generate HTML Files...**.

How it works...

The key to this recipe is using the image widget with the `OnClick` interaction to open a link in a new window/tab. You created an image widget for each of the social media logos and created an `OnClick` interaction to open a link in a new window/tab.

There's more...

For the **Case Editor** window' s **Click to add actions** you could have selected from the actions **Open Link in Popup Window**. **Open link in Parent Window** and **Open Link in Current Window** do not work properly when using the Axure-generated **Sitemap** and **Page Notes** pane of the prototype.

Another approach would be to use a button, badge, or social plugin. To use a social plugin, you would use an inline frame widget and reference the specific code for the social plugin. Buttons, badges, or social plugins can be found at the following links:

- Facebook: `http://www.facebook.com/badges`
- Twitter: `https://twitter.com/about/resources/buttons`
- Pinterest: `http://business.pinterest.com/pin-it-button/`

Adding app store badges – Apple iTunes and Google Play

Adding direct links to app stores is a neat way to make your prototypes appear more realistic. This recipe will show you how to add working Apple iTunes and Google Play badges to your prototypes.

Getting ready

For this recipe, you will need an HTML editor and will need to generate the code for the Apple iTunes and Google Play badges. The guidelines for linking to iTunes can be found at `http://www.apple.com/itunes/link/` and the **Link Maker** tool can be found at `http://linkmaker.itunes.apple.com/us/`.

The guidelines for linking and creating badges can be found at the following links:

- Apple iTunes: `http://www.apple.com/itunes/link/`
- Apple Link Maker tool can be found at `http://linkmaker.itunes.apple.com/us/`
- Google Play: `http://developer.android.com/distribute/googleplay/promote/linking.html`
- Google Play badge generator tool can be found at `http://developer.android.com/distribute/googleplay/promote/badges.html`

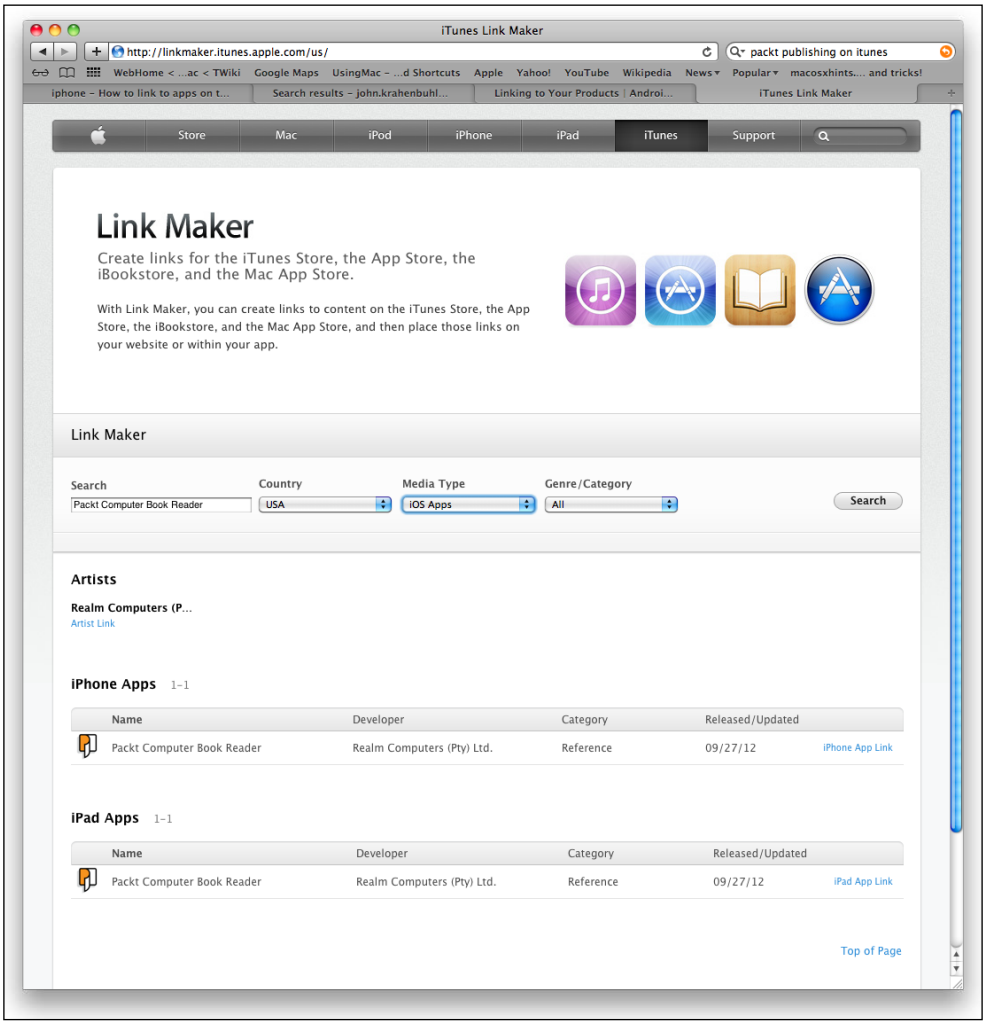
How to do it...

Perform the following steps:

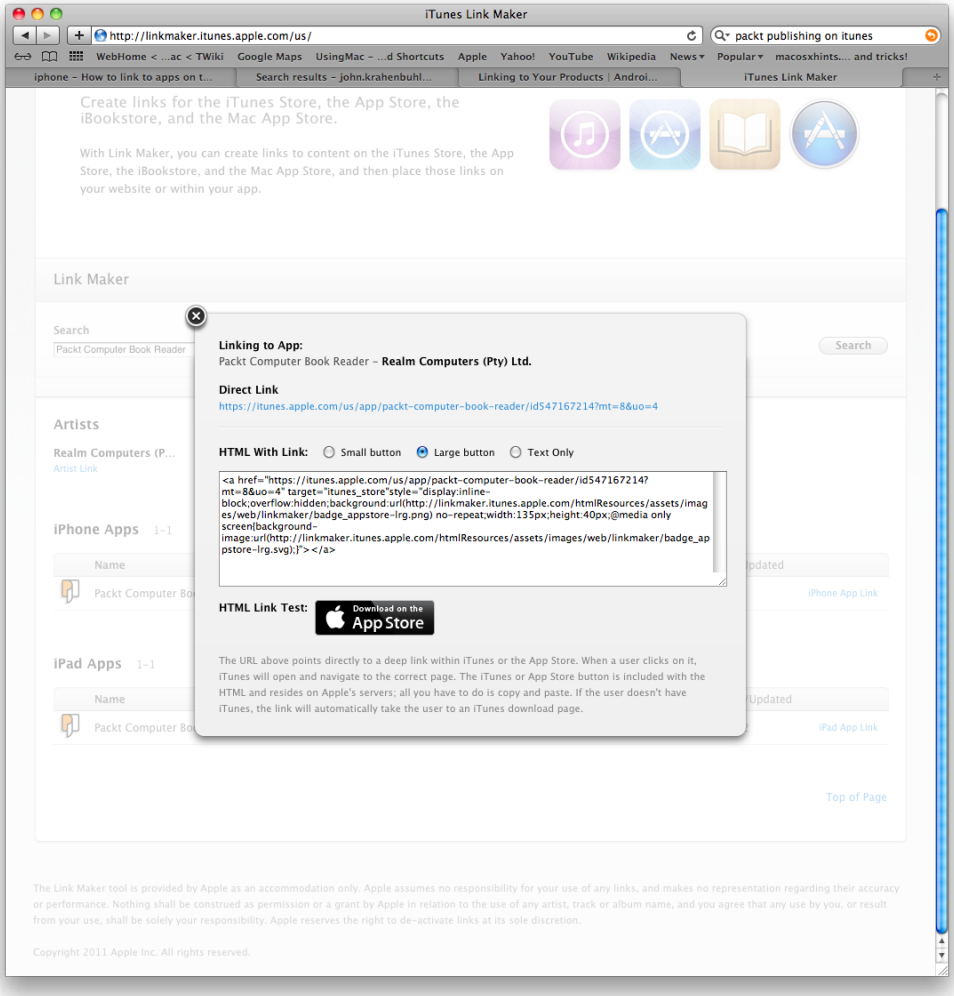
1. Create and save an empty HTML document called `itunes_badge.html` with the following content:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>


  </body>
</html>
```

2. Point your web browser to the Apple Link Maker tool.
3. Enter your search term in the **Search** field, select your **Country**, **Media Type**, and **Genre/Category**, and left-click on **Search**.
4. Select the appropriate result and click on the link.



5. In the pop-up window, next to **HTML with Link:**, click on the radio button that corresponds with either **Small button** or **Large button**.
6. Copy the HTML provided by the Link Maker tool.



7. Insert the HTML from step 1 between the `<body>` and `</body>` tags in the HTML document you created in step 1 and save the HTML document.
8. Start Axure and under **Create New** select **RP File**.

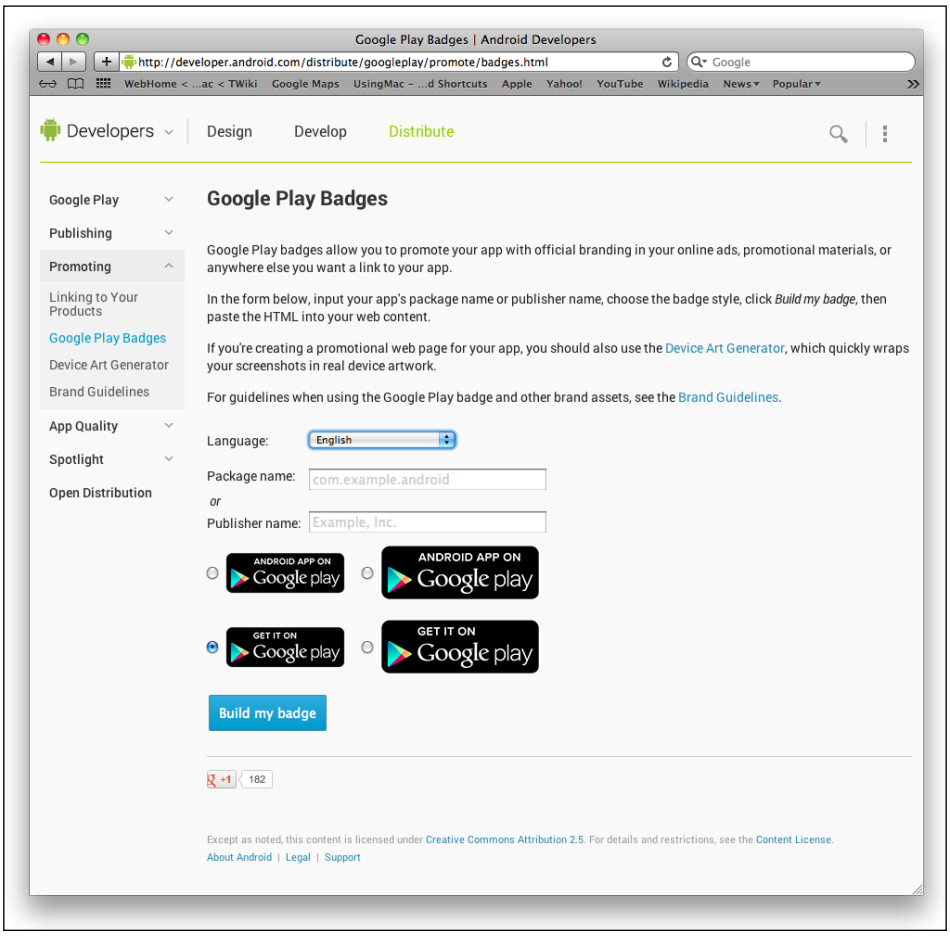
If you already have Axure open, click on **File** in the main menu and then click on **New** in the drop-down menu to create a new RP document.

9. While holding down the mouse button, drag the **Inline Frame** widget and place it at the coordinates (0,0) on the wireframe.
10. Right-click on the **Inline Frame** widget and click on **Toggle Border**.
11. Right-click on the **Inline Frame** widget, click on **Scrollbars**, and click on **Never Show Scrollbars**.
12. Double-click on the **inline frame** on the wireframe.
13. In the **Link Properties** pop up, click on the radio button next to **Link to an external URL or file**.
14. Left-click in the **Hyperlink** field and type **itunes_badge.html**.
15. Click on **OK**.
16. Create and save an empty HTML document called `google_play_badge.html` with the following content:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>

  </body>
</html>
```

17. Point your web browser to the Google Play badge generator tool.

18. Select your language and enter the publisher name for your app.



19. Click the radio button next to the badge you would like to generate.
20. Click on **Build my badge**.
21. Copy and paste the generated HTML from step 20 between the `<body>` and `</body>` tags of the HTML document you created in step 16 and save the HTML document.
22. While holding down the left mouse button, drag the **Inline Frame** widget and place at coordinates (250,0) on the wireframe.
23. Right-click on the **Inline Frame** and click on **Toggle Border**.
24. Right-click on the **Inline Frame** widget, mouse over **Scrollbars**, and click on **Never Show Scrollbars**.

25. Double-click on **Inline Frame** in the wireframe.
26. In the **Link Properties** pop up, left-click on the radio button next to **Link to an external URL or file**.
27. Left-click in the **Hyperlink** field and type `google_play_badge.html`.
28. Click on **OK**.
29. To save a copy of the prototype, click on the **Publish** button on the toolbar and click on **Generate HTML Files...**. You can also generate the prototype by going to the main menu and clicking on **Publish** and then clicking on **Generate HTML Files...**.
30. Note the directory where the prototype is saved.
31. Copy `itunes_badge.html` and `google_play_badge.html` to the same directory as your prototype.
32. In your prototype directory, open `start.html` in a browser to view the prototype.

How it works...

The key to this recipe is using the inline frame widget with an external HTML file set as the default target. The external HTML file contains the HTML code for the badge between the `<body>` and `</body>` tags. You place the external HTML file in the same folder as the root of the prototype output. When the prototype is run, the inline frame widget runs the external HTML file and generates the badge. Clicking on the badge opens the referenced link in a new window.

There's more...

Google also provides a device art generator that will take screenshots of your app and skin them in a device. This is great for helping others visualize your app on their device. The Device Art Generator can be found at `http://developer.android.com/distribute/promote/device-art.html`. For the Device Art Generator to work properly, you will also need to have Google Chrome installed. Google Chrome can be downloaded at `https://www.google.com/intl/en/chrome/browser/`.