Chapter 19: Integration with ArcGIS.com

ArcGIS.com is a Web site for working with maps and other types of geographic information. Visit the site to create maps; find and use maps, applications, and tools; and share your maps and applications with others. Within the site, you will find applications for building and sharing maps. You will also find useful basemaps, data, applications, and tools that you can view and use, plus communities you can join.   For custom web mapping application developers the really exciting news is that you can integrate ArcGIS.com content into your custom developed applications using the ArcGIS Server API for JavaScript.

See maps and apps from people and organizations worldwide

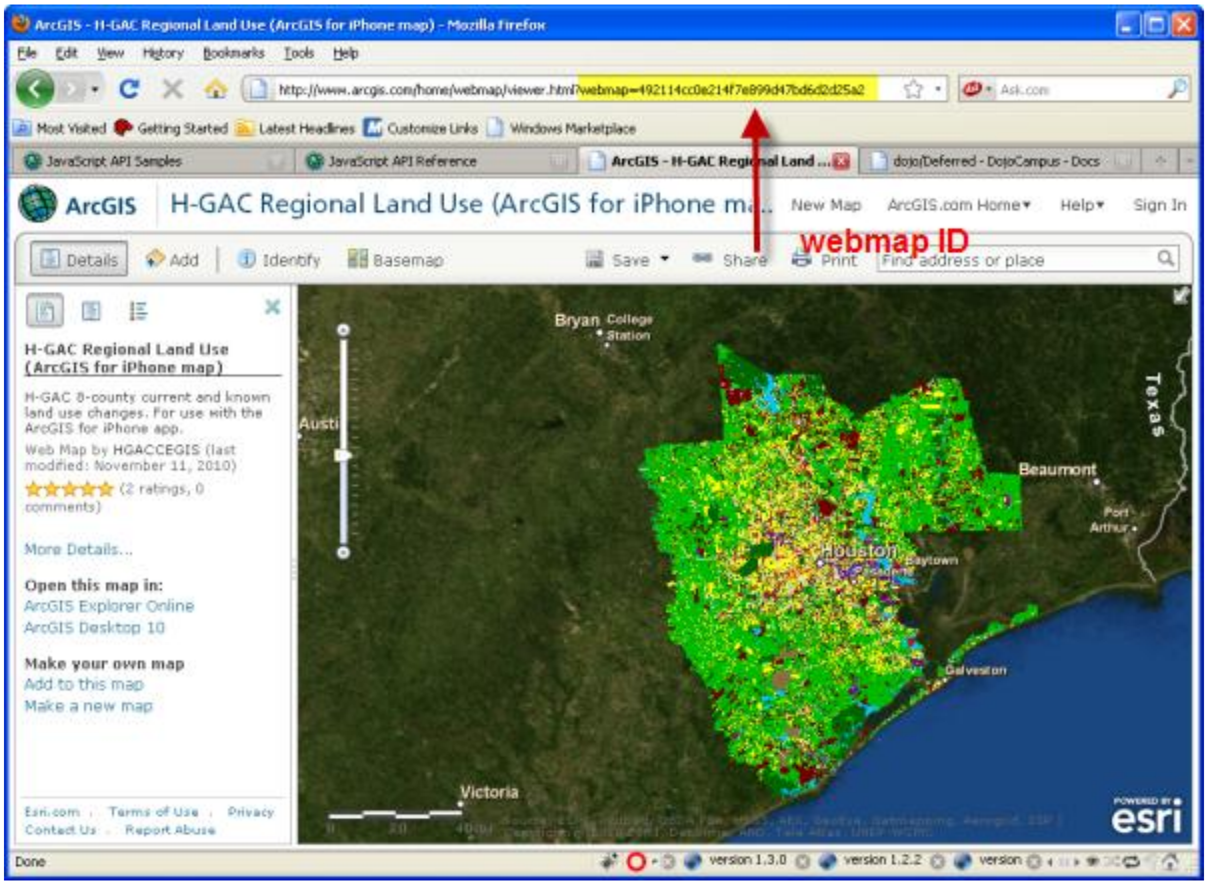
Create a map you can view on the web, desktop and mobile devices.

ArcGIS is an online system for using geographic information everywhere.

The ArcGIS Server API for JavaScript includes a couple utility methods for working with maps from ArcGIS.com. Both methods are found on the esri.arcgis.utils resource. The createMap() method is used to create a map from an ArcGIS.com item.

The WebMap ID

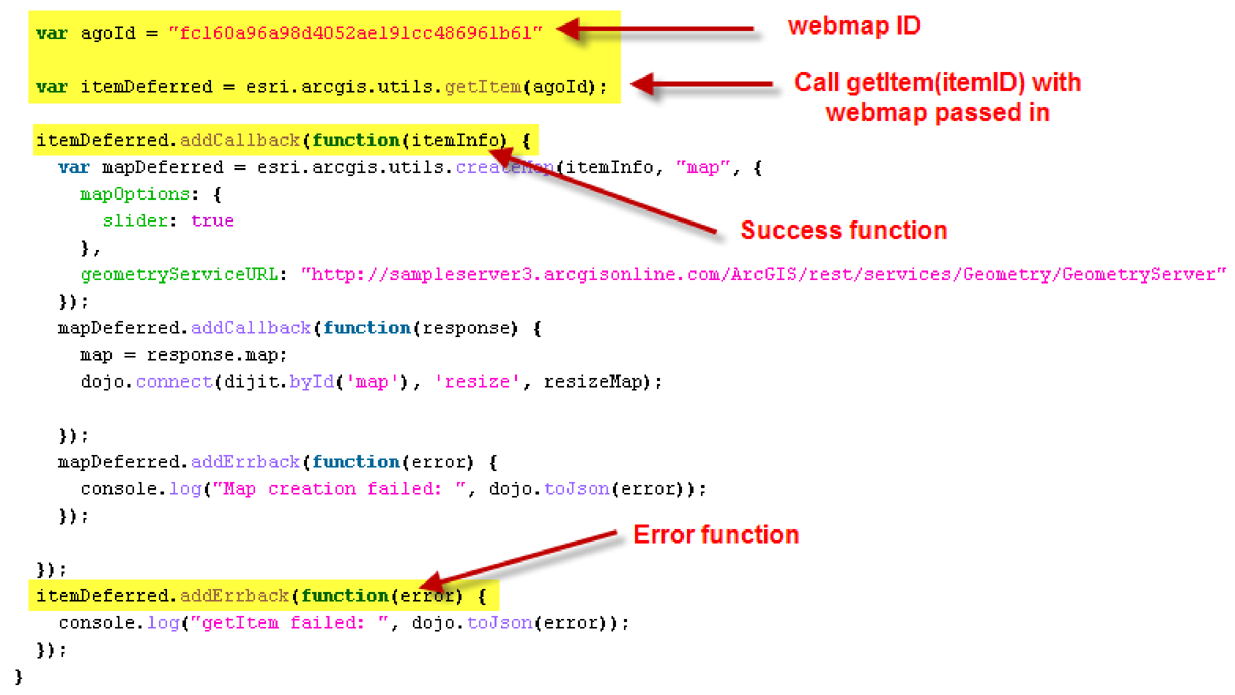
Each map in the ArcGIS.com gallery has a unique ID. This unique ID, called 'webmap' will be important when you begin creating custom applications that integrate maps from ArcGIS.com. To get the 'webmap' ID for a map that you'd like to add to your JavaScript API application simply click a map that you've found which has been shared in ArcGIS.com. The address bar will contain the 'webmap' ID for the map. You'll want to make note of this ID.


ArcGIS - H-GAC Regional Land Use (ArcGIS for iPhone map) - Mozilla Firefox
http://www.arcgis.com/home/webmap/viewer.html?webmap=492114cc0e214f7e899d47bd6d2d25a2
webmap ID
H-GAC Regional Land Use (ArcGIS for iPhone ma...
Details
Add
Identify
Basemap
Save
Share
Print
Find address or place
H-GAC Regional Land Use (ArcGIS for iPhone map)
H-GAC 8-county current and known land use changes. For use with the ArcGIS for iPhone app.
Web Map by HGACCEGIS (last modified: November 11, 2010)
More Details...
Open this map in:
ArcGIS Explorer Online
ArcGIS Desktop 10
Make your own map
Add to this map
Make a new map
Bryan
College Station
Austin
Texas
Beaumont
Port Arthur
Houston
Baytown
Galveston
Victoria
esri

Once you have obtained the 'webmap' ID for the ArcGIS.com map that you'd like to integrate into your custom JavaScript API application you'll need to call the getItem() method, passing in the 'webmap' ID. The getItem() method returns a dojo.Deferred object. Dojo.Deferred is an object built specifically for tasks that may not complete immediately. It allows you define success and failure callback functions that will execute when the task does complete. In this case, a successful completion will pass in an 'itemInfo' object to the success function. This 'itemInfo' object will be used to create the map from ArcGIS.com inside your custom application.

Code Example

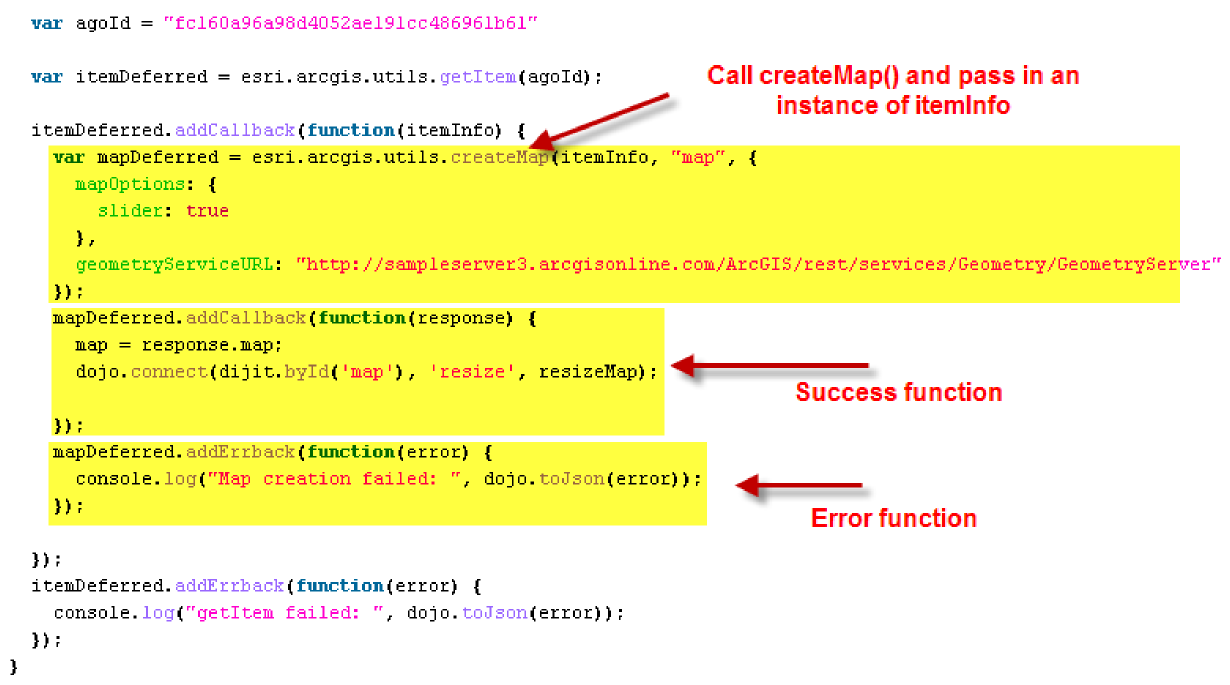
We'll cover this entire function in two separate examples. For now we'll examine the use of the getItem() method along with setting up callback functions for success or failure. In the first line of code we create a variable called 'agoId' and assign it the 'webmap' ID that we'd like to use. Next we call getItem(), passing in the 'agoId' variable containing our 'webmap' ID. This creates a new dojo.Deferred object which we assign to a variable called 'itemDeferred'. Using this object we can then create success and error callback functions. The success function, called 'addCallback' is passed an 'itemInfo' object that we'll use to create our map. We'll cover the actual creation of the map next. In the event of some type of error condition, the 'addErrback' function would be called. Now let's see how the map is created.



```
  var agoId = "fc160a96a98d4052ae191cc486961b61"

  var itemDeferred = esri.arcgis.utils.getItem(agoId);

  itemDeferred.addCallback(function(itemInfo) {
    var mapDeferred = esri.arcgis.utils.createMap(itemInfo, "map", {
      mapOptions: {
        slider: true
      },
      geometryServiceURL: "http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/Geometry/GeometryServer"
    });
    mapDeferred.addCallback(function(response) {
      map = response.map;
      dojo.connect(dijit.byId('map'), 'resize', resizeMap);

    });
    mapDeferred.addErrback(function(error) {
      console.log("Map creation failed: ", dojo.toJson(error));
    });

  });
  itemDeferred.addErrback(function(error) {
    console.log("getItem failed: ", dojo.toJson(error));
  });
}
```

The createMap() method is used to actually create the map from ArcGIS.com. This method takes an instance of ‘itemInfo’ which is returned from a successful call to getItem() or you can simply provide the ‘webmap’ ID. As with any map that you create with the ArcGIS Server API for JavaScript you also need to provide a reference to a <div> container that will hold the map and any optional map options that you’d like to provide. Just as with the getItem(), createMap() also returns a dojo.Deferred object that you can use to assign success and error callback functions.

In the second half of this code example we start by calling the createMap() function and passing in an instance of 'itemInfo' along with the container <div> to hold the map and map options for a slider and geometry service. The result of createMap() is a new dojo.Deferred object which we assign to the variable 'mapDeferred'. With this object we can define success and error callback functions. The success function accepts a 'response' object which contains a 'map' property that we use to retrieve the actual map. The error function runs when a error occurs that would prevent the creation of the map.



```
  var agoId = "fc160a96a98d4052ae191cc486961b61"

  var itemDeferred = esri.arcgis.utils.getItem(agoId);

  itemDeferred.addCallback(function(itemInfo) {
    var mapDeferred = esri.arcgis.utils.createMap(itemInfo, "map", {
      mapOptions: {
        slider: true
      },
      geometryServiceURL: "http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/Geometry/GeometryServer"
    });
    mapDeferred.addCallback(function(response) {
      map = response.map;
      dojo.connect(dijit.byId('map'), 'resize', resizeMap);

    });
    mapDeferred.addErrback(function(error) {
      console.log("Map creation failed: ", dojo.toJson(error));
    });

  });
  itemDeferred.addErrback(function(error) {
    console.log("getItem failed: ", dojo.toJson(error));
  });
}
```