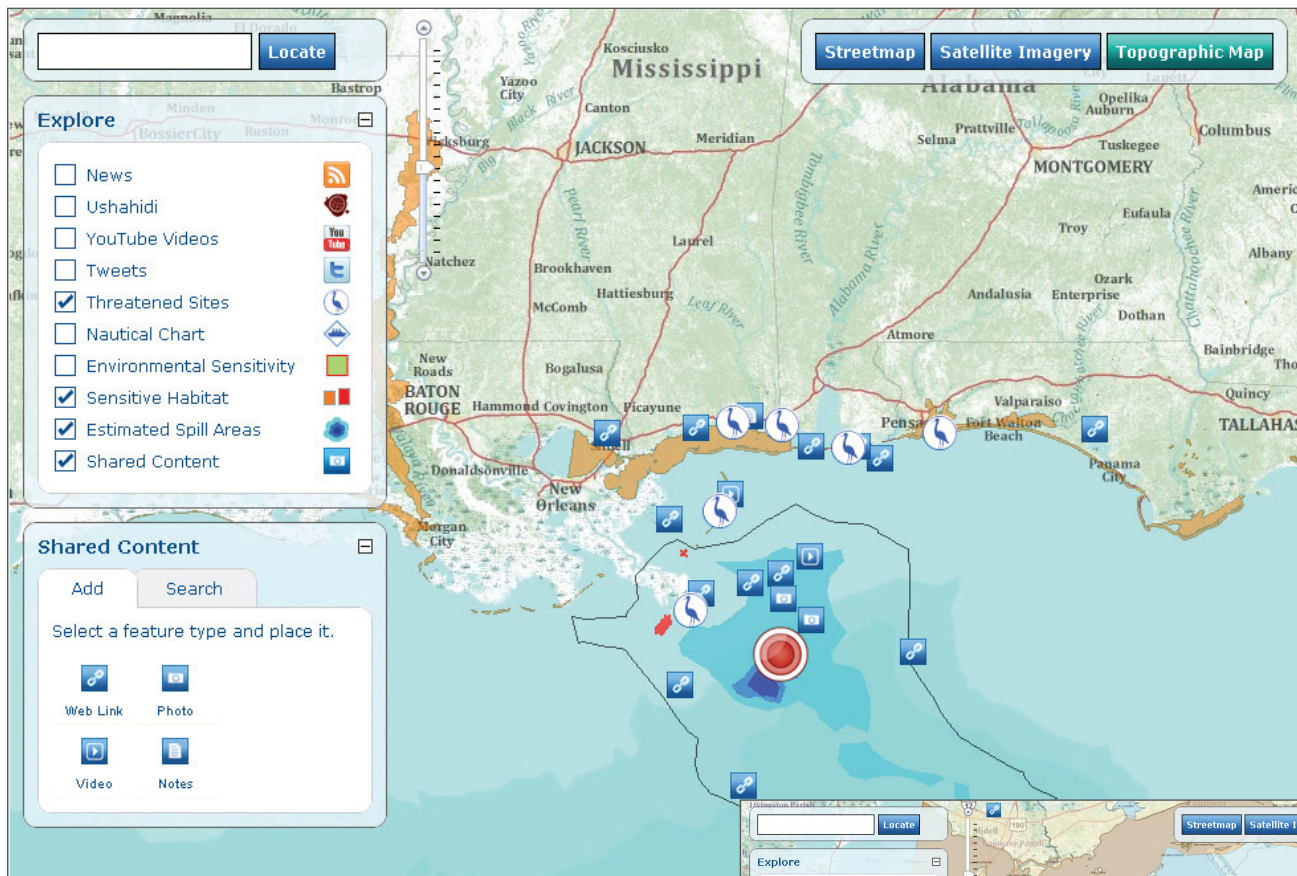


Getting in Touch with Volunteered Geographic Information

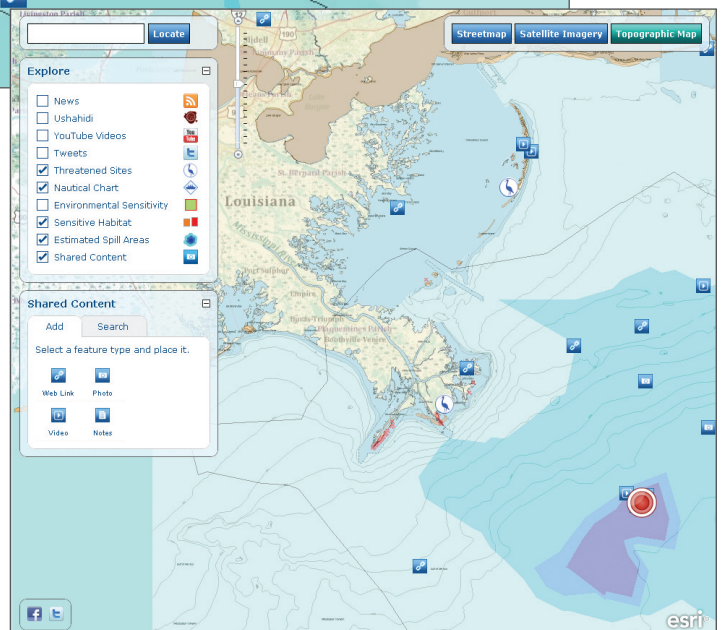
Use a JavaScript API live sample to build a Web editing application

By Bronwyn Agrios and Keith Mann, ESRI



ESRI's Gulf of Mexico Oil Spill map demonstrates volunteered geographic information (VGI) in action. VGI is a term applied to geographic information that has been voluntarily gathered and entered by individuals. This site focuses on the massive oil spill that began on April 20, 2010, as the result of an explosion and fire on board a semi-submersible drilling rig.

VGI Web sites facilitate the creation, assembly, and dissemination of user-generated content. On the basemaps provided, users can define their own content based on firsthand knowledge of local geography. The Oil Spill map application is powered by ArcGIS Server 10 beta hosted in the Amazon Web Service infrastructure. It incorporates VGI-supplied information such as links to online photos, Web sites, GeoRSS feeds, tweets, and YouTube videos that are combined with authoritative basemap data that includes streetmap coverage, satellite imagery, topographic maps, navigational charts, and data on environmentally sensitive sites. Visitors can place new information on the map by choosing a feature icon, clicking on the map, and linking to the content to be added. Learn more by visiting the site at <http://www.esri.com/services/disaster-response/gulf-oil-spill-2010/index.html>.



ESRI's Gulf of Mexico Oil Spill map demonstrates VGI in action.

VGI is a specific example of the community-based information collection model that features content developed by end users. Amazon and Trip Advisor are familiar examples of sites that incorporate user community comments. These comments provide information from real consumers with direct knowledge of the subject and assist potential purchasers in making decisions. You are probably also already familiar with sites that promote geocollaboration and community mapping. Open Street Map and WikiMapia are examples of this type of site. These and other similar geocollaboration sites provide access to informal sources of data and local knowledge about the geography of a place.

All sites that depend on user-generated content must motivate people to act voluntarily. Contributors need to understand how the content they collect supports the site's overall goal. Also, sites that let end users enter information must be managed by an administrator to ensure that the content posted is credible, relevant, and not offensive in content or language and to avoid copyright infringement issues. When end users have been successfully motivated and are contributing to the map, administrators filter the map's geometry to avoid clutter that could make the map illegible. For example, in places that have poorly detailed map coverage or out-of-date or expressive mapping, the rapid entry of information by the user community may result in maps that have overlapping shapes. Site administration is discussed more fully later in this article.

Requirements for Building a VGI Site Using ArcGIS 10

Creating a VGI site requires not only assembling people and software but also designing a workflow, data schema, geodatabase, and map template. A discussion of these requirements follows.

People

The process of designing, constructing, implementing, and administering the site and implementing Web editing will require one or more GIS professionals with knowledge of ArcSDE, geodatabase design, and the use of ArcMap for Web map design and familiarity with publishing services using ArcGIS Server. One or more persons will also be needed to act as site administrators to maintain content credibility. Optionally, the role of the administrator can be limited to setting up a security policy for ensuring that the data is only accessed and/or edited by a select group of people. However, the most important factor to the success of the site is a user community willing to volunteer information.

Required Software Components

The Web application for collecting data should use ArcGIS 10 and one of the ArcGIS Web Mapping APIs that has been configured so all components for editing geographic data are enabled. The following components, configured as described here, are required when setting up a VGI site:

- ArcSDE geodatabase containing features set up for handling Web edits
- An ArcMap document that references one or more feature classes from an ArcSDE geodatabase

- A feature template that defines the properties required for creating a new feature
- The map document and its associated template, published as a map service with feature access
- A Web application with UI controls for making edits to geometry and/or attributes

Designing the Workflow of the ArcSDE Feature Class Schema

Start by preparing the ArcSDE feature classes that will participate in the editing workflows. Keep in mind that the fields and their names (or aliases, if provided) will be visible to the end user of the editing application. Once the feature class has been added to ArcMap as a layer, table properties, such as alias name and visibility, can be tuned further. Please note that the order of the fields can only be defined in the feature class table, so they should be in the order you want the end user to see them in the Web application. Some other things to consider at this stage of data preparation include

- Creating a separate geodatabase feature class to support Web editing if content also exists in the authoritative geodatabase
- How you want the attribute data represented to users who may not be familiar with GIS?
- Defining subtypes and domains (whenever possible) to simplify the attribute editing experience
- Deciding if data can be registered as versioned or edited in a nonversioned environment (Feature classes that participate in nonsimple data types, such as topology or network data, must be registered as versioned.)
- Configuring the feature class to support attachments in the geodatabase. (A feature class that has been configured to use attachments must be added to and published with the map document.)

Preparing the ArcSDE Geodatabase

When preparing the ArcSDE geodatabase, remember that all data that will be added to the map document for editing must come from a single ArcSDE geodatabase and must be registered with that geodatabase. Examine the database connection file to determine whether access to feature classes will be controlled using operating system authentication or database authentication.

Defining Layer and Table Properties in ArcMap

The feature symbology and layer names defined in the map document will be exposed to the Web application end user. Take great care to choose symbols that will be meaningful to non-GIS users and create aliases for all layers.

- When designing the symbols that will be used when editing features, temporarily add the basemap layers that will also be used with the Web application so you can see how features will appear in the legend and on the map when the Web application is in use. In ArcMap, you can further modify how editable layers will appear in the Web application to end users.
- Create aliases for all table fields exposed in the Web editing

Continued on page 52

Getting in Touch with Volunteered Geographic Information

Continued from page 51

interface either in the geodatabase or in ArcMap by changing the attribute table for that feature class. You can hide fields by changing their visibility in ArcMap. Do not hide the Shape field—doing this will prevent the geometry for that layer from displaying in the Web application. Fields can also be made read-only. The order of the fields is defined in the feature class schema—not in ArcMap by changing the layer properties.

- When publishing the map document that will be the source for the editing service, include only layers that will be edited in the Web application. Basemap layers and operational layers that will not be edited in the Web editing application should be published to separate services and mashed up with the editing service in the Web application.

Creating a Feature Template

A feature template packages the settings that have been defined in ArcMap and makes them available to the Web service that will be used for editing. Feature templates define the feature editing experience as well as the information required to create a new feature whether editing in the desktop or the Web. This information includes the layer where the new feature will be stored, the attributes a new feature will be created with, and the default tool used to create that feature. New feature templates are created, and existing ones are edited in ArcMap. Feature templates are saved with the map document used to create them.

When creating a feature template, you will notice that ArcMap layers that are symbolized with either unique value or categorical renderers will have each symbol category exposed as a separate layer in the feature template (and subsequently in the Web editing user interface). When the feature class is also subtyped, each category in the feature template will also maintain the default attribute values as defined for that subtype. If you are not using subtypes, set the default field value for each feature template where appropriate. This makes the attribute editing experience intuitive for end users.

Publishing the Map and Enabling Feature Access on the Service

Finally, the map document is published to ArcGIS Server with both the mapping and feature access capabilities turned on. Both the mapping capability and the Feature Access capability create URLs for the services generated. While a map service serves an image of features, a feature service allows you to serve features over the Internet. Editing on the Web works through a new type of service available with ArcGIS 10: the feature service. In addition to serving features, the feature service also uses the symbology defined in the map document to render the features. Feature service operations include editing and querying.

Defining the User Experience

The end-user experience is directly affected by the design of the feature class, map document, and feature template. The out-of-the-box Web Mapping API components for editing include the Template Picker, Attribute Inspector, and Attachment Editor.

- **The Template Picker** displays a gallery of templates from one or more feature layers. For each template, the symbol and label defined in the map document is displayed. Note that the application developer can overwrite these symbols in the code.
- **The Attribute Inspector** displays the attributes of selected features

from one or more feature layers. If the layer is editable, the feature attributes will be editable. The Attribute Inspector widget honors the domains and subtypes that were defined by the feature service.

- **The Attachment Editor** supports viewing attachments for feature layers that have attachments enabled. If feature layers are from a feature service, the attachment editor will include the ability to create, view, and delete attachments. The application developer/configurer can also determine which API editing components to include in the Web editing application to tailor the editing experience to the end user and type of content being collected.

Administering Ensures the Quality of User-Generated Content

A VGI Web site can become unusable if it is not moderated by an administrator. An administrator eliminates offensive language, inaccurate data, content not related to the site's purpose, and content that could constitute copyright infringement. A best practice for VGI sites is the creation of a simple application that allows non-GIS users to monitor the volunteered content and modify it as needed. Security is another administrative function. Consider implementing ArcGIS Server service security to restrict access to the editable feature service. Remember that the security level is defined at the service level, not feature by feature.

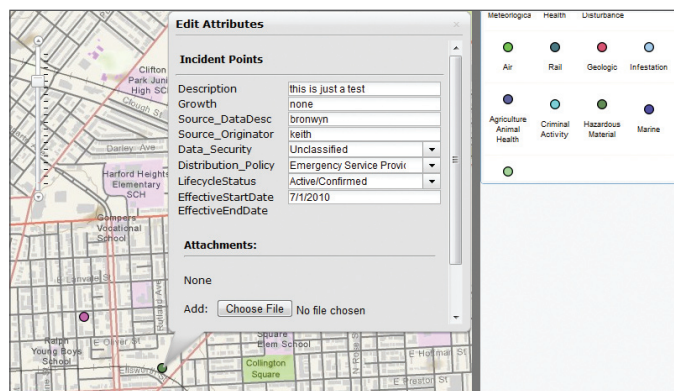
Do It Yourself: Use an Existing VGI Sample to Create Your Own VGI Site

Follow these steps to use a sample Web application that contains feature editing functionality as a template for creating your own VGI site. This hands-on exercise does not require that you have ArcGIS Server 10 but it does require

- A Web browser and access to the Internet
- A Web server running on your machine (Microsoft IIS is used in this example.)
- A text editor (Microsoft Notepad is used in this example.)

View the Live Sample of a Basic VGI App

1. Go to the ArcGIS JavaScript API Resource Center at <http://help.arcgis.com/en/webapi/javascript/arcgis/>.
2. Click the Samples tab and, under JavaScript API Samples, expand the Editing book.
3. Click the first sample, Default Editor.
4. Click the View live sample link in the Description area and explore



This interactive sample is used to demonstrate basic Web editing functionality. Developers can use the sample code to add Web editing to custom Web applications.

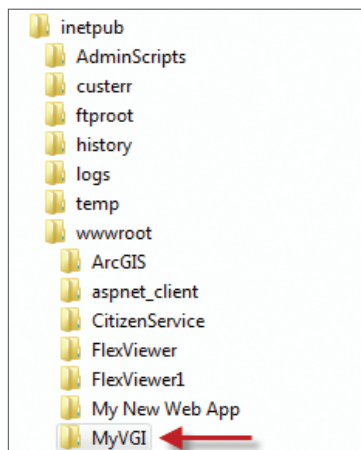
how the sample works.

5. Leave the live sample open.

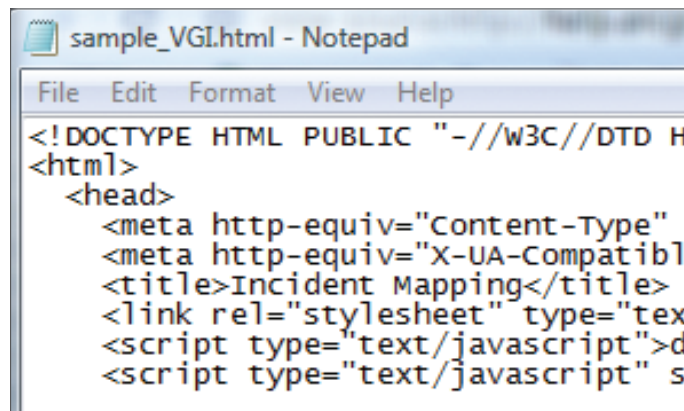
Copy the Application Source Code

The Default Editor widget lets you create and edit features. It includes a Template Picker, an Attribute Inspector, and an Attachment Editor. In these next steps, you'll steal the code for the entire sample so that you can incorporate those functional components into a VGI site hosted on your machine.

1. First, create a folder called MyVGI under the root of your Web server (i.e., C:\inetpub\wwwroot\MyVGI).
2. Next, in the browser containing the live sample for the Default Editor, right-click below the Template Picker (located to the right of the map) and choose View Source (or View Page Source, depending on the Web browser used). Select all page source text, right-click, and click Copy.
3. Now, open Microsoft Notepad and paste the contents into a new Notepad file. Save the file to the MyVGI folder and name it sample_VGI.html. Change the Save as Type to All Files so that the file does not try to save as a (.txt) text file.
4. Finally, navigate to your MyVGI folder and double-click the sample_VGI.html file that you just created to open the application again—this time you will open the file from the code on your machine. Editing functionality will not work properly at this point, but you will fix that in the next step.



Create a folder named MyVGI in your wwwroot folder.

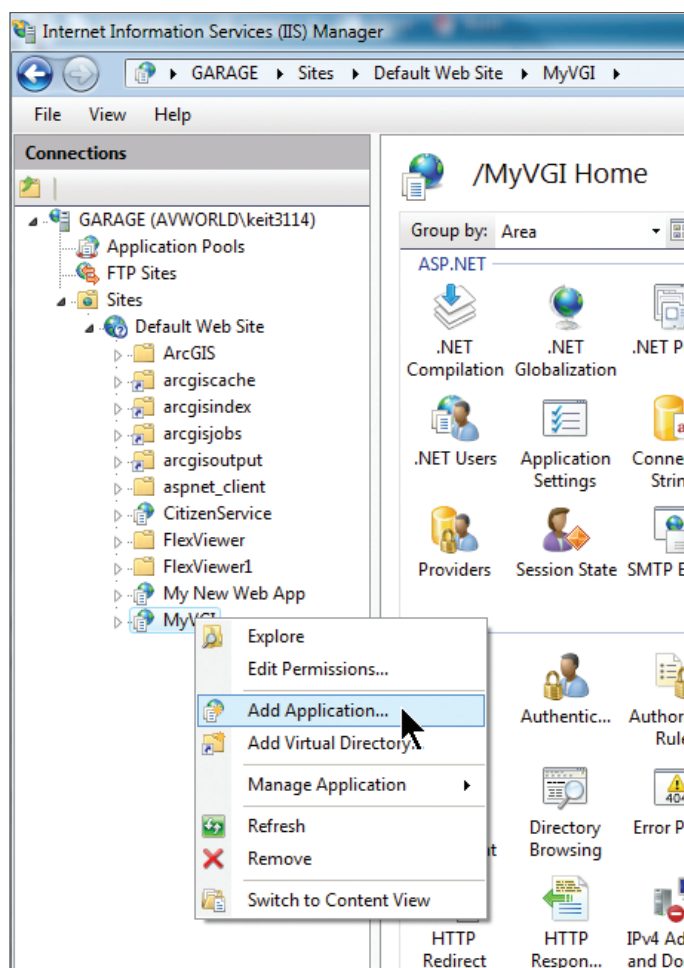


Copy the sample application source text and paste it into a new Notepad document.

Make MyVGI Folder a Web Application Using IIS

In this step, you will use Microsoft IIS to create a Web application from the MyVGI folder.

1. Open your Control Panel, double-click Administrative Tools, then double-click Internet Information Services (IIS) Manager.
2. Although the IIS Manager may look slightly different for XP, Vista, and Windows 7 operating systems, the task is the same for all three—expand the IIS tree structure to expose your MyVGI folder that will be located in the Default Web Site folder, then either add or create the application.
3. For Vista, in the Default Web Site folder, right-click the MyVGI folder and click Add Application. In the Add Application dialog box, name the alias MyVGI and, if necessary, define the Physical path by clicking the ellipses button and navigating to your MyVGI folder. Select it and click OK, then click OK again to close the Add Application dialog box.



In the Windows Vista IIS Manager, click Add Application to add your MyVGI folder.

4. For XP, in the Default Web Site folder, select MyVGI and choose properties. From the Directory tab under Application Settings, click Create. Click OK.
5. Close the IIS Manager.

Continued on page 54

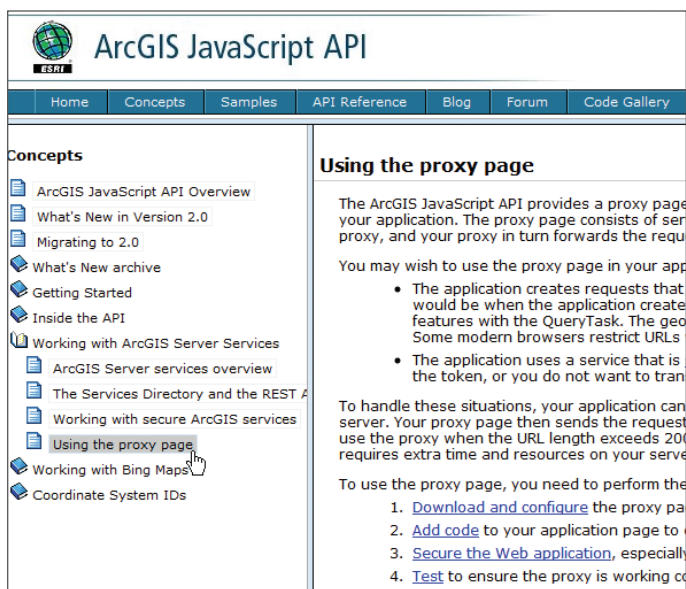
Getting in Touch with Volunteered Geographic Information

Continued from page 53

Modify the Application Code

Now that your application is recognized by your Web server, you'll make a few minor changes to the application code so that it will work on your local machine.

1. First, in your Web browser, go back to the ArcGIS API for JavaScript Resource Center and click the Concepts tab. Expand the Working with ArcGIS Server Services book and click Using the proxy page.
2. Click the Download and configure link (in the body of the Topic text) and click proxypage_net.zip to download the zipped file. Save it to your MyVGI folder.



The proxy page consists of server-side code that runs on your Web server.

3. Extract the proxypage_net.zip to the MyVGI folder.
4. In Notepad, open your sample_VGI.html file and find the section that begins,

```
function init() {
```

Change the path in the line

```
esri.config.defaults.io.proxyUrl = "/arcgisserver/apjs/javascript/proxy/proxy.ashx";
```

to

```
esri.config.defaults.io.proxyUrl =  
"http://<your_machine_name>/MyVGI/proxy.ashx";
```

Save and close sample_VGI.html.

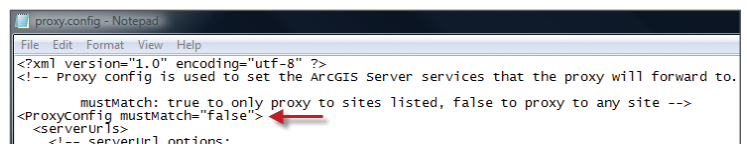
Note: Be sure to replace <your_machine_name> with the name of your machine.

5. In Windows Explorer, navigate to your MyVGI folder, right-click the proxy.config file, and open it in Notepad.
6. Find the following line (near the top) and change "true" in the line <ProxyConfig mustMatch="true"> to "false."
<ProxyConfig mustMatch="false">
Save and close proxy.config.

7. Open a Web browser and go to your MyVGI Web app:

http://<your machine name>/MyVGI/sample_VGI.html

Remember to replace <your machine name> with the name of your machine.



Change ProxyConfig mustMatch to False.

The appearance and functionality of the application will be exactly the same as the sample in the JavaScript API Resource Center, except that now it is hosted on your machine.

Replace the Feature Service with a Different Feature Service

Now that you're hosting the MyVGI app, you will replace the feature service it uses with a different feature service. This action represents what you might do if you were to use your own feature service hosted on your ArcGIS Server. In this case, you will borrow a different feature service from ESRI.

In your Web browser, go to

http://sampleserver3.arcgisonline.com/ArcGIS/rest/services

This page is the ArcGIS Services Directory for sampleserver3 at ESRI. The Services Directory is part of any ArcGIS Server implementation and is a way to explore and access the REST endpoints of ArcGIS Server services. Keep in mind that the services on sampleserver3 are for demo purposes only and should not be used in your production applications.

1. Click BloomfieldHillsMichigan to expose the services in that folder. These services were created using the process described in the first part of this article.
2. Click BloomfieldHillsMichigan/LandusePlanning (FeatureServer) to expose a list of the layers of this service. The service contains three layers: Land Use Points of Interest, Land Use Linear Features, and Land Use Planning Areas.
3. Click Land Use Points of Interest (0) and copy the URL from the address bar of the browser to your system clipboard.
4. If necessary, open your sample_VGI.html file in Notepad again and find the line that begins as follows (near the middle of the file):

ArcGIS Services Directory

[Home](#) > [BloomfieldHillsMichigan](#)

Folder: BloomfieldHillsMichigan

Current Version: 10.0

View Footprints In: [Google Earth](#)

Services:

- [BloomfieldHillsMichigan/LandusePlanning](#) (MapServer)
- [BloomfieldHillsMichigan/LandusePlanning](#) (FeatureServer) ←
- [BloomfieldHillsMichigan/Parcels](#) (MapServer)

Supported Interfaces: [REST](#) [SOAP](#) [Sitemap](#) [Geo Sitemap](#)

Replace the original feature service with a different feature service.

```
var operationsPointLayer = new esri.layers.  
FeatureLayer
```

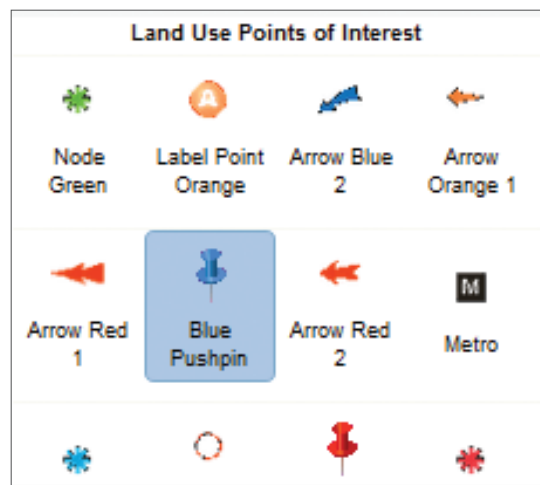
Replace the URL for this feature layer with the URL you have copied for Land Use Points of Interest (0).

- Repeat this process for the remaining two feature layers, copying in the URLs for Land Use Linear Features (1) and Land Use Planning Areas (2) and replacing the URLs for operationsLineLayer and operationsPolygonLayer, respectively. Save the file.
- In your Web browser, open your sample_VGI app again (e.g., http://<your machine name>/MyVGI/sample_VGI.html) or, if you still have it open, refresh the page by pressing F5.

Now the map has been cleared of the VGI content that accumulates in the ArcGIS API for JavaScript sample app, and a new set of feature symbols has replaced the previous set in the editing template.

Take a few moments to add new content to the map.

Keep in mind that while this app is hosted on your machine, the feature are stored in a geodatabase hosted by ESRI. To make this



Your sample_VGI app now uses the feature editing template for the BloomfieldHillsMichigan feature layers.

app truly your own, you need to create your own feature layers as described in the first part of this article.

Conclusion

This exercise stepped you through the process of setting up one of the editing samples on your local machine and updating the references to the feature service to change the content that will be edited by the application.

```
var operationsPointLayer = new esri.layers.FeatureLayer  
("http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/BloomfieldHillsMichigan/LandusePlanning/FeatureServer/0", {  
  mode: esri.layers.FeatureLayer.MODE_ONDEMAND,  
  outFields: ["*"]  
});  
var operationsLineLayer = new esri.layers.FeatureLayer  
("http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/BloomfieldHillsMichigan/LandusePlanning/FeatureServer/1", {  
  mode: esri.layers.FeatureLayer.MODE_ONDEMAND,  
  outFields: ["*"]  
});  
var operationsPolygonLayer = new esri.layers.FeatureLayer  
("http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/BloomfieldHillsMichigan/LandusePlanning/FeatureServer/2", {  
  mode: esri.layers.FeatureLayer.MODE_ONDEMAND,  
  outFields: ["*"]  
});
```

New URLs for the feature layers