[PD-03-019] GIS APIs

Abstract

GIS APIs are collections of library modules that resemble various functionalities of GIS software through programming. GIS APIs evolved from desktop GIS. GIS APIs, as a distributed solution, are interoperable, scalable, light-weight, user-friendly, and versatile to a wide range of GIS users. This entry provides an overview of common GIS APIs, their functionalities as well as other related APIs. The general procedure to develop customized GIS applications is briefly discussed and demonstrated in a case study.

Keywords: Application Programming Interface, distributed GIS, Geographic Information Service, Internet GIS, RESTful, web GIS, web services

Author & citation

Chow, E. and Yuan, Y. (2019). GIS APIs. The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2019 Edition), John P. Wilson (Ed.).
DOI: 10.22224/gistbok/2019.2.15.

Explanation

1. Definitions
2. Introduction
3. GIS APIs
4. Other APIs for Data Retrieval and Analysis
5. Case Study
6. Closing Remarks

1. Definitions

GIS APIs: GIS Application Programming Interfaces are collections of library modules that resemble various functionalities of GIS software through programming

GIService: Geographic Information Service

2. Introduction

As GIS evolved from desktop packages to distributed solution(s), API has become a popular option for developing customized GIS applications (Chow, 2008). Over a web platform, GIS APIs provide a communication protocol to handle client requests and return the processed response back from a server that provides GIS functionalities and/or datasets as a service (i.e., GIService) (Yue, Ramachandran, & Baumann, 2015). The advantage of communicating GIS requests and responses through an API is to enable interoperability across platforms, systems, and devices. It also empowers a light-client architecture so that a client (e.g., a web browser) can process, analyze, and visualize distributed geographic information with

no prior GIS installation or even plug-in(s). GIS APIs are also scalable in that developers can import specific GIS APIs, with or without parallelization, needed for a certain GIS application (Rey, Anselin, Pahle, Kang, & Stephens, 2013). This is consistent with the development of service-oriented architecture in software design, where services are provided as application components to minimize the dependency between different components when designing a large software product (Bouguettaya, Sheng, & Daniel, 2014).

Given the origin of GIS APIs, it is not surprising that desktop GIS developers, such as Esri, QGIS, and GRASS GIS, built their GIS APIs to incorporate many functionalities of their desktop counterparts. However, GIS APIs from other developers, such as Google, remain limited in terms of tools and GIS functionalities. Moreover, some GIS APIs are designed to be developed into a web application (i.e., “online” web mapping) or a customized desktop solution. For example, the ArcPy package developed based on Python is often used for customizing desktop GIS tools. Because GIS APIs evolved from desktop GIS, it is perhaps useful to discuss existing GIS APIs in terms of their functionalities for the “capture, storage, transformation, retrieval, analysis, and display of spatial data” (Clarke, 2003). A common function of GIS APIs is the retrieval and display of spatial data with a dynamic map navigation interface. To many users, being able to view, explore, and overlay map layers is sufficient for basic applications. Another common function of GIS APIs is geocoding (i.e., address matching)- the act of assigning a x- and y-coordinate based on a postal address or a place name. The geocoding function (and sometimes reverse-geocoding) is useful because a significant portion of “Big Data” contains some spatial references, such as spatial coordinates, place names, or geographic metadata (Leszczynski & Crampton, 2016). A few APIs specialize in parsing and geocoding large and heterogeneous data (e.g., Texas A&M Geocoding API, Geoname API, GeoPy).

In the digital era, there has been an overwhelming inundation of geographic data made available through the advancement of remote sensing technologies, crowdsourcing, social media, GPS-enabled smart sensors, Internet of Things (IoT), etc. This creates a large demand for transforming the captured geospatial data into interoperable formats for dissemination and integration. Many data APIs enrich GIS APIs by providing supplementary gateways to query geographic data, including government records (e.g., Census Bureau APIs), social media (e.g., Twitter APIs), private sources (e.g., Google Data APIs), open sources (e.g., Overpass API), etc. The availability of GIS data from multiple sources also presents an opportunity to analyze their geographic distribution, spatial relationship, temporal trend, and semantic pattern. Some APIs (e.g. scikit-learn, TensorFlow API) specialize in analytical functions, like spatial, statistical, mathematical, or machine-learning functions.

3. GIS APIs

Similar to desktop GIS software packages, some GIS APIs are developed by commercial vendors whereas some originated from Free and Open Source Software (FOSS) projects (Table 1). Although commercial GIS APIs may require paid subscriptions, these products often have free versions with reduced functionalities or usage limits. It is important to note that many GIS or data APIs are available in different programming languages. For instance, Twitter allows API access from various popular languages, such as Python, R, or Java. Altogether, these GIS APIs provide the building blocks to develop distributed GIS

applications. Besides the GIS APIs listed in Table 1 and a select few that are briefly discussed in the following sections, other GIS APIs include Shapely, Geopandas, Carto, Turf, here, cisco location APIs, Estimote, etc.

Table 1. A list of selected GIS APIs

Name	Licensing	Programming Language(s)	Usage	URL
ESRI ArcGIS APIs	Pay-per-use	Python, JavaScript	Desktop/terminal & web development	ArcGIS API for Python ArcGIS API for JavaScript
QGIS APIs	FOSS	C++, Python	Desktop/terminal & web development	QGIS
GRASS GIS APIs	FOSS	C, Python	Desktop/terminal & web development	GRASS GIS APIs
Google Maps APIs	Pay-per-use	JavaScript, REST	Web development	Google Maps Platform Documentation
Microsoft Bing Maps APIs	Pay-per-use	JavaScript, TypeScript, C#	Web development	Microsoft APIs
OpenLayers API	FOSS	JavaScript	Web development	OpenLayers API
Leaflet API	FOSS	JavaScript	Web development	Leaflet
Mapbox web services APIs	Pay-per-use	cURL, CLI, JavaScript, Python, Java, REST, Swift	Web development	Mapbox APIs
Geonames API	FOSS	Java	Geocoding & web development	Geonames API
TAMU Geoservices Geocoder	FOSS	REST	Geocoding & web development	Texas A&M Geoservices Geocoding APIs
PySAL	FOSS	Python	Spatial analysis	Python Spatial Analysis Library
GISolve Open Service API	FOSS	REST	Web development & data integration	GISolve Open Service API
GeoPandas	FOSS	Python	Data integration & processing	GeoPandas

3.1 Esri ArcGIS APIs

Esri has developed multiple GIS APIs to complement its flagship desktop GIS software, “ArcGIS,” and its core server for data, service, and application management, “ArcGIS Server.” The two main types of API libraries developed by Esri are Python-based APIs and JavaScript-based APIs.

Esri provides several Python-based APIs for customization. For example, ArcPy is a Python package that provides an effective way to access the data types, spatial analysis functions, and geoprocessing tools in ArcMap through various Python functions, classes, and modules. These ArcPy Python functions correspond to the tools in ArcToolbox, hence, manipulating vector, raster, and tabular data in ArcPy is straight-forward for an experienced ArcMap user with basic Python knowledge. Another Python-based library is “ArcGIS API for Python”,

which provides a high-level encapsulation of ArcGIS functionalities (e.g., managing, creating, and analyzing GIS data) for developing web GIS applications. It is implemented using the online and web GIS platform provided by ArcGIS Online and ArcGIS Enterprise, and is also compatible with the new platform ArcGIS Pro. Compared to desktop-based ArcPy, “ArcGIS API for Python” provides a simple solution for integrating ArcGIS into web application development.

ArcGIS API for JavaScript is a commercial JavaScript library for building interactive web mapping applications. The newest version 4.x integrates both 2D and 3D map displays and provides seamless support for various map services published to an ArcGIS server, such as tiled map services, feature services and raster services. It also supports other data formats, such as graphics layers, CSV layers, GeoRSS Layers, and WMS Layers. This API implements many powerful analytical tools and widgets, such as geocoding, search, directions, measurement, track current location, etc. Since it is built upon Dojo core and Dijit library, familiarity with the Dojo toolkit is recommended before implementing ArcGIS JavaScript API.

3.2 QGIS APIs

QGIS, a continuously growing FOSS GIS project, also offers a Python API to implement its versatile GIS engine for building customized applications. Similar to Esri ArcPy, QGIS APIs offer a versatile package of many GIS functionalities and can process various GIS data models (e.g., vector, raster, database) and formats (e.g., Shapefile, JSON, GeoTiff, etc.). Although QGIS APIs are not designed to develop web applications, there are plug-ins (e.g., qgis2web) that can help QGIS interface with other web GIS APIs like OpenLayers and Leaflet.

3.3 Google Maps APIs

Google Maps APIs are commercial pay-per-use JavaScript libraries for interactive web mapping. It is one of the most beginner-friendly and convenient APIs for creating interactive online maps and appealing front-end visualization. This API library supports multiple data formats, such as Keyhole Markup Language (KML), Graphic feature layers, GeoRSS, etc. It also integrates many unique powerful data layers and tools from Google Maps, such as traffic layer, transit, routing, searching, and geocoding. However, the main drawback of Google Maps APIs is its limited spatial analysis functionality and its limited support for Esri shapefile format, one of the most commonly used file types in the GIS field. In addition to JavaScript APIs, Google Maps APIs also offer an Embed API and a Static API for minimal scripting. Hence, Google Maps APIs are most appropriate for users who develop light-weight web mapping applications with limited spatial analysis needs and minimal coding.

3.4 Microsoft Bing Maps APIs

Microsoft Bing Maps supports various map APIs for users with different needs, including but not limited to: V8 web control for basic map display (supports JavaScript and TypeScript); Windows 10 Universal Windows Platform for displaying maps with 2D, 3D, or Streetside views on the Windows 10 platform (supports C# and XAML); REST services and Spatial data services for geocoding, reverse-geocoding, querying, routing, and other rest functions (supports C# and VB); and Fleet management services for distance matrix computing, travel-time optimization, and asset tracking.

3.5 OpenLayers API

OpenLayers is a high-performance open-source JavaScript API for creating a dynamic map in any web page. It supports multiple data types, such as tiled map layers (e.g., OpenStreetMap, Bing, Mapbox, and Stamen) and vector layers (e.g., GeoJSON, KML, GML, and Mapbox vector tiles). It is also convenient for loading markers and other external tabular data using OpenLayers. The process of creating advanced map visualizations, like animations, is straightforward. Compared to Google Maps APIs, OpenLayers is free and also beginner-friendly for creating a basic map display. However, it has much less support for RESTful services such as geocoding and routing. Although less stable than commercial products, OpenLayers, as an active and powerful FOSS mapping library, is commonly used for research projects in universities and non-profit organizations.

3.6 Leaflet API

Leaflet is an emerging open-source JavaScript library that aims to deliver light-weight interactive web maps on multiple platforms (e.g., mobile). Similar to many map APIs, Leaflet can easily overlay tile layers, vector geometries, and many other GIS data formats, including shapefiles, Geodatabase, GeoJSON, CSV, KML, GeoTIFF, etc. Moreover, the availability of many plug-ins extends the open project with spatial operations. For example, a variety of heatmap plugins can derive and render the kernel density surfaces for a very large dataset (e.g., tweets) as a tile layer that changes dynamically at different zoom levels on the fly. Leaflet also supports other spatial operations, such as geocoding, clustering, buffering, point-in-polygon containment, great circle routing, measuring, etc. It is also noteworthy that there are some Leaflet plugins that can interface with the spatial data files or web services hosted by other GIS software vendors, including ESRI, QGIS, R, etc.

4. Other APIs for Data Retrieval and Analysis

There are other APIs that provide either geographic data or analytical services (e.g., machine learning) that can supplement GIS APIs in the development of advanced GIS applications (Table 2). While some APIs (e.g. OpenStreetMap APIs, Census Bureau APIs), can deliver GIS data, it is noted that some APIs (e.g. Twitter APIs, WhitePages API) can only retrieve data with geographic reference (e.g. latitude/longitude or postal address) that are not necessarily in GIS data format. It requires additional effort to parse and process, such as geocoding, displaying latitude/longitude, joining data with existing GIS data like census units) in order to visualize them in GIS for further analysis. Even though some APIs, such as Scikit learn or TensorFlow, are scientific toolkits, they are included in this discussion to illustrate a broader framework of API integration.

Table 2. A list of selected GIS-related APIs

Name	Primary Function(s) related to GIS APIs	Programming Language(s)	Documentation
Twitter APIs	Data	REST	Twitter APIs
Flickr API	Data	REST	Flickr APIs
OpenStreetMap API	Data	REST	OpenStreetMap API
Census Bureau APIs	Data	REST	Census Bureau APIs
WhitePages API	Data	REST	WhitePages Pro API

Scikit-learn API	Analytics	Python	Scikit-learn API
TensorFlow API	Analytics	Python, JavaScript, C++, JAVA, Swift	TensorFlow API

4.1 Twitter APIs

The standard Twitter APIs include basic functions such as streaming and searching, whereas more advanced functionalities, such as conducting complex queries and metadata enrichments (e.g., to retrieve improved profile information), are provided in the Premium APIs and Enterprise APIs. Most non-commercial applications of Twitter use the free standard APIs to collect data. The featured functionalities of the free APIs include: collecting real-time streaming data, searching tweets within 7 days, filtering and sampling tweets, retrieving basic user profile information, etc. It is important to note the limitation of free standard APIs before implementing them. First, there is a limit on the number of requests made per 15 minutes. Second, the standard APIs can only obtain around 1% of all tweets, which raises questions of data completeness.

4.2 OpenStreetMap API

OpenStreetMap (OSM) adopts the same collaborative model as Wikipedia, creating a free and editable map project that is available to users all over the world. The map data from OSM is crowdsourced from users and considered the most valuable output of this project. OSM provides a REST API to retrieve, edit, and save the raw geodata to/from the back-end database. Users are able to specify the target area to download using a bounding box or more complicated queries. The downloaded data can be saved as an .osm file, which is an XML-based file format that can be converted to Esri shapefiles using open source GIS tools such as QGIS.

4.3 U.S. Census Bureau APIs

As a U.S. federal agency, the Census Bureau disseminates publicly available demographic datasets through their APIs, including TIGERweb, decennial Censuses, the American Community Survey, economic surveys, etc. Many of these APIs provide access to a specific dataset via RESTful web services and return a JSON object for parsing.

4.4 Scikit-learn APIs

Built upon mathematical Python libraries like NumPy, SciPy, and matplotlib, Scikit-learn is a Python package that provides statistical and data mining tools, including generalized linear models, Naïve Bayes classifications, decision trees, ensemble methods, Support Vector Machines, neural networks, etc. While Scikit-learn is designed for general scientific investigation, these algorithms are useful for conducting cluster analysis, feature extraction, geovisualization, pattern recognition, and prediction common in GIScience research.

4.5 TensorFlow API

TensorFlow is a deep learning Python package developed by Google for advanced Neural Network (NN) modeling. This open-source library is built upon Keras, a high-level API specification that uses layers to organize the input data, configure neuron weights, and

setup the activation function to build a convolution NN model. In addition to multi-language support, this API is also flexible in offering a low-level API for experienced developers and also a Lite version for mobile devices. This deep learning API expands the array of analytical tools to be integrated with GIS APIs.

5. Case Study

This case study demonstrates the process of applying one of the most widely adopted GIS APIs, Google Maps JavaScript API, to create a basic web mapping application for displaying the seven wonders of the world.

5.1 Setup

Similar to other commercial GIS APIs, the first step of using Google Maps APIs is to obtain an API key (i.e., an access token), which is a unique identifier that tracks and identifies the usage of APIs calling from a certain application. It is necessary for maintaining sustainable usage of API functions. Most commercial GIS APIs still provide free access for educational or research use up to a predefined usage limit. For example, users of Google Maps JavaScript APIs will receive $200 of credit each month under fair use (Google, 2018).

5.2 Loading Google Maps APIs

Figure 1 shows how to load Google Maps APIs in JavaScript, and there is no installation required. After loading the library/package, developers will be able to call the pre-defined functions, tools, and widgets in their own applications.

```
<script async defer
  src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=
initMap">
</script>
```

Figure 1. Loading Google Maps APIs in JavaScript

5.3 Applying API classes, functions, and tools

In this example, our goal is to create a marker for each of the seven wonders, and when clicking on a marker, an info-window with the description of the wonder will display. There are three main classes used in this case study: map, marker, and infowindow. The specific attributes and methods of these classes can be found in the API documentation (Google, 2018).

First, we created a basic map object as a container of the markers (Figure 2), then we populated an array with the description of the image address for each wonder. Next, using a for loop, we created the marker and infowindow for each wonder (Figure 3). Note that we also registered a “click” event for each marker so that when clicking on the marker, an

infowindow with description will display. Figure 4 shows the final product.

```
var originalMapCenter = new google.maps.LatLng(26.3351, 17.2283)
var map = new google.maps.Map(document.getElementById('map'),
{
      center: originalMapCenter,
      zoom: 2

});
```

Figure 2. Example code – creating a map object.

```
for( let i = 0; i < wonders.length; i++ ) {
      var position = new google.maps.LatLng(wonders[i][0], wonders[i][1]);
      marker = new google.maps.Marker({
               position: position,
               map: map,

      });
      map.setCenter(marker.getPosition());
      google.maps.event.addListener(marker,'click',(function(marker,i){
               return function(){
               infowindow.setContent(wonders[i][2]);
               infowindow.open(map,marker);

      };})(marker,i));

}
```

Figure 3. Example code – creating info windows in a for loop.



Figure 4. An example of a web mapping application created using Google Maps API.

6. Closing Remarks

APIs provide a flexible way for software development and component integration. In particular, GIS APIs provide a visualization and analytical platform to explore the underlying pattern and formulate questions about the relationship between form and process. In light of the distribution of spatial-data based products, spatial data infrastructure, and cloud computing, GIS APIs are an important component that shapes the continual development of future GIS for multi-disciplinary scientific investigation.

References

Chow, T. E. (2008). The potential of maps APIs for Internet GIS Applications. Transactions in GIS, 12(2), 179-191.

Clarke, K. C. (2003). Getting Started with Geographic Information Systems (4th ed.). Upper Saddle River, New Jersey, USA: Prentice Hall.

ESRI. (2018). ArcGIS Developer Program Pricing. Retrieved Dec 7, 2018, from ArcGIS for Developer.

Google. (2018). Pricing for Maps, Routes and Places. Retrieved Dec 7, 2018.

Leszczynski, A., & Crampton, J. (2016). Introduction: Spatial Big Data and everyday life. Big Data & Society, July–December, 1-6.

Rey, S. R., Anselin, L., Pahle, R., Kang, X., & Stephens, P. (2013). Parallel optimal choropleth map classification in PySAL. International Journal of Geographic Information Science, 27(5), 1023-1039.

Yue, P., Ramachandran, R., & Baumann, P. (2015). Editorial: Intelligent GIServices. Earth Science Informatics, 8(3), 461-462.