Simio

Forward Thinking

Simio API Note: Interfacing with GIS Data

Document History
Creation: 15 Feb 2018
Revisions:
14 Mar 2018 – Including extra DLL instructions.
26 Mar 2024 – Include providers for Bing, Google, and ArcGIS

Contents

Overview

GIS (Geographic Information System) data can be an important source of data for Simio models. However, the collection and loading of this data into a simulation can be cumbersome.

This GisAddIn Simio Add-In was constructed to demonstrate how Simio can be extended to work with 3rd party GIS APIs (Application Programming Interfaces) to automatically construct Simio models.

This Simio Add-In interface provides a demonstration of this GIS-to-Simio capability. It can be used to automatically construct simulation paths from routes obtained via a GIS service.

After being launched by Simio, the GisAddIn does this by:

1. Prompting a user for two locations: From and To.
2. Getting GIS information to provide a set of highway directions (a route) using a web-based GIS API.
3. Using GIS-obtained route to construct Simio objects. Specifically, paths between a 'from' and 'to' node.

It is constructed as an Simio "Add-In", which by convention means that it is a DLL that is called by Simio desktop application while in the Simulation Design mode of Simio.

The Add-In is launched from the "Project Home" ribbon by selecting from the "Select Add-In" button, and then choosing GIS Interface.
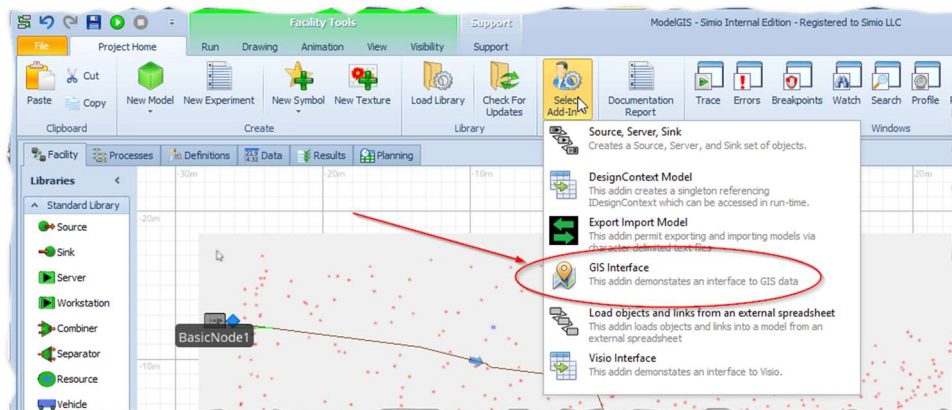
Once launched, the Add-In brings up its UI, which is a WinForms 'test' screen that allows you to:

- Select a provider (Bing, Google, or ArcGIS),
- Enter to enter the "From" and "To" GeoCode, and
- then press a button to Query a route from the GIS API and Apply it to the FacilityView

You can then go to the Facility View and use View > View All to see the entire route.

Here is the result after selecting each provider tab, querying for a route and then applying a route:
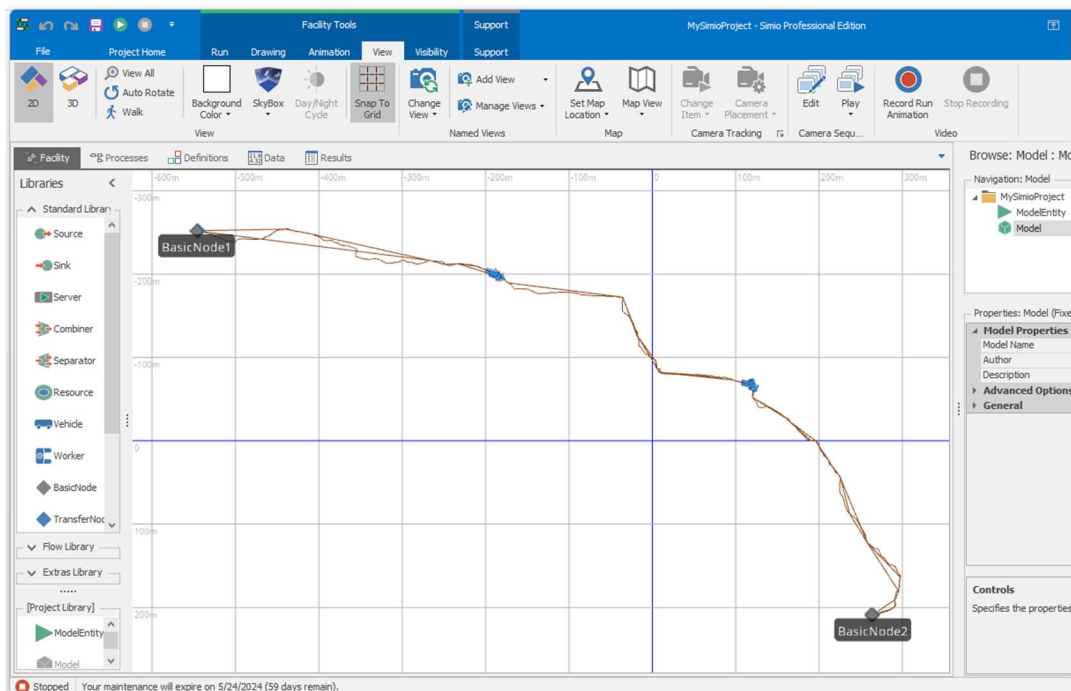
On each provider tab (e.g. Bing Maps) are found boxes to enter a "From" and "To". GIS providers today are very flexible with what you can enter here. For example, you could enter street addresses and a zip code, or just a city with a state (in which case the GIS provider chooses a default location), or even simply a zip code.
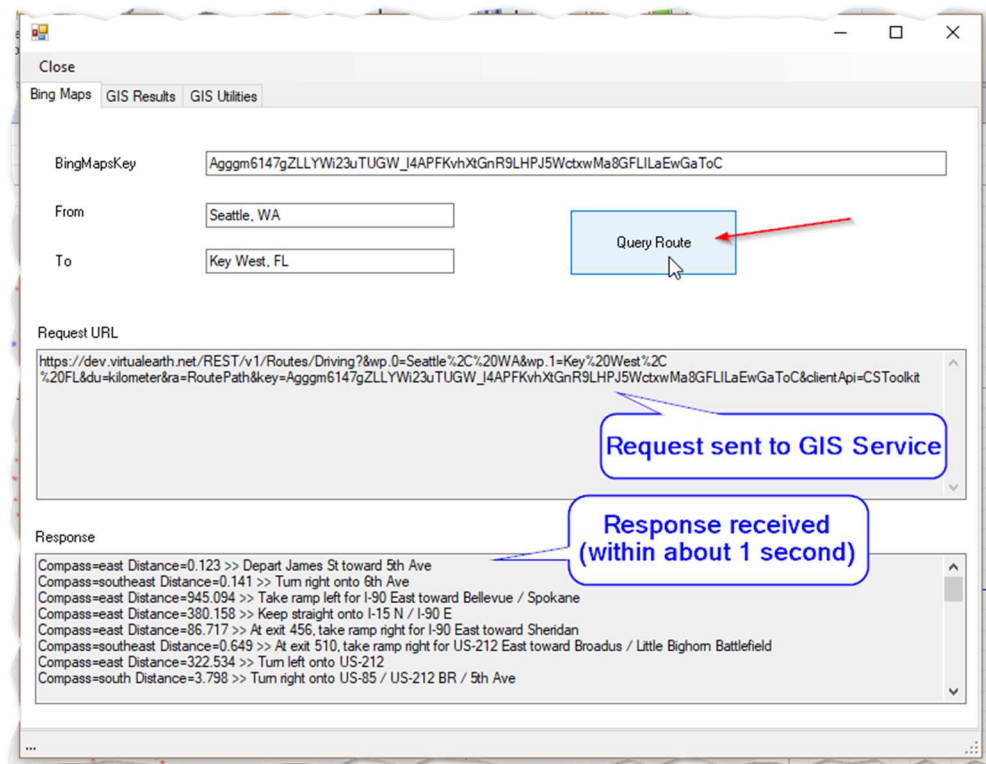
Figure 1 - The GisAddIn User Interface

Once the "From" and "To" are chosen, press the button and the GisAddIn will construct a "Route' request query and send it to the GIS Provider. When you are satisfied with your route, close the form and the Add-In will construct the Simio objects in the proper location on the Simio Facility view.

The Simio objects are constructed comprising two "BasicNode" nodes with a Path between them. These nodes are attached to a Simio Source and Sink so that the model is immediately available to run.

Note that for this Add-In, the dimensions were chosen so that a unit of latitude and longitude each correspond to a meter in Simio's Facility coordinate system. The convention for this demonstration uses a bounding box that is from 20 to 50 North latitude and -60 to -130 longitude, with the lat/lon mid points of -95, 35 being centered at the Simo origin (0,0).

GisAddIn can be invoked repeatedly to add more routes to the model.

Here is a screenshot of a Simio Facility view with three routes added to a Model, and the model running (note the green triangle Entities). Note that the pointillism 'map' is simply an imported symbol showing cities within the contiguous US.
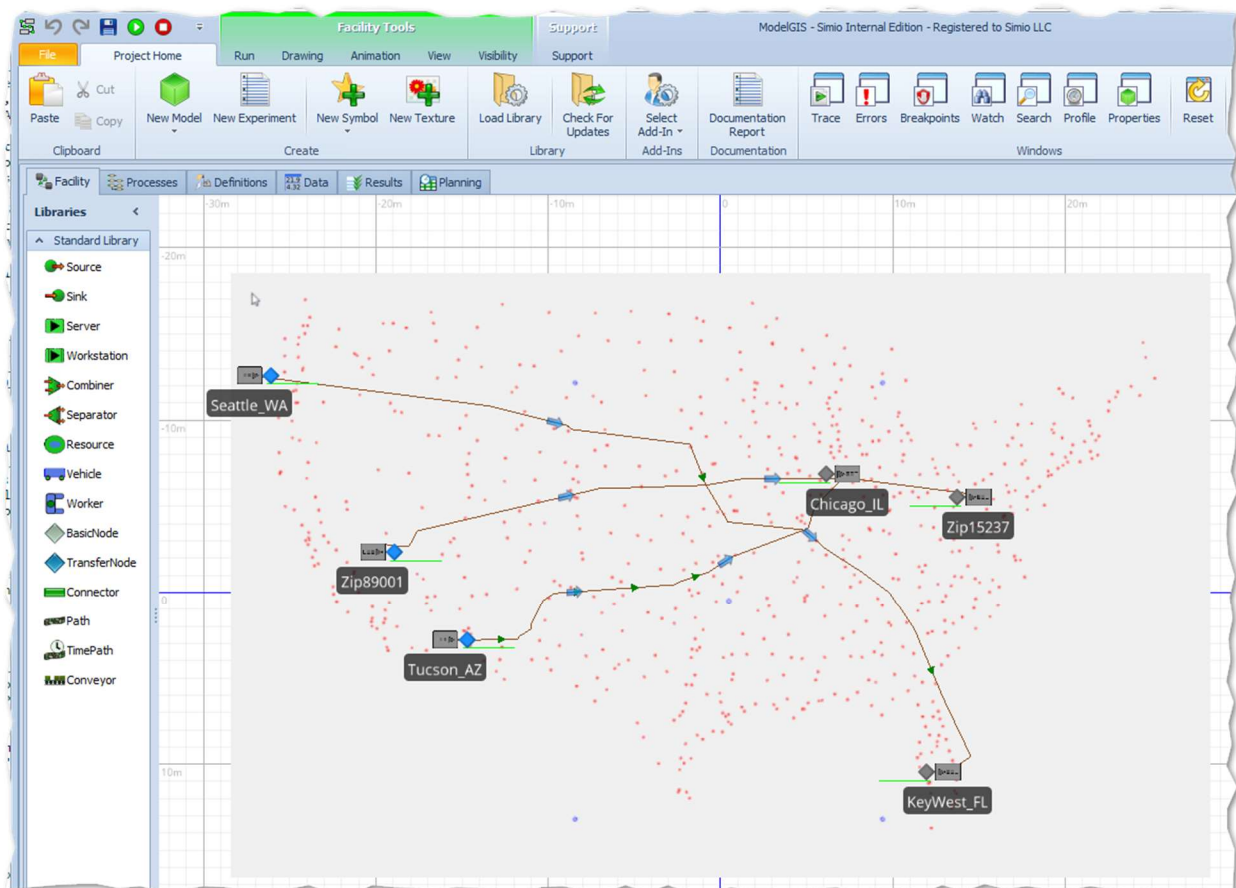
Installing the GIS Add-In

The GIS Add-In is distributed in GitHub as a repository at SimioLLC\GisInterface, which includes all source code, binary, a sample Simio model, and documentation:

When the binaries are placed in a standard Simio DLL folder (such as %username% > Documents > SimioUserExtensions) the Simio desktop will find them and place the Add-In in the toolbar when it runs:
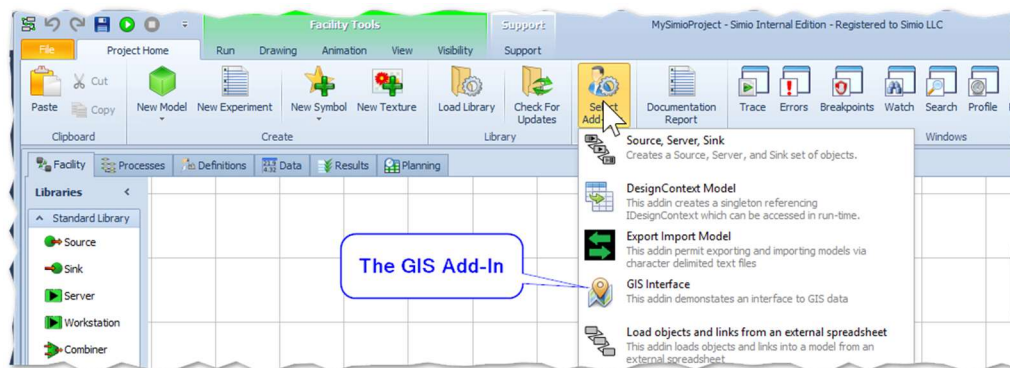
Modifying/Developing the Add-In

If you are interested in examining and/or modifying the Add-In for your purposes (for example, in Visual Studio) you can clone the repository to a local location of your choosing.

You can download a video to help you with this procedure by going to here:

https://github.com/SimioLLC/Videos

And then downloading and viewing GetExampleFromGitHub.mp4
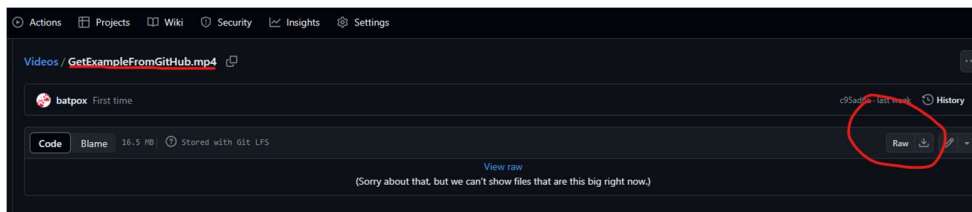
Each provider will have its own procedure for allowing you to access the data. As of this writing (March 2024) each of the three demonstrated providers requires you so sign on as a developer and provides you with a free key for development. You can use the web and/or ChatGPT to assist you.

GIS – Getting the Data from the Web

GIS data used in the GisAddIn is gathered from the web with what is called a REST-compliant service on the internet. The REST (REpresentational State Transfer). What this means is that the calling techniques – regardless of the GIS Service Provider used - will likely be consistent.

How the data is returned will most likely be XML or JSON, but the organization of the data will vary widely depending on the provider. Each provider will likely have their own 'kit' or SDK (Software Development Kit) to provide us with the tools to easily implement the communications. But, it will always end up being a Request to the service, following (within a few seconds) by a Response with our data.

Each Provider will require a "key" to make requests of their system. This is because they must maintain an expensive infrastructure to supply this information, and because they wish accountability of the users of their system. Often this key will be free if your volume of requests is below a given threshold.

For example, Microsoft, via Bing Maps and their Virtual Earth service, a REST-full API (Application Programming Interface). A 'free' key available, so long as you stay below a certain data transaction level.

These are the same services that power most smart-phone, desktop, and web applications, but in developing this Add-In we are going to use the services directly.

In fact, you can copy the contents of the Add-In's "Request URL" textbox (place your cursor in the Add-In's Request URL box) and then:

1. CTRL-A to select all,
2. CTRL-C to cut, and then
3. go to your browser's URL and CTRL-V to paste.

Then hit the ENTER and you'll see the results:

https://dev.virtualearth.n ×

← → C 🔒 Secure | https://dev.virtualearth.net/REST/v1/Routes/Driving/?&wp.0=Seattle%2C%20WA&wp.1=Key%20West%2C%20FL&du=kilometer&ra=RoutePath&key=Agggm6147gZLLYWi23u...   entApi=CSToolkit

📍 Google Chrome isn't your default browser. [Set as default]

The 'request' URL

The response JSON-formatted data.

{"authenticationResultCode":"ValidCredentials","brandLogoUri":"http:\/\/dev.virtualearth.net\/Branding\/logo_power... © 2018 Microsoft and its suppliers. All rights reserved. This API cannot be accessed and the co
may not be used, reproduced or transmitted in any manner without express written permission from Microsoft Corpora...   Total":1,"resources...s.microsoft.com\/search\/local\/us\/rest\
[24.55194,-122.329455,47.714,-80.12884],"id":"v69,h-223455169,i0,a0,cen-
US,dAAAAAAAAA1,y0,s1,m1,o1,t4,wkfC9v0HNR0Bs3AnCF2VewA2~AjOI155R3TQBAADgARGq_z4A0~U2VhdHRsZSwgV0E1~~~~~v11,wa5kMx_ONOECcTUcAN3NUwA2~AjOI154hncgtAADgAZFGAT8A0~S2V5IFdl...   "Kilometer","durationUnit":"Second","rou
{"type":"Point","coordinates":[24.554502,-81.800232]},"actualStart":{"type":"Point","coordinates":[47.603575,-122.329455]},"alternateVias":[],"cost":0,"description":"...   [24.208826,-82.39576,24.901587,-81.1634
FL","point":{"type":"Point","coordinates":[24.554501,-81.800232]},"address":{"adminDistrict":"FL","adminDistrict2":"Monroe","countryRegion":"United States","for...   Key
West","confidence":"High","entityType":"PopulatedPlace","geocodePoints":[{"type":"Point","coordinates":[24.554501,-81.800232],"calculationMethod":"Roof...sageTypes":["Display"]}],"matchCodes":["Ambiguous"]}],"itineraryItems":
[{"compassDirection":"east","details":[{"compassDegrees":67,"endPathIndices":[1],"locationCodes":["114-10561"],"maneuverType":"DepartStart","mode":"Driving","names":["James St"],"roadType":"Arterial","startPathIndices":
[0]}],"exit":"","iconType":"Auto","instruction":{"formattedText":null,"maneuverType":"DepartStart","text":"Depart James St toward 5th Ave"},"maneuverPoint":{"type":"Point","coordinates":
[47.603575,-122.329455]},"sideOfStreet":"Unknown","tollZone":"","towardsRoadName":"5th Ave","transitTerminus":"","travelDistance":0.123,"travelDuration":18,"travelMode":"Driving"},{"compassDirection":"southeast","details":[{"compassDegrees":13
[3],"locationCodes":["114-08569"],"maneuverType":"TurnRight","mode":"Driving","names":["6th Ave"],"roadType":"Arterial","startPathIndices":[1]}],"exit":"","iconType":"Auto","instruction":{"formattedText":null,"maneuverType":"TurnRight","text":"
Ave"},"maneuverPoint":{"type":"Point","coordinates":[47.60415,-122.32806]},"sideOfStreet":"Unknown","tollZone":"","transitTerminus":"","travelDistance":0.141,"travelDuration":20,"travelMode":"Driving"},{"compassDirection":"east","details":
[{"compassDegrees":133,"endPathIndices":[15],"locationCodes":[47.60415,-122.32806]},"maneuverType":"TakeRampLeft","mode":"Driving","roadType":"Ramp","startPathIndices":[3]},{"compassDegrees":72,"endPathIndices":[2794],"locationC
04119","114N04119","114-08270","114-08271","114-04117","114-04116","114N04116","114-04115","114N04115","114-04114","114N04114","114N04113","114N04112","114-04111","114N04111","114-04110","114N04110","114-04109","114N04109","114-04108","114N041
04107","114N04107","114-04106","114-04105","114N04105","114-04104","114N04104","114-04103","114-04102","114-04101","114N04101","114-04100","114-04099","114-04098","114N04098","114-04682","114N04682","114-04681","114N04681","114-
04680","114N04680","114N04679","114-04678","114N04678","114-04677","114N04677","114-04860","114-04859","114-04858","114-04857","114N04857","114-04856","114N04856","114-04855","114N04855","114-04854","114-04
04852","114N04852","114-04851","114-04849","114N04849","114-04848","114N04848","114-04847","114N04847","114-04846","114N04846","114-04845","114N04845","114-04844","114N04844","114-04843","114N04843","114-04842","114-04841","114-04840","114-048
04838","114-04832","114N04832","114-04831","114N04831","114-04830","114-04829","114N04829","114-04828","114N04828","114-04827","114N04827","114-04826","114N04826","114-04825","114-04824","114N04824","114-04823","114-048
04571","114N04573","114-04574","114N04574","114-04575","114N04575","114-04576","114-04579","114-04580","114N04580","114-04581","114N04581","114-04582","114N04582","117-04123","117N04123","117N04122","117-04121","117N04121","117-04120","117N041
04119","117N04119","117-04118","117-04117","117N04117","117N04116","117-04116","117N04115","117N04115","117-06301","117N06301","117-04114","117-04192","117N04192","117-04191","117N04191","117-04190","117-04189","117N04189","117-041
04186","117N04186","117-04185","117N04185","117-04184","117N04184","117-04183","117N04183","117-04182","117N04182","117-04181","117N04181","117-04180","117N04180","117-04179","117N04179","117-04178","117N04178","117-04177","117N04177","117-04176","117N04
04174","117N04174","117-04395","117N04395","117-04394","117N04394","117-04393","117N04393","117-04392","117N04392","117-04391","117N04391","117-04390","117N04390","117-04389","117N04389","117-04388","117N04388","117-04387","117N04387","117-043
04385","117N04385","117-04384","117N04384","117-04383","117N04383","117-04382","117N04382","117-04381","117N04381","117-04380","117N04380","117-04379","117N04379","117-04378","117N04378","117-04377","117N04377","117-04376","117-04375","117N043
04374","117N04374","117-04373","117N04373","117-04372","117N04372","117-04371","117N04371","117-04370","117N04370","117-04369","117N04369","117-04368","117N04368","117-04367","117N04367","117-04366","117N04366","117-04365","117N04365","117-04364","117-04363","117N04
04362","117N04362","117-04361","117N04361","117-04360","117N04360","117-04359","117N04359","117-04358","117-04357","117N04357","117-04356","117N04356","117-04355","117N04355","117-04354","117N04353","117-04
04351","117N04351","117-04350","117N04350","117-04349","117N04349","117-04348","117N04348","117-04347","117N04347","117-04346","117N04346","117-04345"],"maneuverType":"Merge","mode":"Driving","names":["I-90 East"],"roadType":"RampThenHighwayLeft","text":"Take ra
toward Bellevue \/ Spokane"}],"maneuverPoint":{"type":"Point","coordinates":[47.60307,-122.32707]},"sideOfStreet":"Unknown","signs":["Bellevue","Spokane","I-90
East"],"tollZone":"","transitTerminus":"","travelDistance":945.094,"travelDuration":29919,"travelMode":"Driving","warnings":[{"severity":"None","text":"Entering Idaho","warningType":"AdminDivisionChange"},
{"origin":"47.70705,-117.00718","severity":"Minor","text":"Minor Road hazard: Between Idaho\/Washington State Line and ID-3\/Exit 34 - Snow on the road.","to":"47.623311,-116.583781","warningType":"RoadHazard"},
{"origin":"47.630583,-116.604257","severity":"Minor","text":"Minor Road hazard: Between ID-97\/Exit 22 and Latour Creek Rd\/Exit 40 - Snow on the road.","to":"47.54675,-116.331388","warningType":"RoadHazard"},
{"origin":"47.6243,-116.645","severity":"Minor","text":"Incident: Between ID-97\/Exit 22 and Lookout Pass\/Exit 0 - Traffic problem.","to":"47.45608,-115.69539","warningType":"Miscellaneous"},{"origin":"47.546747,-116.331431","severity":"Minor
hazard: Between Exit 39 and I-90-BL\/ID-4\/Exit 62 - Snow on the road.","to":"47.47296,-115.91214","warningType":"RoadHazard"},{"origin":"47.47265,-115.91451","severity":"Minor","text":"Minor Road hazard: Between Frontage Rd\/Exit 61 and Look
Snow on the road.","to":"47.45608,-115.69539","warningType":"RoadHazard"},{"origin":"47.46848,-115.78931","severity":"Minor","text":"Warning: At I-90-BL\/Exit 69 - No through traffic for heavy trucks.","to":"47.46814,-115.77819","warningType":"
{"severity":"None","text":"Entering Montana","warningType":"AdminDivisionChange"},{"origin":"47.39509,-121.51946","severity":"Moderate","text":"Moderate Congestion","to":"47.3951,-121.5191","warningType":"TrafficFlow"}]},{"compassDirection":"e
[{"compassDegrees":83,"endPathIndices":36421,"locationCodes":["117-04400","117N04400","117-04399","117N04399","117-04398","117-04397","117N04397","117-04396","117N04396","117-04301","117-04300","117N04300","117-04299","117N04299","117-04298"

The GIS data that can be received from these providers is not simply routing. Each provider has a large and rapidly growing set of features.
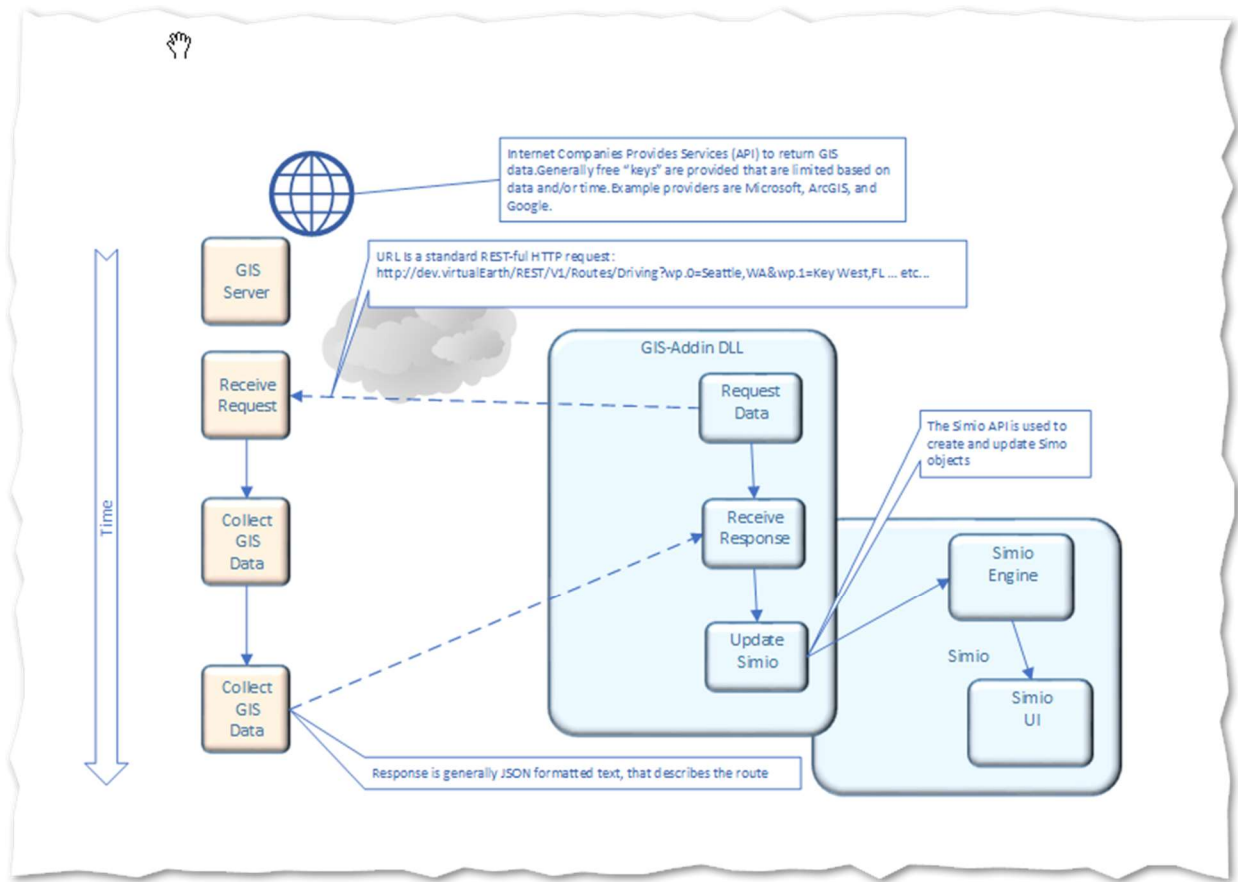
For example, Bing Maps can supply routes for Trucks, where the characteristics of the truck (e.g. weight) are taken into consideration.

Other Responses include distance matrices and even solutions to the Travelling Salesman problem.

Data Provider Examples

This section shows examples of data providers and the Add-In is used to process the data. The first API (Feb 2018) is Bing Maps, with the intent that more will be added as they are encountered.

Most of the GIS APIs work basically in the same fashion, as illustrated by the diagram below:

Simio
F o r w a r d   T h i n k i n g

Internet Companies Provides Services (API) to return GIS data. Generally free "keys" are provided that are limited based on data and/or time. Example providers are Microsoft, ArcGIS, and Google.

GIS Server

URL is a standard REST-ful HTTP request :
http://dev.virtualEarth/REST/V1/Routes/Driving?wp.0=Seattle,WA&wp.1=Key West,FL ... etc...

Time

Receive Request

Collect GIS Data

Collect GIS Data

GIS-Addin DLL

Request Data

Receive Response

Update Simio

The Simio API is used to create and update Simio objects

Simio Engine

Simio

Simio UI

Response is generally JSON formatted text, that describes the route

GIS Service Provider: Microsoft's Bing Maps (VirtualEarth)

Bing Maps provides a GIS data service for getting information.

For example, the following URL:

http://dev.virtualearth.net/REST/V1/Routes/Driving?wp.0=Pittsburgh,PA&wp.1=Louisville,KY&optmz=distance&routeAttributes=routePath&key=Agggm6147gZ▓▓▓▓▓▓▓▓▓▓▓▓▓▓ Ma▓▓▓▓▓aToC

This my key, but you can get your own for free!

Will return Response GIS data in JSON format:
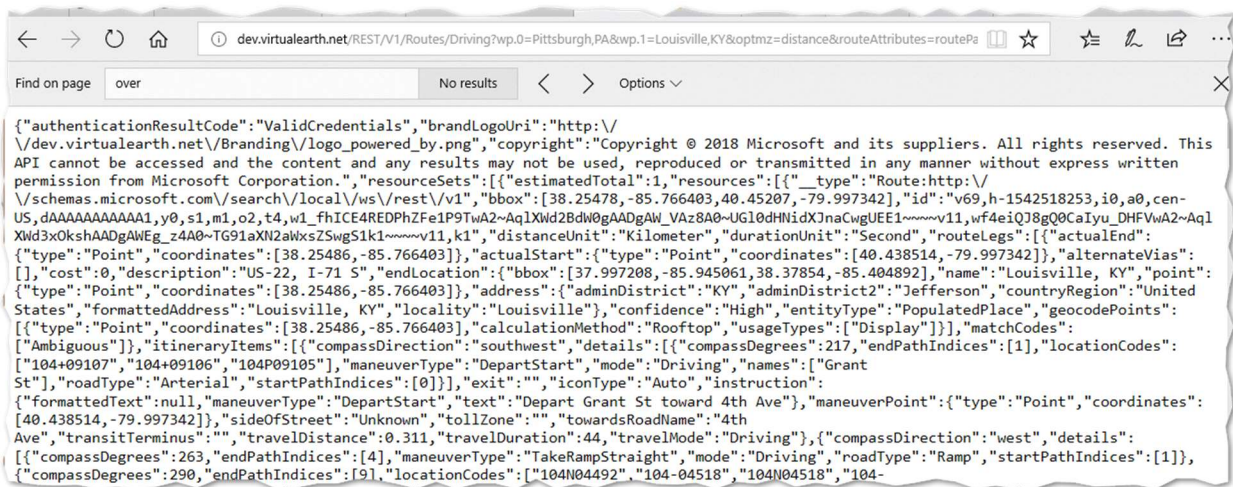
Figure 2 - A Portion of the Response Data in Unformatted JSON

Which can be viewed in Notepad++ with the JSON Viewer plugin:
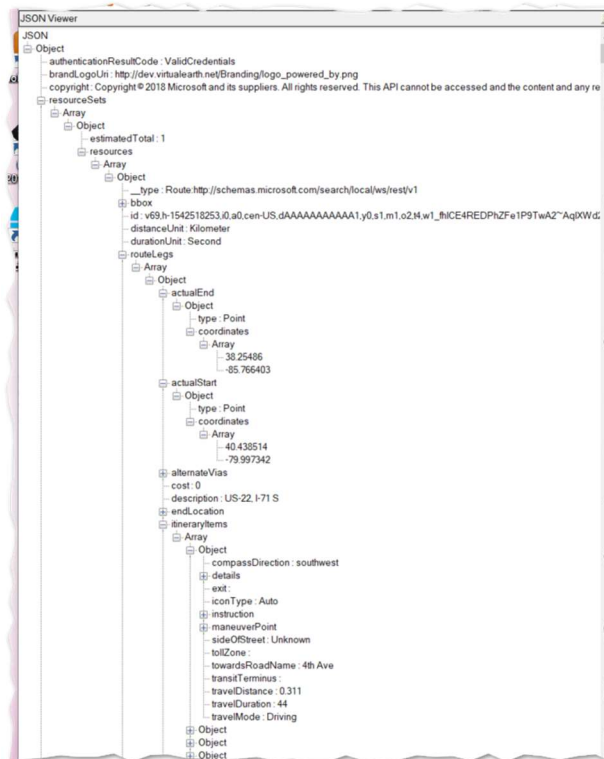
Response Information Detail

The response information contains all the information about all the paths from here to there. And it is a lot of data and can be controlled by a myriad of options. Refer to the Appendix for more detailed information on this interface.

The add-in processes this information, simplifies it according to the user's wishes, and then creates Simio nodes and links from the data.

From the Tree outline from Notepad++, you can see that file contains:

1. A bounding box, in lat/lon coordinates
2. An array of routeLegs
3. An array of routePaths

The routePath is an array of coordinates. Each adjoining coordinates (lat,lon) are referenced (by index) by a RouteLeg itineraryItems, which are essentially driving instructions.

JSON Viewer
- id : v69,h-1542518253,i0,a0,cen-US,dAAAAAAAAAAA1,y0,s1,m1,o2,t4,w1_fhICE4REDPhZFe1P9TwA2~AqIXW
- distanceUnit : Kilometer
- durationUnit : Second
- routeLegs
  - Array
    - Object
      - actualEnd
        - Object
          - type : Point
          - coordinates
            - Array
              - 38.25486
              - -85.766403
      - actualStart
      - alternateVias
      - cost : 0
      - description : US-22, I-71 S
      - endLocation
        - Object
          - bbox
          - name : Louisville, KY
          - point
          - address
          - confidence : High
          - entityType : PopulatedPlace
          - geocodePoints
          - matchCodes
      - itineraryItems
        - Array
          - Object
          - Object
            - compassDirection : west
            - details
            - exit :
            - iconType : Auto
            - instruction
              - Object
                - formattedText
                - maneuverType : RampToHighwayStraight
                - text : Take ramp for I-376 W / US-30 W / US-22 W toward Airport / Ft Pitt Bridge
            - maneuverPoint
              - Object
                - type : Point
                - coordinates
                  - Array
                    - 40.43605
                    - -79.99907
            - sideOfStreet : Unknown
            - signs
              - Array
                - Airport
                - Ft Pitt Bridge
                - I-279
            - tollZone :
            - transitTerminus :
            - travelDistance : 1.307
            - travelDuration : 68
            - travelMode : Driving
            - warnings
              - Array
                - Object
                  - origin : 40.43664,-80.00097
                  - : Minor

Figure 3 - Response Data Shown in a Tree Structure
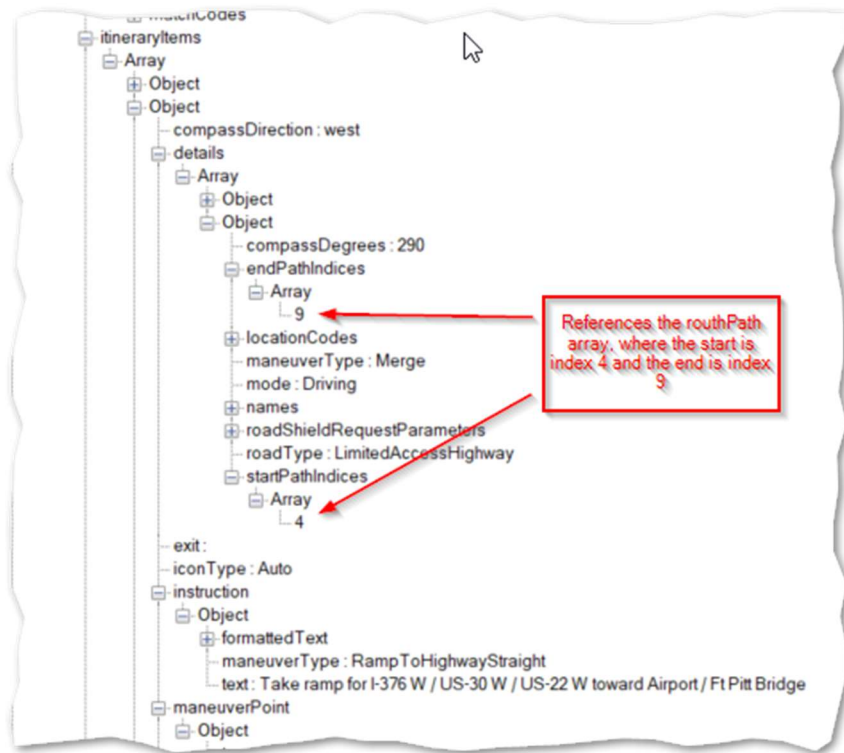
Figure 4 - Response Data Showing Itinerary Details

Data Processing

To help with the processing, the GisAddIn employs the BingMapsRESTToolkit package. This package can be located and installed in Visual Studio by invoking Tools > NuGetPackageManager .

PM>Install-Package BingMapsRESTToolkit

Bing Maps Key

This link:

https://msdn.microsoft.com/en-us/library/ff428642.aspx

explains the rules and the procedures for getting a Bing Maps Key.

The Simio Side

Now lets talk about the Simio side of this equation. From the docs, we have these objects:
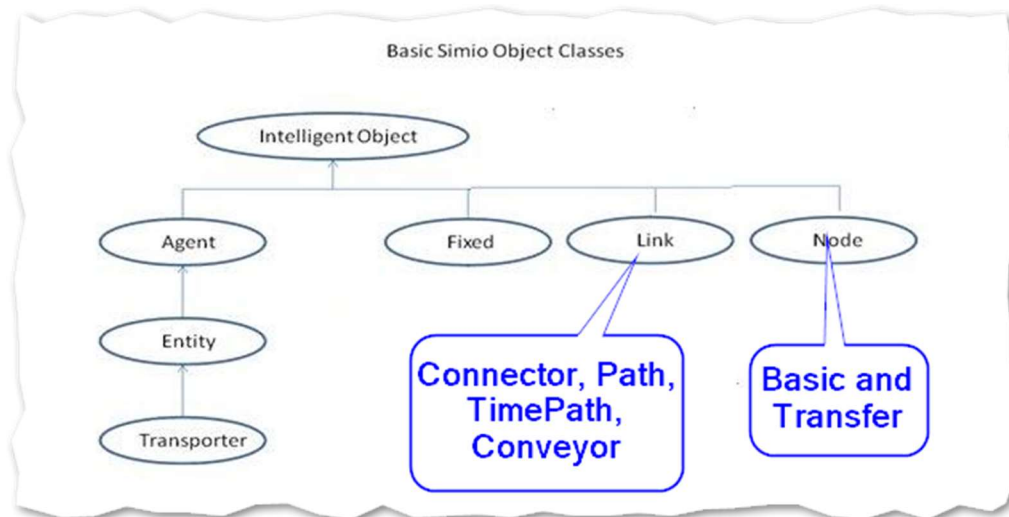
For this demonstration we are going to create two Basic Nodes (at the beginning and end), and we'll use the Path to connect them.

For this example, the Path Link was used. However, you should consider the TimePath as well, as it provides a way to set the time for the entire route. (This time was included in the Bing Maps response data).

Because each GIS Provider is going to have a different format for the response, a canonical set of data classes is provided in the GisAddIn, and it is these classes that the methods that build the Simio objects from. These can be found in the SimioMapData.cs file.

Methods within the GisHelpers.cs file include helper methods for serializing and deserializing JSON data. There is also a method used to convert lat/lon data to Simio's Facility coordinates.

A Note on Coordinates

Note that by convention we say lat-lon instead of the reverse, however usually the longitude is thought of as the "x" axis. Also remember that Simio is a 3D visualization model. As such, we are generally "looking down" on the X-Z coordinate space, where Z becomes more negative as it goes away from us. So generally, we are translating lat/lon to an (X,Y,Z) of something like (LON, 0, -LAT).

Additionally, with geo-coordinates we are translating positions on a sphere to the Simio plane. For small distances and/or close to the equator the differences are minor. But larger distances and particularly

when they extend into the high latitudes will not display correctly. In such cases you can modify the code and select a projection that is more suitable.

Appendix – References

Bing Maps (VirtualEarth) References

Using the REST Services with .NET

https://msdn.microsoft.com/en-us/library/jj819168.aspx

Microsoft has produced an Open Source project that incorporates 'best practices' for using their API and maintains it on GitHub as BingMapsRESTToolkit.

https://github.com/Microsoft/BingMapsRESTToolkit/

Here is a sample of the documentation; the The RouteRequest Class:

RouteRequest Class

A request that calculates routes between waypoints. Inherits from the BaseRestRequest class.

Methods

Name	Return Type	Description
Execute()	Task<Response>	Executes the request.
Execute(Action<int> remainingTimeCallback)	Task<Response>	Executes the request.
GetRequestUrl()	string	Gets the request URL to perform a query for route directions.

Properties

Name	Type	Description
RouteOptions	RouteOptions	Options to use when calculate route.
Waypoints	List<SimpleWaypoint>	Specifies two or more locations that define the route and that are in sequential order. A route is defined by a set of waypoints and viaWaypoints (intermediate locations that the route must pass through). You can have a maximum of 25 waypoints, and a maximum of 10 viaWaypoints between each set of waypoints. The start and end points of the route cannot be viaWaypoints.

Extended Properties

Some additional options have been added to the route request to increase its functionality.

Name	Type	Description
BatchSize	int	The maximium number of waypoints that can be in a single request. If the batchSize is smaller than the number of waypoints, when the request is executed, it will break the request up into multiple requests, thus allowing routes with more than 25 waypoints to be . Must by between 2 and 25. Default: 25.
WaypointOptimization	TspOptimizationType	Specifies if the waypoint order should be optimized using a travelling salesmen algorithm which metric to optimize on. If less than 10 waypoints, brute force is used, for more than 10 waypoints, a genetic algorithm is used. Ignores IsViaPoint on waypoints and makes them waypoints. Default: **false** **Warning:** If travel time or travel distance is used, a standard Bing Maps key will need to be required, not a session key, as the distance matrix API will be used to process the waypoints. This can generate a lot of billable transactions.

BingMapsRESTToolkit Samples

There are two sample projects provided; Console and WPF. The WPF is much more full-featured than the console, and is probably your best choice for seeing how the Toolkit operates and for grabbing pieces of code.

WPF Example Screen
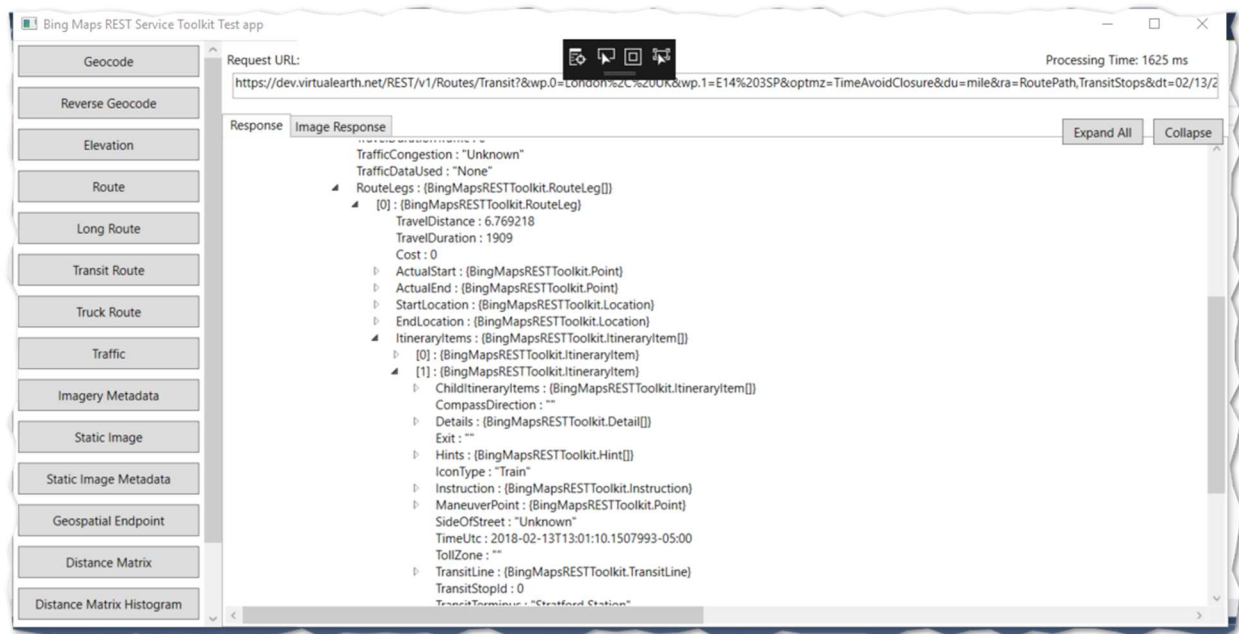
Note that there are Long, Transit, and Truck routes as well as plain Route.

Here is a sample of the "Route" Code, showing how to employ Route options:

```csharp
private void RouteBtn_Clicked(object sender, RoutedEventArgs e)
{
    var r = new RouteRequest()
    {
        RouteOptions = new RouteOptions(){
            Avoid = new List<AvoidType>()
            {
                AvoidType.MinimizeTolls
            },
            TravelMode = TravelModeType.Driving,
            DistanceUnits = DistanceUnitType.Miles,
            Heading = 45,
            RouteAttributes = new List<RouteAttributeType>()
            {
                RouteAttributeType.RoutePath
            },
            Optimize = RouteOptimizationType.TimeWithTraffic
        },
        Waypoints = new List<SimpleWaypoint>()
        {
            new SimpleWaypoint(){
                Address = "Seattle, WA"
            },
            new SimpleWaypoint(){
                Address = "Bellevue, WA",
                IsViaPoint = true
            },
            new SimpleWaypoint(){
                Address = "Redmond, WA"
            }
        },
        BingMapsKey = BingMapsKey
    };

    ProcessRequest(r);
}
```

And also some code for handling the ProcessRequest. Note that the method is an "Async" method, and uses the "await" keyword to keep your code from stopping other threads - like the UI – in your application.

```csharp
/// This method has a lot of logic that is specific to the sample.
/// To process a request you can easily just call the Execute method on the request.
/// </summary>
/// <param name="request"></param>
15 references
private async void ProcessRequest(BaseRestRequest request)
{
    try
    {
        RequestProgressBar.Visibility = Visibility.Visible;
        RequestProgressBarText.Text = string.Empty;

        ResultTreeView.ItemsSource = null;

        var start = DateTime.Now;
        //Execute the request.
        var response = await request.Execute((remainingTime) =>
        {
            if (remainingTime > -1)
            {
                _time = TimeSpan.FromSeconds(remainingTime);
                RequestProgressBarText.Text = $"Time remaining {_time} ";
                _timer.Start();
            }
        });

        RequestUrlTbx.Text = request.GetRequestUrl();
        var end = DateTime.Now;
        var processingTime = end - start;
        ProcessingTimeTbx.Text = string.Format(CultureInfo.InvariantCulture,
            "{0:0} ms", processingTime.TotalMilliseconds);

        var nodes = new List<ObjectNode>();
        var tree = await ObjectNode.ParseAsync("result", response);
        nodes.Add(tree);
        ResultTreeView.ItemsSource = nodes;
        ResponseTab.IsSelected = true;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
```

Simio

Forward Thinking

Appendix – References

USGS

https://www.usgs.gov

Bing Maps REST Services

https://msdn.microsoft.com/en-us/library/ff701713.aspx

BingMapsRESTToolkit (Preferred interface, according to Microsoft)

https://github.com/Microsoft/BingMapsRESTToolkit/

Google Web Services > Directions API

https://google-developers.appspot.com/maps/documentation/directions/