Google Maps API
Nov. 2024 \./ Max Kleiner

- Maps with out of the box demos
- Google Directions / Shell / Scripts
- OpenStreetMap OSM mapbox 
- https://github.com/maxkleiner/restcountries
- JSON, EdgeView2 and Geocoding
- This session shows you various ways of using Maps and directions in your application.

TDirections

- We use a simple class TDirections with 4 events.
- Component for Delphi, Python and Lazarus.
- I use an URI, a struct Tlatlong, a JSON and a HTTPGet object

In the archive
1274_GoogleDirForm2GeocodeWeather_General2request2.pas

you find the script, model and data you need, which works with Lazarus, Delphi, Jupyter and maXbox.

```
type TTlatlong = record
       lat, long: double;
       descript: string;
     end;
```

Google Maps Directions

- You can get directions for driving, public transit, walking, ride sharing, cycling, flight, or motorcycle on Google Maps. If there are multiple routes, the best route to your destination is blue, all other routes are gray.
- Some directions in Google Maps are in development and may have limited availability.
- Script Ref: http://www.softwareschule.ch/examples/directions3.htm
- https://sourceforge.net/projects/maxbox/files/Examples/EKON/EKON28/1274_GoogleDirForm2GeocodeWeather_General2request2.pas/download

Cross-platform component

- **TestUnit From Cologne to Graz**
- get geocoords: lat: 50.9473 – lon: 6.9503 Cologne
- get geocoords: lat: 47.0739 – lon: 15.4168 Graz
-
- latlong:= TAddressGeoCodeOSM(‘Gereonswall 66, Cologne, Germany’);
- latlong:= TAddressGeoCodeOSM(‘Hauptbahnhof, Graz, Austria’);
-
- UnitTest GeoCode

  https://www.google.com/maps/dir/
  050.94730,006.95031/047.07389,015.41674/@050.94730,006.95031,9z

  https://www.google.com/maps/dir/
  046.94724,007.45158/046.55764,015.64559/@046.94724,007.45158,9z

GUI or Console

Form64 Googel Directions WebForm5

Google Directions NavUtils5

From

Source Longitude: 6.950307406241427

Source Latitude: 50.9472988

6° 57' 1.107"E 50° 56' 50.276"N

To

Dest Longitude: 15.416744749733317

Dest Latitude: 47.0738881

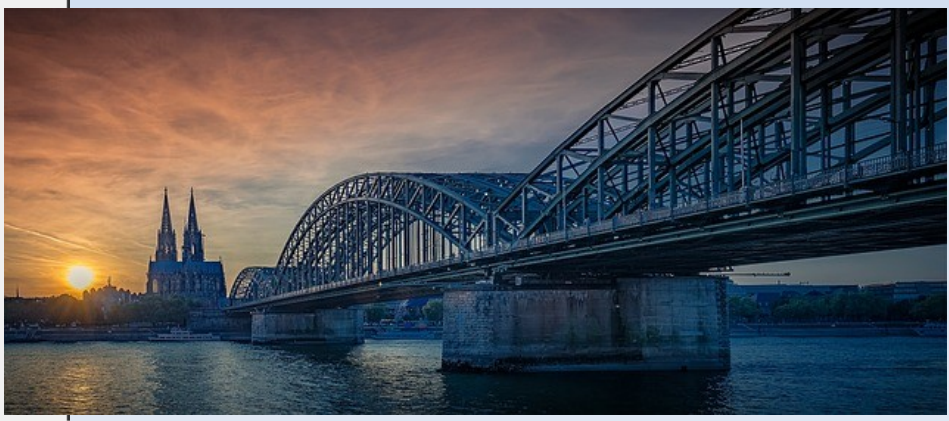
15° 25' 0.281"E 47° 4' 25.997"N

Check To See Link In Browser | Go to Google..

https://www.google.com/maps/dir/050.94730,006.95031/047.07389,015.41674/@050.9

Distance as Crow Flys =752.1km

GeoCoords: Dest: lat: 47.0739 - lon: 15.4167



```
latlong:=  TAddressGeoCodeOSM5('Hauptbahnhof, Graz, Austria');
writeln('OSM5 res back_: '+latlong.descript);
>>> OSM5 res back_: Coords: lat 47.07391  lng 15.41681 Hauptbahnhof, Europ

writeln('get geoCoords: '+format(' lat: %.4f - lon: %.4f',[latlong.lat,lat
OpenWeb('https://www.latlong.net/c/?lat='+flots(latlong.lat)+'&long='+flot
```

Be aware of

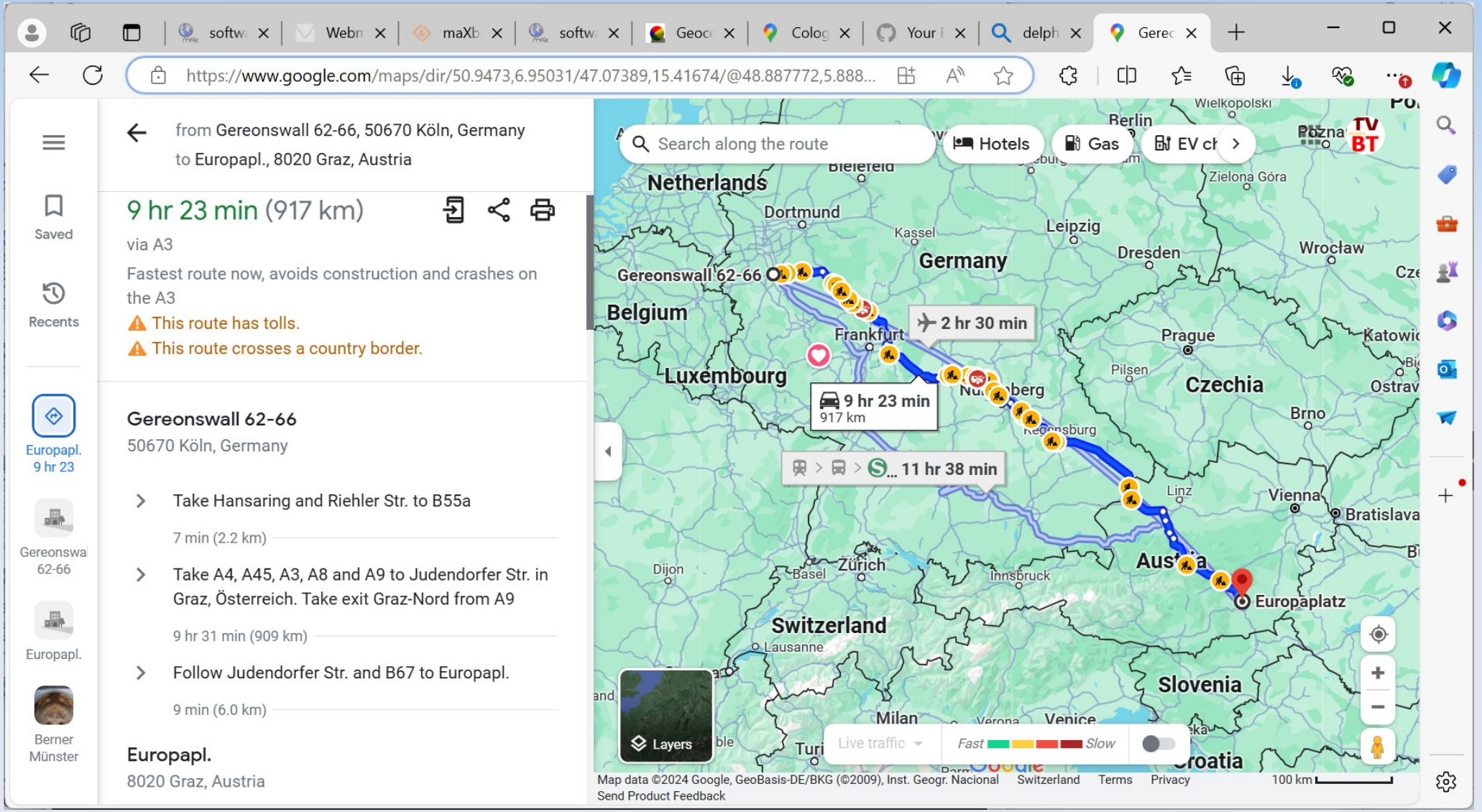
You should in particular verify that you have set a custom HTTP referrer or HTTP user agent (windown.useragent:=) that identifies your application, and that you are not overusing the service with massive bulk requests. Otherwise you get following message:

You have been blocked because you have violated theusage policyof OSM’s Nominatim geocoding service. Please be aware that OSM’s resources are limited and shared between many users. The usage policy is there to ensure that the service remains usable for everybody.

Important: Stay alert when you use directions on Google Maps. Always be aware of your surroundings to ensure the safety of yourself and others. When in doubt, follow actual traffic regulations and confirm signage from the road or path that you’re on.

Demo: https://jasontpenny.com/blog/2009/01/11/google-maps-in-a-twebbrowser-from-delphi-directions/

Graph Control



Code Control II

```
1  procedure TDirectionsBtnGoGoogleClick(Sender: TObject);
2  Var
3    LocFrom,LocTo:RNavigateLongLat;
4    Long,Lat:Double;
5  begin
6    LocFrom:=RNavigateLongLat.create;
7    Locto:=RNavigateLongLat.create;
8    Long:=RealFrmDegreeText(EdtLong.Text);
9    Lat:= RealFrmDegreeText(EdtLat.Text);
10   LocFrom.CreateDec(Long,Lat);
11   Long:=RealFrmDegreeText(EdtLong2.Text);
12   Lat:= RealFrmDegreeText(EdtLat2.Text);
13   LocTo.CreateDec(Long,Lat);
14   // { 0 Start 1 End 3 Center }
15   EdtGoogleLink.Text:=LocFrom.GoogleLinkDirectionsTo(LocTo, 0);
16   LBlCrowFlies.Caption:='Distance as CrowFlys =
17                  '+FormatFloat('0.0km',LocTo.MetresFrom(LocFrom)/1000);
18   if CBxGoNow.Checked then
19      LocFrom.GoGoogleDirectionsTo(LocTo, 0);
20 end;
```

Demo: 1234_NeuralNetwork2_XOR_sampleEU_EKON27.pas

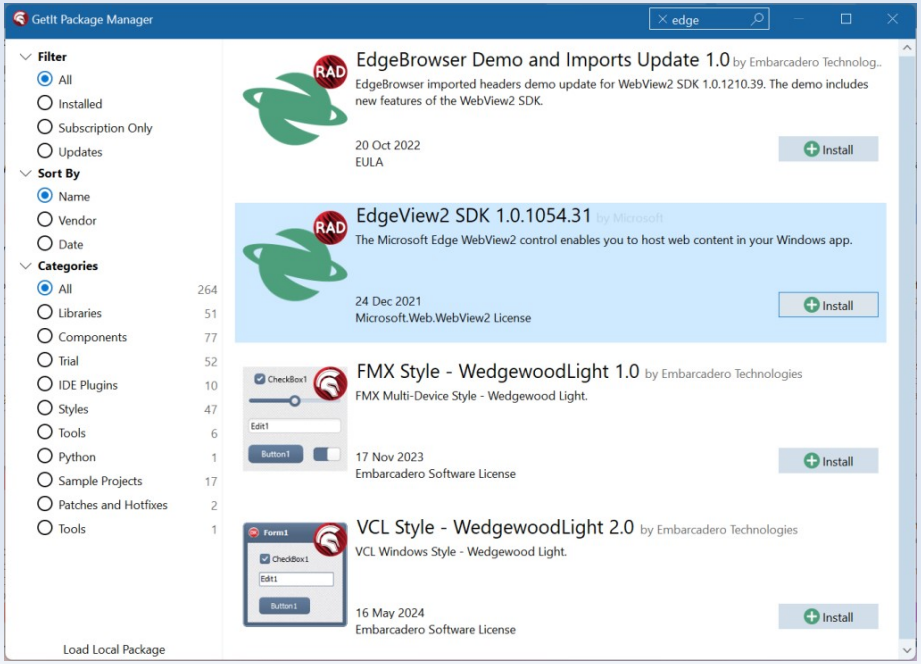
Other Packages

- IDE for Console or Terminal

•TGoMaps (Komponente, kommerziell, 599 € + MwSt.)
•TGoMaps Lite (Komponente, kostenlos, stark eingeschränkte Version von TGoMaps, dient als "Testversion" von TgoMaps)

•TGAgisGoogleMap und TGAgisGoogleGeo (Komponenten, kommerziell, 799 USD + MwSt.)
•TECMap (Komponente, kommerziell, 500 € + MwSt., Testkomponente auf Anfrage)
•TMS WebGMaps (Komponenten, kommerziell, 95 € + MwSt.)
•https://www.tmssoftware.com/site/tmsfncmaps.asp?r=vcl
•

•DelphiMaps (Komponenten, MPL 1.1)
•GMLib (Komponenten, LGPL 3.0)
•Google Maps Library v1 (closed)



RestClient Lib

- The REST library is available for all platforms that are supported by Delphi. The REST Library framework focuses on JSON as the representation format. XML is not explicitly supported.
- Use “TRESTClient“, “TRESTRequest” and “TRESTResponse” components.
- Notice that they are automatically connected together (RESTRequest has “Client” and “Response” properties).f
- Rest for API-s of GEO services. speech recognition, image classification, OpenCL, big data, data science, sentiment-analysis, computer vision2 with more or less SVG.

https://github.com/maxkleiner/neural-api/

Behind the Scene...

RestClient: URL, cookies, auth, security – RestRequest: resources, params - RestResponse: JSON or UTF8 parsing, binding

```
function TAddressGeoCodeOSM8(AURL, location, aApikey: string): tlatlong;
var Httpreq: THttpRequestC; httpres: string;
    Body: TMultipartFormBody;  jsn: TMcJsonItem;
begin
  httpreq:= THttpRequestC.create(self);
  httpreq.headers.add('Accept: application/json; charset=utf-8');
  httpreq.useragent:= USERAGENT5;
  httpreq.SecurityOptions:= [soSsl3, soPct, soIgnoreCertCNInvalid];
  try
   if httpreq.get(Format(AURL,[location])) then begin
     httpres:= (httpreq.Response.ContentAsString)
     writ('debug back '+FormatJson(httpres));
     jsn:= TMcJsonItem.Create;
     jsn.AsJSON:= httpres;
     result.lat:= jsn.at(0,'lat').asnumber;
     result.long:= jsn.at(0,'lon').asnumber;
     result.descript:= Format('Coords: lat %2.5f  lng %2.5f %s place_id:
                          [result.lat,result.long,jsn.at(0,'display_name')
                                              jsn.at(0,'place_id').asinteger])
    end else Writeln('APIError '+inttostr(Httpreq.Response.StatusCode2));
    //StrReplace(httpres, '[{', '{');
  finally
    writeln('Status3: '+gethttpcod(httpreq.Response.statuscode2))
    httpreq.Free;
    sleep(200)
    jsn.Free;
  end;
end;
```

Maps and API Keys

```
{$I .\NINJAAPIKEY.INC}

const
   N_APIKEY = NINJA_APIKEY;
   L_APIKEY = LAYER_APIKEY;
```

File NINJAAPIKEY.INC

```
Const
  NINJA_APIKEY = 'gwM+25ePKBKfgnz40Q+s+w==sFaryouAPIKey;
// insert your private ChatGTP API key here
  LAYER_APIKEY = 'DNwCF9Rf6y1AmSSednjn8ZyourAPIKey';
```

Call:

```
t_latlong:= TAddressGeoCodeOSM8(URL_APILAY_GEO,
                                'Athen', L_APIKEY);
```

Traps by Agent or Limits

- First we need to set the TIdHTTP property UserAgent (the UserAgent is what a browser uses to identify itself to the HTTP server) to an valid Agent, if you use the default value Mozilla/3.0 (compatible; Indy Library) you will get a awful message like this HTTP 1.1/ 403 Forbidden. so we can change this value to Mozilla/3.0 or to another valid agent.

  Then we need build the url to request the image from an location. a valid URL look like this

- http://maps.google.com/maps/api/staticmap?center=40.714728,-73.998672&zoom=12&size=400x400&sensor=false.
- A limit is enforced to prevent abuse and/or repurposing of the Static Maps API, and this limit may be changed in the future without notice. If you exceed the 24-hour limit or otherwise abuse the service, the Static Maps API may stop working for you temporarily. If you continue to exceed this limit, your access to the Static Maps API may be blocked.

Static Maps

```
var
  StreamData :TMemoryStream;
  JPEGImage  : TJPEGImage;
begin
  EditURL.Text:=buildUrl;//build the url with the params
  StreamData := TMemoryStream.Create;
  JPEGImage  := TJPEGImage.Create;
  try
    try
     idhttp1.Get(EditURL.Text, StreamData); //Send the request and get the image
     StreamData.Seek(0,soFromBeginning);
     JPEGImage.LoadFromStream(StreamData);//load the image in a Stream
     ImageMap.Picture.Assign(JPEGImage);//Load the image
    Except On E : Exception Do
     MessageDlg('Exception: '+E.Message,mtError, [mbOK], 0);
    End;
  finally
    StreamData.free;
    JPEGImage.Free;
  end;
end;
```

Use of the Google Static Maps API is subject to a query limit of 1000 unique (different) image requests per viewer per day. Since this restriction is a quota per viewer, most developers should not need to worry about exceeding their quota.

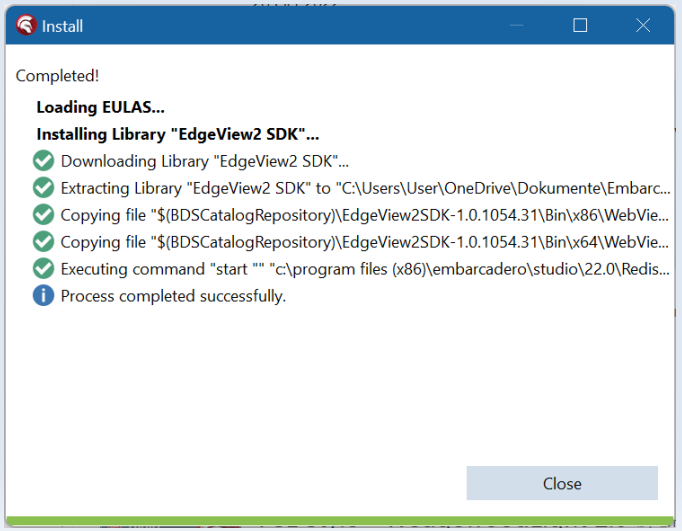
- https://developers.google.com/maps/documentation/maps-static/overview?hl=en

Form Create

```
type
  TfrmMain = class(TForm)
    WebBrowser1: TWebBrowser;
    LabelAddress: TLabel;
    PanelHeader: TPanel;
    ButtonGotoLocation: TButton;
    XPManifest1: TXPManifest;
    MemoAddress: TMemo;
    ButtonGotoAddress: TButton;
    LabelLatitude: TLabel;
    LabelLongitude: TLabel;
    Longitude: TEdit;
    Latitude: TEdit;
    CheckBoxTraffic: TCheckBox;
    CheckBoxBicycling: TCheckBox;
    CheckBoxStreeView: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure ButtonGotoAddressClick(Sender: TObject);
    procedure ButtonGotoLocationClick(Sender: TObject);
    procedure CheckBoxTrafficClick(Sender: TObject);
    procedure CheckBoxBicyclingClick(Sender: TObject);
    procedure CheckBoxStreeViewClick(Sender: TObject);
  private
    { Private declarations }
    HTMLWindow2: IHTMLWindow2;
  public
    { Public declarations }
  end;
```

EdgeView2 SDK

- McJson use a Simple Object-Pascal native code using TList as internal data structure with a Single-pass string parser and is just one unit (McJSON) and just one class(TMcJsonItem) integrated in maXbox or Lazarus.
- RAD Studio 10.4 Sydney brings support for working with web content through the Chromium-based Edge WebView2 browser control in VCL applications via the new TEdgeBrowser component.
- Demo: *1312_API_Demo64_5_javascript_maps2edge.txt*




JS-Solution

- in this sample application you can use the traffic layer, Bicycling layer and the street View Control to activate a panorama view.
- Integrate dynamic, interactive, custom maps, location data and geospatial data into your apps by embedding JS:

```
48  const
49  HTMLStr: String = //i put The code for the web page page wich load the google maps in
50  '<html> '+
51  '<head> '+
52  '<meta name="viewport" content="initial-scale=1.0, user-scalable=yes" /> '+
53  '<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=true">
54  '<script type="text/javascript"> '+
55  ''+
56  ''+//Declare the globals vars to be used in the javascript functions
57  '  var geocoder; '+
58  '  var map;  '+
59  '  var trafficLayer;'+
60  '  var bikeLayer;'+
61  ''+
```


Web Platforms

- As a Delphi GUI/Console (with or without JS or EdgeView):
- **TMS FNC Maps**
- Universal Delphi & C++ Builder components for mapping, geographical data, timezone visualization, routes and directions calculation. Use your mapping service of choice like Google Maps, OpenStreetMap, Leaflet, Azure Maps, Here Maps in VCL, FMX and WEB core apps
- **PositionStack** API
  https://colab.research.google.com/github/maxkleiner/maXbox/blob/master/EKON24_SimpleImageClassificationCPU.ipynb

  As static maps:

  You can use online map services (e.g. Google Map or Microsoft Bing); they both provide static map services alongside their default dynamic map interfaces which work inside a web browser.
  They also support geo-location queries; that is, you can send an address to their server, and retrieve a coordinate.

https://github.com/maxkleiner/maXbox/blob/master/objectdetector3.ipynb
https://blogs.embarcadero.com/this-is-how-to-make-a-geocoding-app-in-5-minutes/

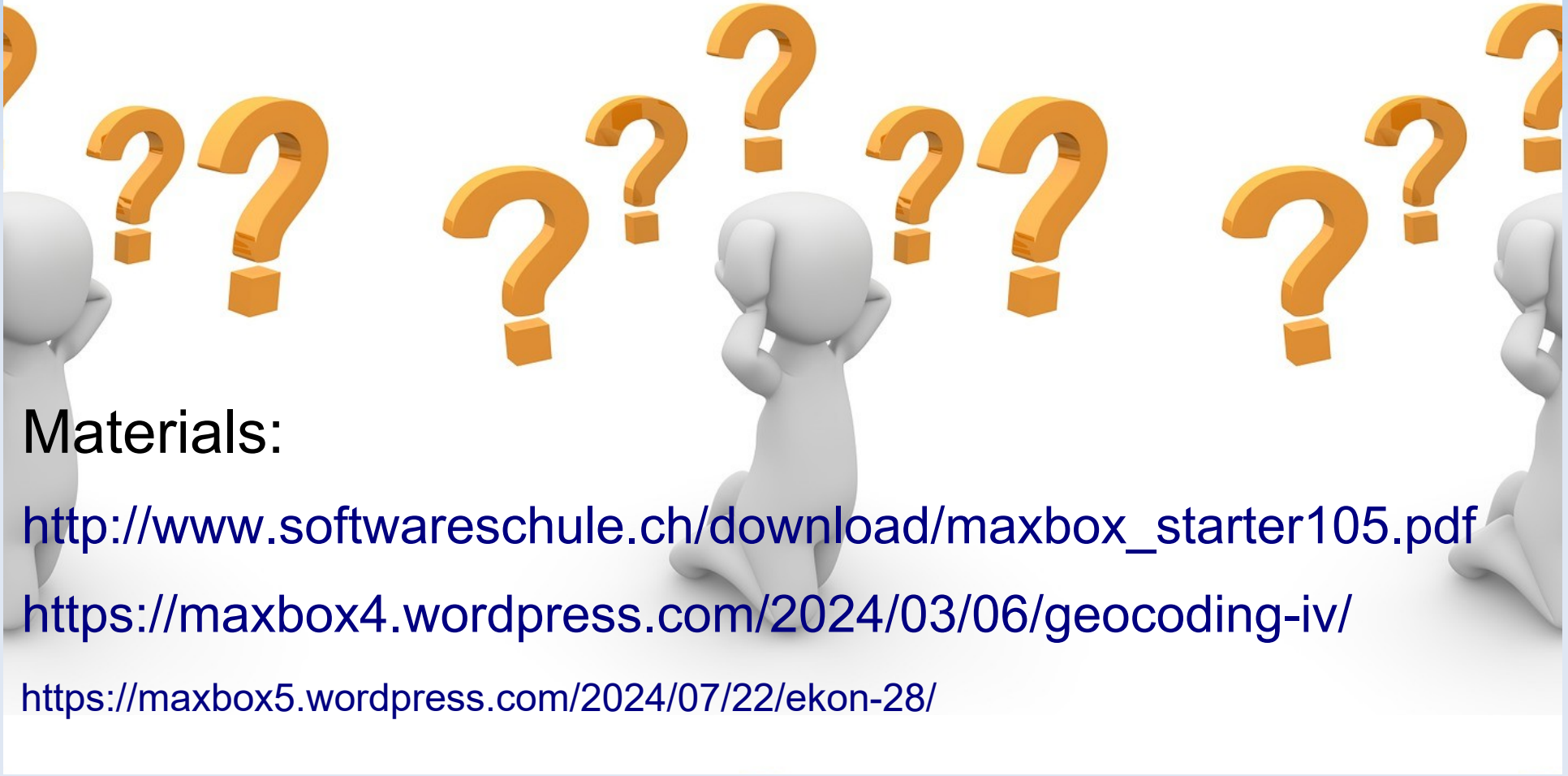

Conclusion

- Integrate a component like TDirections class or a service (API) like OSM on a Form
- Integrate a component like TDirections on a Form and open external Browser
- Integrate a component like TDirections in a Shell, CLI and embed or open internal Browser like EdgeView
- **Different Map Solutions (static or dynamic):**
- Google Maps
  Maps, Directions, Geocoding, Static Maps, TimeZone, Elevation, Location, Places
- Microsoft Azure Maps
  Maps, Geocoding, Directions, Static Maps, TimeZone, Places
- MapBox
  Maps, Geocoding, Directions, Static Maps
- OpenLayers (OpenStreetMap, Nominatim)
  Maps, TileServers, Geocoding

Method: Compnent or API
Maps: **Offline** or Online
View: Form or Browser


Geo Cod

Thanks for coming!



Materials:

http://www.softwareschule.ch/download/maxbox_starter105.pdf

https://maxbox4.wordpress.com/2024/03/06/geocoding-iv/

https://maxbox5.wordpress.com/2024/07/22/ekon-28/