

GEOCODER INTERFACE

AUFBAU

Schnittstelle zwischen der WPF Anwendung und dem Node.js basierten Geocoder Rest Service.

Das Interface basiert auf der RestSharp (restsharp.org) zum Erstellen der HTTP Requests, und der Newtonsoft JSON Library (www.newtonsoft.com) zum Serialisieren der Daten.

VERWENDUNGSZWECK

Als Schnittstelle zwischen dem Rest Service und der C# Applikation bzw. dem Arcgis Plugin

LIBRARIES

Das Interface basiert auf der RestSharp Library zum Erstellen der http Requests, und der Newtonsoft JSON Library zum (De-)Serialisieren der Daten.

RestSharp enthält in seiner letzten Version zwar mit Methoden zum Serialisieren von JSON Objekten, allerdings kam es zu Problemen mit komplexeren Objekten.

FUNKTIONSWEISE

Über das Rest Service sind drei Funktionen abgreifbar:

- Die Liste der implementierten Geocoder
- Geocoding einer Adresse auf deren Koordinaten
- Reverse Geocoding von Koordinaten zu einer Adresse

In den beiden letzten Fällen wird in JSON Objekt von folgendem Format retourniert:

```
{  "properties": {
    "address": "",
    "country": "",
    "zip": "",
    "city": "",
    "street": "",
    "housenumber": ""},
  "geometry": {
    "type": "Point",
    "coordinates": [LAT, LNG]
  },
  "epsg": ""}
```

Das Interface deserialisiert und retourniert in den entsprechenden Methoden dieses Object mit der gleichen Struktur. Die Geocoding Provider werden als Liste übergeben.

RestAPI request = new RestAPI(); // Initialisierung der Request Klasse

servicelist providers = request.getServices(); // Abfragen der Verfügbaren Geocoder

codingObject myadress = new codingObject(); *// Initialisiert ein Objekt, das vom Aufbau her dem JSON Objekt des Geocoders entspricht.*

myadress.properties.address = "Musterstraße 11, 1111 Musterville"; *//Adresse wird als string zugewiesen.*

codingObject adressobject = request.getCoordinates(provider, adresse); *// die Methode für das (forward) Geocoding, sendet das serialisierte Properties Object als GET Request und bekommt ein Object im o.g. JSON Format retour. Dieses wird deserialisiert von der Methode zurückgegeben. Die einzelnen Properties können dann direkt abgegriffen werden.*

codingObject adressobject = request.getAdress(providers.providers[0], myadress); *// die Methode für das reverse Geocoding sendes nur das serialiserte Geometry-Object und erhält analog zum (forward) Geocoding ein JSON Object als Response des REST Services zurück, dessen Properties anschließend abgegriffen werden können.*