Instantiation

tzinfo

location

weather

Geo

The Geo plugin gives you a handful of location-based features.

Namespace: `\Web`
File location: `lib/web/geo.php`

Instantiation

Return class instance

```
$geo = \Web\Geo::instance();
```

The Geo class uses the Prefab (prefab-registry) factory wrapper, so you can grab the same instance of that class at any point of your code.

tzinfo

Returns information about specified Unix time zone

```
array tzinfo ( string $zone )
```

This method accepts any supported timezone strings (http://php.net/manual/en/timezones.php) and returns an array as follows:

array (size=5) :

- **offset**: The time zone offset measured relatively to GMT
- **country**: ISO 3166-1-alpha-2 country code as per PHP DateTimeZone::getLocation (http://www.php.net/manual/en/datetimezone.getlocation.php)
- **latitude**: latitude as per PHP DateTimeZone::getLocation (http://www.php.net/manual/en/datetimezone.getlocation.php)
- **longitude**: longitude as per PHP DateTimeZone::getLocation (http://www.php.net/manual/en/datetimezone.getlocation.php)
- **dst**: A boolean flag that reflects daylight savings time adjustment

For example:

```
/** @var \Web\Geo $geo */
$geo = \Web\Geo::instance();
var_dump($geo->tzinfo('Australia/Darwin'));
/* returns:
array (size=5)
  'offset' => float 9.5
  'country' => string 'AU' (length=2)
  'latitude' => float -12.46667
  'longitude' => float 130.83333
  'dst' => boolean false
*/
```

location

Return geolocation data based on a specified or auto-detected IP address

```
array|FALSE location ( [ string $ip = NULL] )
```

Example:

```
$geo = \Web\Geo::instance();
var_dump($geo->location()); // locate client IP
/* returns:
array (size=12)
  'request' => string '85.2.88.43' (length=10)
  'credit' => string 'Some of the returned data includes GeoLite data created by MaxM
ind, available from <a href=\'http://www.maxmind.com\'>http://www.maxmind.com</a>.'
(length=145)
  'city' => string 'Lutry' (length=5)
  'area_code' => string '0' (length=1)
  'dma_code' => string '0' (length=1)
  'country_code' => string 'CH' (length=2)
  'country_name' => string 'Switzerland' (length=11)
  'continent_code' => string 'EU' (length=2)
  'latitude' => string '46.504902' (length=9)
  'longitude' => string '6.6852' (length=6)
  'region_code' => string '23' (length=2)
  'region_name' => string '23' (length=2)
*/
```

weather

Return weather data based on specified latitude/longitude

```
array|FALSE weather ( float $latitude, float $longitude, string $key )
```

Example:

```php
$geo = \Web\Geo::instance();
$loc = $geo->location('95.143.172.183'); // locate specific IP
$key = 'SECRET'; // The API key
var_dump($geo->weather($loc['latitude'],$loc['longitude'],$key));
/* returns:
array (size=17)
  'weatherCondition' => string 'n/a' (length=3)
  'clouds' => string 'few clouds' (length=10)
  'observation' => string 'ETHF 171620Z 33002KT 9999 FEW160 SCT210 05/00 Q1025 BLU+'
(length=56)
  'windDirection' => int 330
  'ICAO' => string 'ETHF' (length=4)
  'elevation' => int 181
  'countryCode' => string 'DE' (length=2)
  'cloudsCode' => string 'FEW' (length=3)
  'lng' => float 9.2666666666667
  'temperature' => string '5' (length=1)
  'dewPoint' => string '0' (length=1)
  'windSpeed' => string '02' (length=2)
  'humidity' => int 70
  'stationName' => string 'Fritzlar' (length=8)
  'datetime' => string '2013-12-17 16:20:00' (length=19)
  'lat' => float 51.116666666667
  'hectoPascAltimeter' => int 1025
*/
```