Location Class

Contains methods for accessing the component fields of geolocation compound fields.

Namespace

system

Usage

Each of these methods is also equivalent to a read-only property. For each getter method you can access the property using dot notation. For example, `myLocation.getLatitude()` is equivalent to `myLocation.latitude`.

You can't use dot notation to access compound fields' subfields directly on the parent field. Instead, assign the parent field to a variable of type `Location`, and then access its components.

```
Location loc = myAccount.MyLocation__c;
Double lat = loc.latitude;
```

Important

"Location" in Salesforce can also refer to the Location standard object. When referencing the Location object in your Apex code, always use `Schema.Location` instead of `Location` to prevent confusion with the standard Location compound field. If referencing both the location object and the Location field in the same snippet, you can differentiate between the two by using `System.Location` for the field and `Schema.Location` for the object.

Example

```
// Select and access the Location field. MyLocation__c is the name of a geolocation field
Account[] records = [SELECT id, MyLocation__c FROM Account LIMIT 10];
for(Account acct : records) {
   Location loc = acct.MyLocation__c;
   Double lat = loc.latitude;
   Double lon = loc.longitude;
}

// Instantiate new Location objects and compute the distance between them in different way
Location loc1 = Location.newInstance(28.635308,77.22496);
Location loc2 = Location.newInstance(37.7749295,-122.4194155);
Double dist = Location.getDistance(loc1, loc2, 'mi');
Double dist2 = loc1.getDistance(loc2, 'mi');
```

- Location Methods

Location Methods

- **getDistance(firstLocation, secondLocation, unit)**
  Calculates the distance between the two specified locations, using an approximation of the haversine formula and the specified unit.
- **getLatitude()**
  Returns the latitude field of this location.
- **getLongitude()**
  Returns the longitude field of this location.
- **newInstance(latitude, longitude)**
  Creates an instance of the `Location` class, with the specified latitude and longitude.

getDistance(toLocation, unit)

Calculates the distance between this location and the specified location, using an approximation of the haversine formula and the specified unit.

Signature

```
public Double getDistance(Location toLocation, String unit)
```

Parameters

toLocation

Type: Location

The `Location` to which you want to calculate the distance from the current `Location`.

unit

Type: String

The distance unit you want to use: `mi` or `km`.

Return Value

Type: Double

getDistance(firstLocation, secondLocation, unit)

Calculates the distance between the two specified locations, using an approximation of the haversine formula and the specified unit.

Signature

```
public static Double getDistance(Location firstLocation, Location secondLocation, String unit)
```

Parameters

firstLocation

Type: Location

The first of two locations used to calculate distance.

secondLocation

Type: Location

The second of two locations used to calculate distance.

unit

Type: String

The distance unit you want to use: `mi` or `km`.

getLatitude()

Returns the latitude field of this location.

Signature

```
public Double getLatitude()
```

Return Value

Type: Double

getLongitude()

Returns the longitude field of this location.

Signature

```
public Double getLongitude()
```

Return Value

Type: Double

newInstance(latitude, longitude)

Creates an instance of the `Location` class, with the specified latitude and longitude.

Signature

```
public static Location newInstance(Decimal latitude, Decimal longitude)
```

Parameters

latitude

Type: Decimal

longitude

Type: Decimal

Return Value

Type: Location

Commerce Cloud
Lightning Design System
Einstein
Quip

Trailhead
Sample Apps
Podcasts
AppExchange

Blog
Salesforce Admins
Salesforce Architects