**Title:** Redis GEODIST

***Excerpt:*** The GEODIST command returns the distance between the given two members stored at a specified geospatial index. By default, the distance is returned in meters. It is possible to specify the unit in feet, kilometers, or miles as well.

**Permalink:** redis-geodist

**Category:** Redis

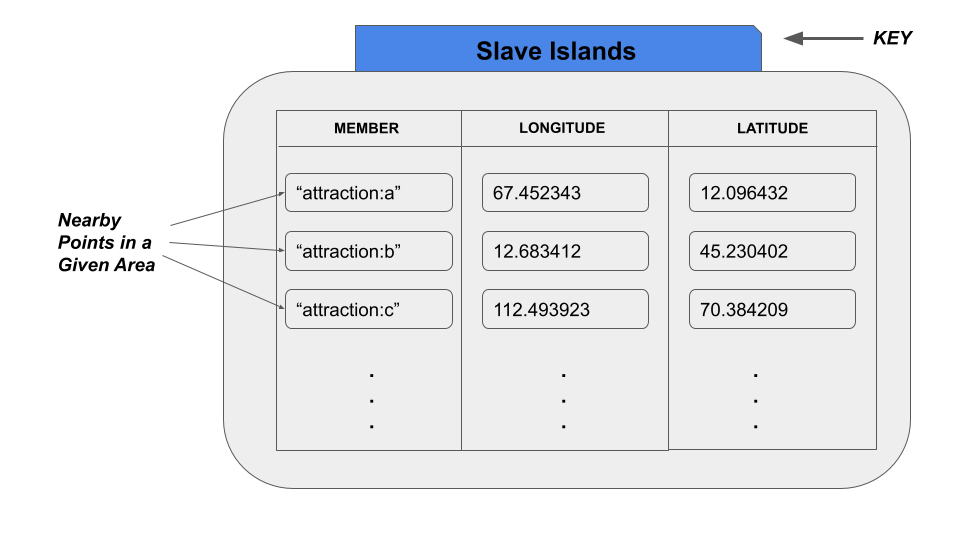
# 

# 

# 

# Geospatial Indexes in Redis

Redis starts supporting geospatial data types from version 3.2.0. This is one of the unique features that Redis offers over the other data stores. The geospatial data structure can hold coordinates of a given location. With the rich set of operations, it is possible to query locations and their coordinates based on different criteria.



Each member is pushed into a Redis sorted set referenced by a key which enables you to query those geospatial items efficiently and in an ordered manner at a later time. One of the best usages of this data structure is that you can easily query the distance between two geo-locations(members) using the GEODIST command introduced by Redis. This will be a detailed guide on the GEODIST command, syntax, and its use cases.

## Distance Calculation

Redis uses the Haversine mathematical formula to calculate the distances for the geospatial indexes which assumes that the earth is a perfect sphere. We all know that the earth is not a perfect sphere. So, a 0.5% error can be introduced by the distance calculation. This can be ignored for applications like social media, nearby place finders, etc. But for error-critical applications, this should be used with care.

# The GEODIST Command

The GEODIST command is used to calculate the distance between two members stored at a given geospatial key. This command accepts two mandatory member parameters and a geospatial key(index) which points to a sorted set. The distance is returned in meters(m). Optionally, you can specify the unit as well. It supports kilometers, miles, and feet too.

The syntax of the GEODIST command is as follows.

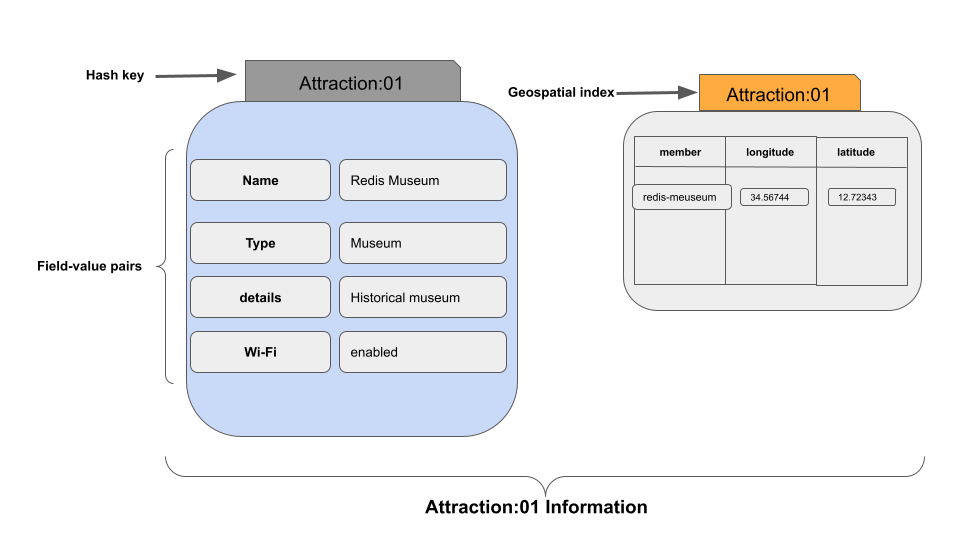
| GEODIST geospatial\_index/key member\_1 member\_2 [M | KM | FT | MI] |
| --- |

It returns the distance as a double value. The value depends on the unit that you’ve specified. Whenever the command encounters non-existing or missing members, it returns NULL.

## Use Case - Get the Distance Between Two Attractions

Let’s assume that a tourist company introduced a mobile application to find the details of attractions in a given country. The mobile application uses the Redis data store at its backend to store the details of attractions including their geospatial coordinates.

They have been using the Redis hashes to store the basic information related to the attractions. For the geolocation coordinate part, they are using Redis geospatial data structure.



Let’s create a new geospatial index that contains two members.

| geoadd attractions:newyork 23.596920 67.104923 "lazy mountain" 134.694013 45.394022 "sabra river" |
| --- |

Now the *“lazy mountain”* and *“sabra river”* members should be stored in a sorted set which is referenced by the key *attractions:newyork.*

Let’s use the GEODIST command to calculate the distance between the above two members. By default, it should return the distance in meters.

| geodist attractions:newyork "lazy mountain" "sabra river" |
| --- |

As you could see that we have specified the geospatial index as *attractions:newyork.* Next the two members as strings.



As expected, the output is returned in meters. The double data type has been used to represent the distance.

Retrieve the Distance in Kilometers

It is possible to retrieve the above distance in kilometers as well. You just need to pass the optional argument *‘km’* as follows.

| geodist attractions:newyork "lazy mountain" "sabra river" km |
| --- |

Output:



Retrieve the Distance in Miles

The ‘mi’ argument should be specified to calculate the distance between the given members in miles.

| geodist attractions:newyork "lazy mountain" "sabra river" mi |
| --- |

Output:



Retrieve the Distance in Feet

Whenever you need to know the number of feet from one member to another, just specify the unit as *‘ft’.*

| geodist attractions:newyork "lazy mountain" "sabra river" ft |
| --- |

Output:



Whenever you specify non-existing members, the GEODIST command should return null. Let’s specify two non-existing members as follows.

| geodist attractions:newyork "lazy" "sabra" |
| --- |

Output:



If the specified key doesn’t exist, the command will return null.

| geodist attractions:washington "lazy mountain" "sabra river" ft |
| --- |

In the above example, the *attractions:washington* key isn’t available. So, the output is NULL as shown in the following.



# Conclusion

In summary, the Redis geospatial data type has been introduced to hold coordinates of geo-locations. As mentioned, the longitude and latitude values are stored for a given member. This is represented as a sorted set in the memory. Several commands are available to operate on geospatial data types. The GEODIST is one of the useful commands that operate on geospatial indexes to return the distance between specified members. As discussed in the introduction section, the distance is returned in meters by default. The GEODIST command accepts optional arguments to return the distance in other units as well such as kilometers, feet, and miles.