Geospatial Search

MongoDB offers a number of indexes and query mechanisms to handle geospatial information. This section demonstrates how to create and use geospatial indexes with the Ruby driver.

The examples on this page use a sample collection called `restaurants` in the `test` database. A sample dataset is available for download.

The following is a sample document in the `restaurants` collection:

```
{
  "address": {
     "building": "1007",
     "coord": [ -73.856077, 40.848447 ],
     "street": "Morris Park Ave",
     "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
     { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
 { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

The following example creates a `2dsphere` index on the `address.coord` field:

```
client = Mongo::Client.new([ '127.0.0.1:27017' ], :database => 'test' )
client[:restaurants].indexes.create_one( { 'address.coord' => '2dsphere' })
```

Once the index is created, you can use several operators to query against it, including the $near, $geoWithin, and $geoIntersects operators. The following example uses the `$near` operator to find all restaurants within 500 meters of the given coordinates.

```
client = Mongo::Client.new('mongodb://127.0.0.1:27017/test')
collection = client[:restaurants]

collection.find(
    { 'address.coord' =>
      { "$near" =>
        { "$geometry" =>
          { "type" => "Point",  "coordinates" => [ -73.96, 40.78 ] },
            "$maxDistance" => 500
        }
      }
    }
  ).each do |doc|

    #=> Yields a BSON::Document.

  end
```

To find all documents with a location within the perimeter of a given polygon, use the `$geoWithin` operator:

```
client = Mongo::Client.new('mongodb://127.0.0.1:27017/test')
collection = client[:restaurants]

collection.find(
    { "address.coord" =>
      { "$geoWithin" =>
       { "$geometry" =>
             { "type" => "Polygon" ,
            "coordinates" => [ [ [ -73, 40 ], [ -74, 41 ], [ -72, 39 ], [ -73, 40 ] ] ]
         }
        }
      }
    }
  ).each do |doc|

    #=> Yields a BSON::Document.

  end
```