ARYATECHNO

MongoDB - Aggregation

Topics : MongoDB
Written on December 30, 2023

MongoDB Aggregation is a powerful framework for transforming and manipulating documents within a collection. Aggregation pipelines allow you to process data in a step-by-step manner, performing various operations on the documents. Here's an overview of MongoDB Aggregation:

Aggregation Pipeline Stages:

1. **$match Stage:**
   - Filters the documents based on specified criteria.

   ```
   db.collection_name.aggregate([ { $match: { field: value } } ]);
   ```

2. **$project Stage:**
   - Shapes the documents by specifying fields to include or exclude.

   ```
   db.collection_name.aggregate([ { $project: { included_field: 1, excluded_field: 0 } } ]);
   ```

3. **$group Stage:**
   - Groups documents based on specified criteria and calculates aggregate values.

   ```
   db.collection_name.aggregate([ { $group: { _id: "$field", count: { $sum: 1 } } } ]);
   ```

4. **$sort Stage:**
   - Sorts the documents based on specified fields and order.

   ```
   db.collection_name.aggregate([ { $sort: { field: 1 } } ]);
   ```

5. **$limit Stage:**
   - Limits the number of documents passed to the next stage.

   ```
   db.collection_name.aggregate([ { $limit: 10 } ]);
   ```

6. **$skip Stage:**

    - Skips a specified number of documents before passing them to the next stage.

    ```
    db.collection_name.aggregate([ { $skip: 5 } ]);
    ```

7. **$unwind Stage:**

    - Deconstructs an array field from the input documents, creating a new document for each array element.

    ```
    db.collection_name.aggregate([ { $unwind: "$array_field" } ]);
    ```

8. **$lookup Stage:**

    - Performs a left outer join to another collection in the same database.

    ```
    db.collection_name.aggregate([ { $lookup: { from: "other_collection", localField:
    "field_in_input_collection", foreignField: "field_in_other_collection", as: "new_field" } } ]);
    ```

9. **$project Stage (again):**

    - Further shapes the documents after preceding stages.

    ```
    db.collection_name.aggregate([ { $project: { final_field: "$field" } } ]);
    ```

Aggregation Example:

```
db.sales.aggregate([
  {
    $match: { date: { $gte: ISODate("2023-01-01"), $lt: ISODate("2023-02-01") } }
  },
  {
    $group: {
      _id: "$product",
      totalSales: { $sum: "$quantity" },
      averagePrice: { $avg: "$price" }
    }
  },
  {
    $sort: { totalSales: -1 }
  },
  {
    $limit: 5
  }
]);
```

This example calculates total sales and average price per product for a specific date range, sorts the results by total sales in descending order, and limits the output to the top 5 products.