

## Database

### 1. Cache:

Since Redis does not support object as a query term, we used MongoDB as a cache directly. Each time a client enters a search query, it will first look at Mlab MongoDB store. If the query is found, it would fetch the response and return to the client. Otherwise, the get request would call Yelp and Google Custom Search APIs and then load the search result to a collection called 'search' in MongoDB.

### 2. Data Schemas:

Since authentication and OAuth is handled by Auth0, we do not need the user schema for Mongoose anymore.

As previously introduced, the tripSchema is storing the userId of user who created it, its own name, restaurants the user visited and a isComplete field for status check.

```
const tripSchema = new Schema({
  userId: { type: String, required: true }, //store the user who created this trip
  name: { type: String, required: true }, //store the name of the trip
  restaurants: { type: Array, required: false }, //store the restaurants visited during this trip
  isComplete: { type: Boolean, required: false } //whether this trip is current(this is optional)
});
```

### 3. Sequence for Cache:

