

MongoDB Assignment

Guidelines:

- i. This assignment is mandatory for everyone*
- ii. There will only be a single attempt for each exam and **no deadline extension in case of assignments***
- iii. Any case of unfair means or **plagiarism would lead to debarring** in final placements without any further consideration.*
- iv. The assignment solutions should be uploaded on Github and links to Github repository links should be shared with the coach for code review. Make sure to add appropriate comments in code wherever possible.*

Problem 1:

Build an API to add and retrieve college student details from a database

Steps:

1. Create a connection with local or remote Mongo database
2. Build a model in express.js for student details
The model should have following parameters of type string:-
name, collegeName, location
3. Next build a route as shown below and write logic to add student details sent from API into MongoDB database

Request URL:- `http://localhost:3000/student/add`

Request Method:- POST

Request JSON:-

```
{
  "studentFirstName": "Ramesh",
  "collegeName": "IIT",
  "location": "Mumbai"
}
```

Response JSON:-

```
{  
  "result": "Success"  
}
```

4. Once student details are saved to database, build another API which can be used to get stored student details based on student name as shown below:-

GET localhost:3000/student/getStudentDetails?name=Ramesh

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Ramesh	
Key	Value	Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "name": "Ramesh",  
3   "collegeName": "I.I.T.",  
4   "location": "Mumbai"  
5 }
```

Status: 200 OK Time: 986 ms

Request URL:- *localhost:3000/student/getStudentDetails?name=Ramesh*

Request Method:- GET

Query Param:- name

Response JSON:-

```
{
  "name": "Ramesh",
  "collegeName": "I.I.T.",
  "location": "Mumbai"
}
```

Learnings:

How to make connection to mongodb database using express.js

Problem 2:

Build a quotes generation API

Steps:

1. Create a connection with local or remote Mongo database
2. Build a model in express.js for quotes details
The model should have following parameters of type *string*:-
quote, author
3. Next build a API route as shown below and write logic to add a quote sent from API below into MongoDB database

The screenshot displays a REST client interface. At the top, the request method is set to **POST** and the URL is `http://localhost:3000/quote/add`. A red arrow points to the **POST** label with the label **Request Method**. Below the URL bar, the **Body** tab is selected, showing a JSON payload: `{ "quote": "I was never afraid of failure; for I would sooner fail than not be among the greatest.", "author": "JOHN KEATS" }`. A red arrow points to this payload with the label **Data to be sent with API**. The bottom section shows the **Test Results** tab with a status of **200 OK** and a time of **220 ms**. The response body is displayed in **JSON** format: `{ "result": "success" }`. A red arrow points to this response with the label **API Response**.

Request URL:- `http://localhost:3000/quote/add`

Request Method:- POST

Request JSON:-

```
{
  "quote": "I was never afraid of failure; for I would sooner fail than not be among the greatest.",
  "author": "JOHN KEATS"
}
```

Response:-

```
{
  "result": "success"
}
```

4. Once quote details are saved to database, build another API which can be used to get all the saved quotes as shown below:-

The screenshot displays a REST client interface for a request named 'MongoDBGetAllQuotes'. The request method is 'GET' and the URL is 'http://localhost:3000/quote/getAll'. The 'Params' tab is active, showing a table with one query parameter: 'Key' with a value of 'Value' and a description of 'Description'. A red arrow points from the text 'Request Method' to the 'GET' dropdown menu.

The 'Body' tab is also active, showing the JSON response in 'Pretty' format. The response is an array of two objects. The first object has a quote by 'HARPER LEE' and the second has a quote by 'JOHN KEATS'. A red arrow points from the text 'Sample Response' to the JSON body.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
[
  {
    "quote": "You never really understand a person until you consider things from his point of view. Until you climb inside of his skin and walk around in it.",
    "author": "HARPER LEE"
  },
  {
    "quote": "I was never afraid of failure; for I would sooner fail than not be among the greatest.",
    "author": "JOHN KEATS"
  }
]
```

Request URL:- `http://localhost:3000/quote/getAll`

Request Method:- GET

Sample Response JSON:-

```
[
  {
    "quote": "You never really understand a person until you consider things from his point of view. Until you climb inside of his skin and walk around in it.",
    "author": "HARPER LEE"
  },
  {
    "quote": "I was never afraid of failure; for I would sooner fail than not be among the greatest.",
    "author": "JOHN KEATS"
  }
]
```

Learnings

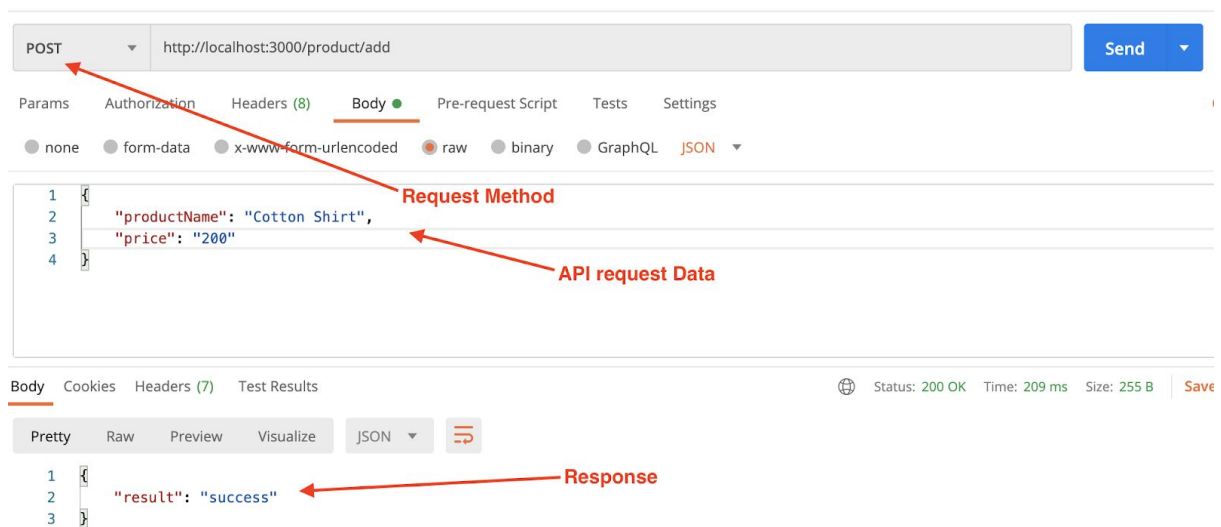
How to create APIs with database connection in express.js

Problem 3:

Build a product details API

Steps:

1. Create a connection with local or remote Mongo database
2. Build a model in express.js for quotes details
The model should have following parameters of type *string*:-
productName, price
3. Next build a route shown below and write logic to add a product detail sent from API into MongoDB database



Request URL:- http://localhost:3000/product/add

Request Method:- POST

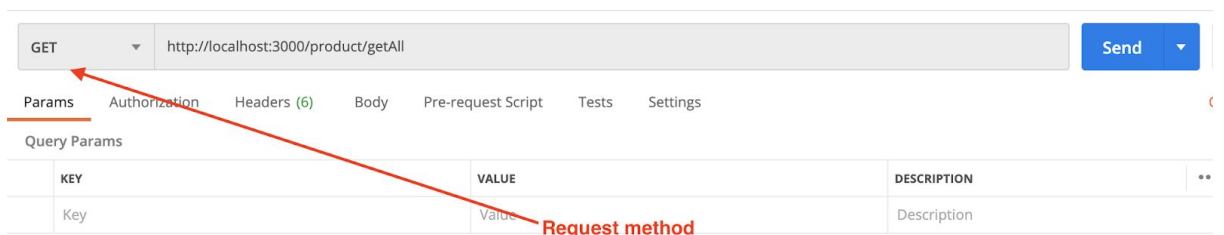
Request JSON:-

```
{
  "productName": "Cotton Shirt",
  "price": "200"
}
```

Response:-

```
{
  "result": "success"
}
```

5. Once product details are saved to database, build another API which can be used to get all the saved products as shown below:-



Request URL:- http://localhost:3000/product/getAll

Request Method:- GET

Sample Response JSON:-

```

[
  {
    "productName": "Full Shirt",
    "price": 100
  },
  {
    "productName": "Cotton Shirt",
    "price": 200
  }
]

```

Learnings

How to create APIs with database connection in express.js