**6. Spring Boot**

1. What is an API?

An API, or Application Programming Interface, is a set of defined rules that enable different applications to communicate with each other. It defines how different software components should interact and exchange data with each other, enabling them to work together seamlessly. It acts as an intermediary layer that processes data transfers between systems, letting companies open their application data and functionality to external third-party developers, business partners, and internal departments within their companies

1. What are http methods?

HTTP methods are used to perform CRUD operations on resources according to the rules and conventions of the HTTP protocol.

The most commonly used HTTP methods are:

* GET: Retrieves a representation of a resource from the server.
* POST: Submits data to the server to create a new resource.
* PUT: Updates an existing resource on the server or creates a new resource if it does not exist.
* DELETE: Deletes a resource from the server.
* PATCH: Partially updates an existing resource on the server.

1. Create a spring boot application
2. Create a new controller class
3. Add a new end-point to return and string

Text

Description automatically generated

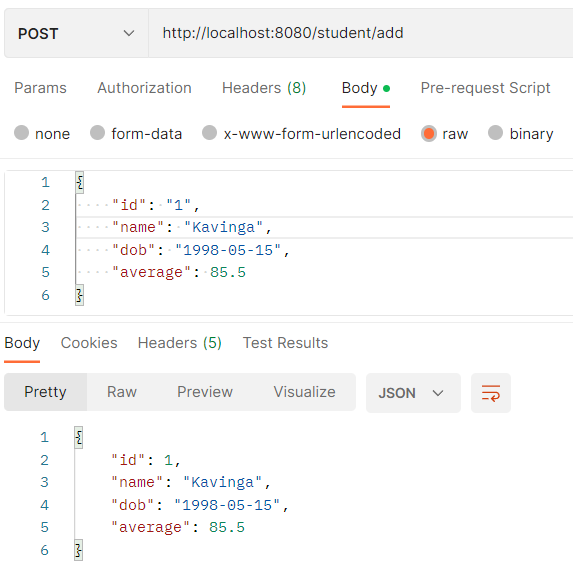
1. Test the endpoint with postman

Graphical user interface, text, application, email

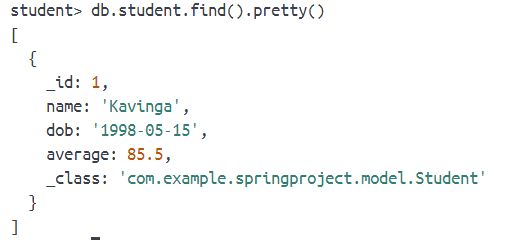
Description automatically generated

1. Create a Student class (attributes: id, name, date of birth, average)
2. Create a new controller class for students
3. Create a new service class for students
4. Add an endpoint to get list of students
5. Add an endpoint to get a student with id
6. Add new endpoint to create a student
7. Run mongodb as a docker container
8. Insert student received to endpoint created in step 12 to database

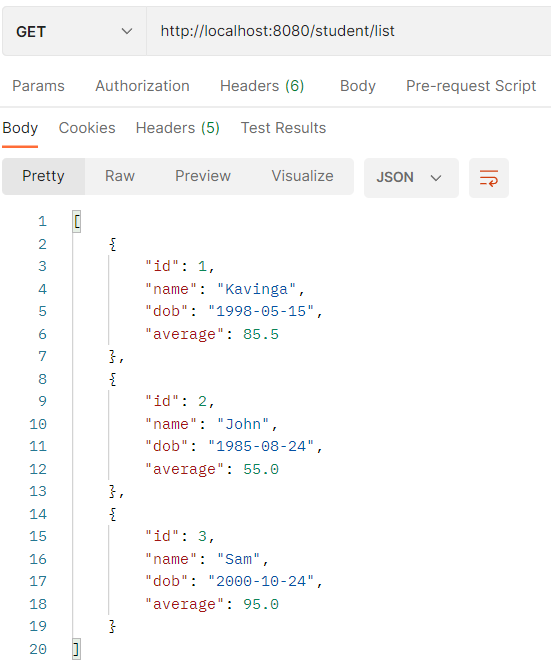
**Postman Request and Response:**

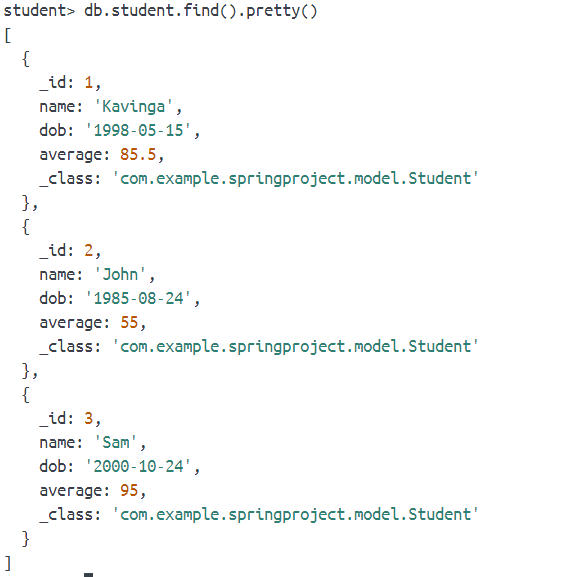


**Database after inserting:**

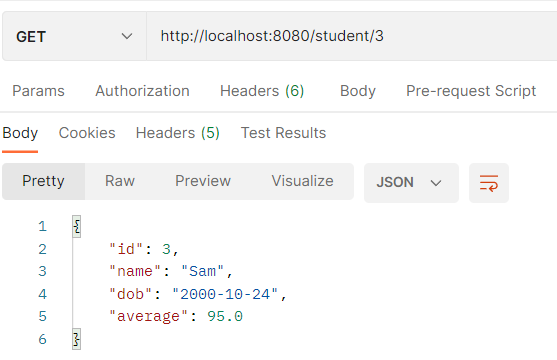


1. Extend step 10 and 11 to query data from database

**Request URL and Response for list of students: Database:**

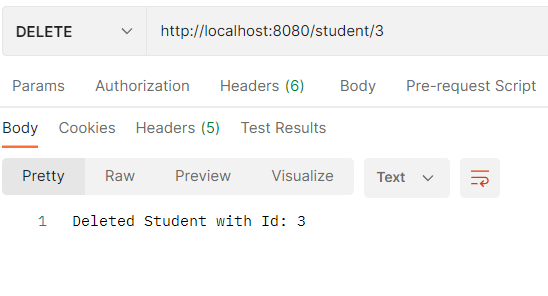


**Request URL and Response specific student:**

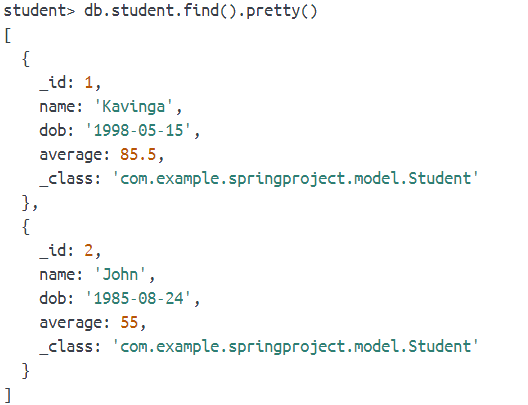


1. Add an endpoint to delete a student with id.
2. Delete the student with id from database

**Request URL and Response to delete specific student:**



**Database after Delete Student:**



1. What are http status codes

They indicate the outcome of a client's request for a resource on the server. HTTP status codes are three-digit numbers that are grouped into five classes, based on the first digit of the code, which indicates the general category of the response.

The HTTP status codes can be categorized as follows:

* Successful (2xx): These status codes indicate that the server has successfully received, understood, and accepted the client's request.
  + Examples include 200 OK, 201 Created, 204 No Content, etc.
* Redirection (3xx): These status codes indicate that the client must take additional action to complete the request.
  + Examples include 301 Moved Permanently, 302 Found, 303 See Other, etc.
* Client Errors (4xx): These status codes indicate that the client's request contains bad syntax or cannot be fulfilled by the server.
  + Examples include 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, etc.
* Server Errors (5xx): These status codes indicate that the server failed to fulfill a valid request.
  + Examples include 500 Internal Server Error, 501 Not Implemented, 502 Bad Gateway, 503 Service Unavailable, etc.

1. Briefly explain the meaning of following status codes

200, 201, 301, 400, 401, 403, 404, 405, 500, 501, 502, 503, 504

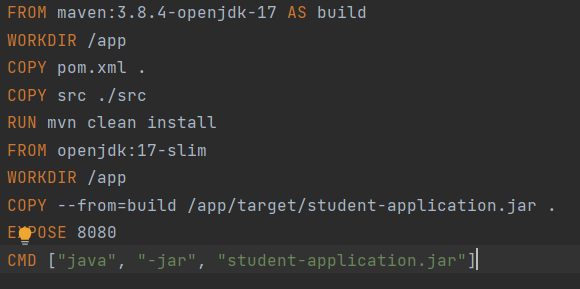
* 200 OK: This status code indicates that the client's request has been successfully fulfilled by the server, and the server is returning the requested data in the response.
* 201 Created: This status code indicates that the client's request has been successfully fulfilled, and a new resource has been created on the server as a result.
* 301 Moved Permanently: This status code indicates that the requested resource has been permanently moved to a different location. Clients should update their bookmarks or references to the new location.
* 400 Bad Request: This status code indicates that the client's request is malformed or invalid, and the server is unable to understand or process it.
* 401 Unauthorized: This status code indicates that the client's request requires authentication, and the client has not provided valid authentication credentials.
* 403 Forbidden: This status code indicates that the client's request is valid, but the server refuses to authorize it. This may be due to insufficient permissions or authentication.
* 404 Not Found: This status code indicates that the requested resource could not be found on the server. This may be due to a mistyped URL or the resource being removed or not existing.
* 405 Method Not Allowed: This status code indicates that the HTTP method used in the client's request is not allowed for the requested resource. For example, trying to use a POST request on a resource that only allows GET requests.
* 500 Internal Server Error: This status code indicates that the server encountered an error while processing the client's request. This is a generic error response that indicates something went wrong on the server side.
* 501 Not Implemented: This status code indicates that the requested resource or functionality is not implemented on the server.
* 502 Bad Gateway: This status code indicates that the server acting as a gateway or proxy received an invalid response from an upstream server.
* 503 Service Unavailable: This status code indicates that the server is currently unavailable or undergoing maintenance, and the client's request cannot be fulfilled at the moment.
* 504 Gateway Timeout: This status code indicates that the server acting as a gateway or proxy did not receive a timely response from an upstream server.

1. Using docker-compose run spring boot application and mongodb

**Docker-compose file:**



**Dockerfile to create and execute springboot project:**



1. Create new branch “spring-boot-app-v1” and push the project you created
2. Add your codes and answer sheet to a directory named “spring-boot-basic-training-v1” and push it to your training github repository
3. Create a pull request to main branch and assign it to your trainer