**5. Spring Boot**

1. What is an API?

Application programming interface is what it stands for. When two software components need to communicate with one another, an API is a collection of protocols and instructions implemented in a programming language like C++ or JavaScript. To enable users to find and obtain the necessary information, APIs operate in the background. Consider APIs as agreements that govern the communication between two software systems.

1. What are http methods?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Summary | CRUD | Accepts Request Body | Idempotnent | Safe |
| GET | To fetch a single resource or group of resources | Read | No | Yes | Yes |
| PUT | To update an entire resource in one go | Update | Yes | Yes | No |
| POST | To create a new resource | Create | Yes | No | No |
| PATCH | To partially update a resource | Update | Yes | No | No |
| DELETE | To delete a resource | Delete | No | Yes | No |
| OPTIONS | To get information on permitted operations | Read | No | Yes | Yes |
| HEAD | To get metadata of the endpoint | Read | No | Yes | Yes |
| TRACE | For diagnosing purposes | Read | No | Yes | Yes |
| CONNECT | To make the two-way connection between the client and the resource. | - | No | No | No |

1. Create a spring boot application
2. Create a new controller class
3. Add a new end-point to return and string
4. Test the endpoint with postman
5. Create a Student class (attributes: id, name, date of birth, average)
6. Create a new controller class for students
7. Create a new service class for students
8. Add an endpoint to get list of students
9. Add an endpoint to get a student with id
10. Add new endpoint to create a student
11. Run mongodb as a docker container
12. Insert student received to endpoint created in step 12 to database
13. Extend step 10 and 11 to query data from database
14. Add an endpoint to delete a student with id.
15. Delete the student with id from database
16. What are http status codes

HTTP response status codes indicate whether a specific [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP) request has been successfully completed. Responses are grouped in five classes:

[Informational responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#information_responses) (100 – 199)

[Successful responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#successful_responses) (200 – 299)

[Redirection messages](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection_messages) (300 – 399)

[Client error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#client_error_responses) (400 – 499)

[Server error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#server_error_responses) (500 – 599)

1. Briefly explain the meaning of following status codes

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

200: (OK) the request succeeded. The result meaning of "success" depends on the HTTP method

201: (Created) The request succeeded, and a new resource was created as a result. This is typically the response sent after POST requests, or some PUT requests.

301: (Moved Permanently) The URL of the requested resource has been changed permanently. The new URL is given in the response.

400: (Bad Request) the server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

401: (Unauthorized) although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.

403: (Forbidden) The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource. Unlike 401 Unauthorized, the client's identity is known to the server.

404: (Not Found) The server cannot find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 forbidden to hide the existence of a resource from an unauthorized client. This response code is probably the most well-known due to its frequent occurrence on the web.

405: (Method Not Allowed) The request method is known by the server but is not supported by the target resource. For example, an API may not allow calling DELETE to remove a resource.

500: (Internal Server Error) the server has encountered a situation it does not know how to handle.

501: (Not Implemented) the request method is not supported by the server and cannot be handled. The only methods that servers are required to support (and therefore that must not return this code) are GET and HEAD.

502: (Bad Gateway) this error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.

503: (Service Unavailable) The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded. Note that together with this response, a user-friendly page explaining the problem should be sent. This response should be used for temporary conditions and the Retry-After HTTP header should, if possible, contain the estimated time before the recovery of the service. The webmaster must also take care about the caching-related headers that are sent along with this response, as these temporary condition responses should usually not be cached.

504: (Gateway Timeout) this error response is given when the server is acting as a gateway and cannot get a response in time.

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

