**Representational State Transfer(REST)-API using spring boot**

J2EE standard architecture: -

Let deploying an application in a server and client request using browser via desktop,mobile,etc. The request is being sent. The presentation layer(controller) accepts the request and moves to service layer which provide services and business logic. The Dao/Repository layer works as taking out and putting into the required data from the database and provides data for business logic.

Key Points:

The Controller accepts the user request.

The Java Class being managed by the service layer.

The data being accessed by Dao Layer which requires to service layer.

Finally, the fetched data being sent to client or user via response.

Diagram

Description automatically generated

**API URLs for Course Application: -**

|  |  |  |
| --- | --- | --- |
| **Method** | **API URLs**  **URLs starts with base URL** | **Operations** |
| GET | /courses | Get all course |
| GET | /courses/{courseId} | Get Single Course of given ID in URL |
| POST | /courses | Add new course |
| PUT | /courses | Update the course |
| DELETE | /courses/{courseId} | Delete the course ID |

IDE being used as Spring Tool Suite 4

API Calls via POSTMAN

Local Port being used as 9090.

Package being used as “com.springrest.springrest”.

“SpringrestApplication.java” is the starting point of the java running program.

“pom.xml”, few dependencies are taking into consideration like Spring Web, MySQL Driver and Spring Data JPA.

The “application.properties” file states various conditions as properties for the project.

**API requests defined**

APIs provide a structured way for one application to access the capabilities of another application. Typically, this communication happens over the internet by way of an API server. A client application (like a mobile app) sends a request to the server, and after the request processes the server returns a response to the client.

A request includes the URL of the API endpoint and an HTTP request method. The method indicates the action you want the API to perform. Here are some of the most common methods:

GET retrieves data from an API.

POST sends new data to an API.

PATCH and PUT update existing data.

DELETE removes existing data.

**Sending a request**

When you're ready, open Postman and send your first request.

1. Select **+** in the workbench to open a new tab.
2. Enter postman-echo.com/get for the request URL.
3. Select **Send**.