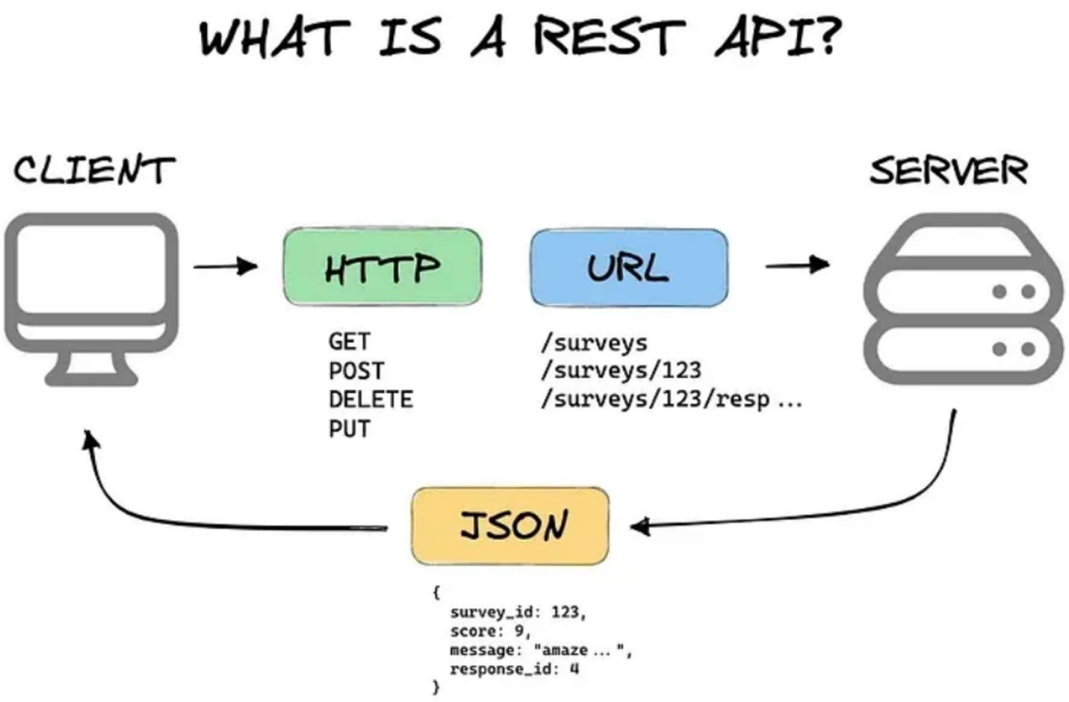
REST API

What is REST API :-

- REST API (**Representational State Transfer API**) is a way for **two systems to communicate with each other** over the internet using the **HTTP protocol.**



- **REST APIs allow** clients (like Postman, mobile apps, web apps) to perform actions such as:

  - Saving data
  - Reading data
  - Updating data
  - Deleting data

Client sends a request → Server sends a response in JSON format.

Key Concepts of REST API :-

1. Client–Server Architecture
2. Stateless
3. Resource-Based
4. HTTP Methods
5. JSON Format
6. Uniform Interface
7. Layered System

HTTP Methods :-

Method	Meaning	Example
GET	Read Data	GET / students
POST	Create Data	POST / students
PUT	Update full data	PUT / students / 5
PATCH	Update part of data	PATCH / students / 5
DELETE	Delete data	DELETE / students / 5

Short Summary :-

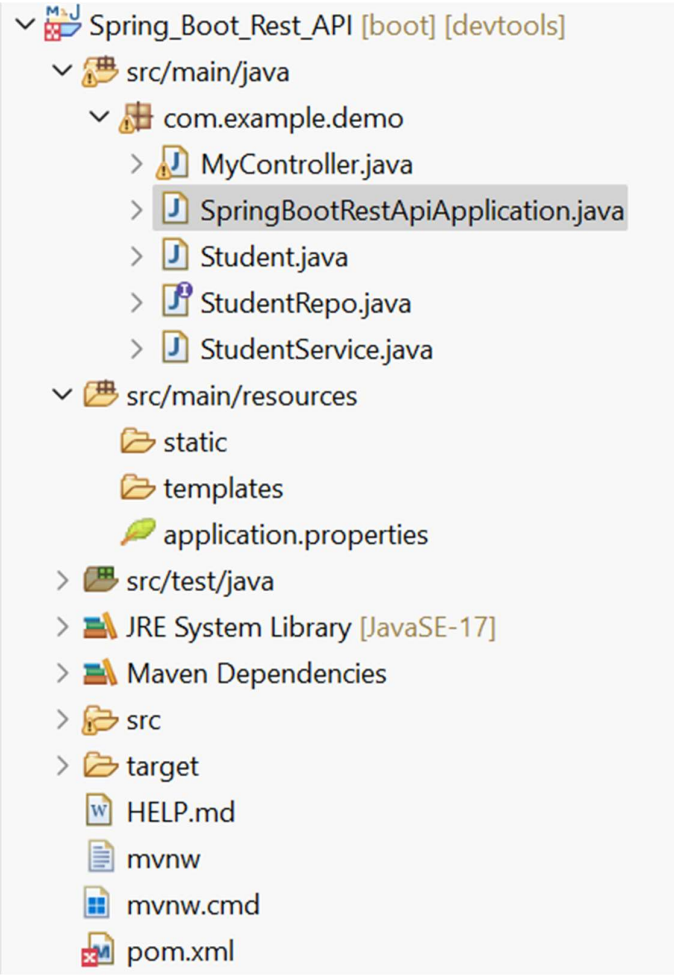
- **REST API:** A way to communicate between client and server using HTTP.
- **Stateless:** Every request is independent.
- **Resource-based:** Data = resources accessed via URLs.
- **Uses HTTP Methods:** GET, POST, PUT, DELETE.
- **Uses JSON:** For sending data.
- **Simple, scalable, fast.**

Spring Boot + REST API

Project Description :-

- This project is a **Spring Boot REST API** for handling **Student Registration**.
- It allows a client (Postman, React, Android, etc.) to **send student details**, and the API will store student information in the **database using Spring Data JPA.**

Project Structure :-



Step 1 :- pom.xml

```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>4.0.0</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>Spring_Boot_Rest_API</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>Spring_Boot_Rest_API</name>
    <description>Demo project for Spring Boot</description>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies> Add Spring Boot Starters...
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-webmvc</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-webmvc-test</artifactId>
            <scope>test</scope>
        </dependency>


        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
```

```
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <!-- New Added -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

Step 2 :- SpringBootRestApiApplication.java

```java
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringBootRestApiApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringBootRestApiApplication.class, args);
    }

}
```

Step 3 :- Student.java

```java
package com.example.demo;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity //Table Create
@Table(name="student")  //Table Name
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY) //Autoincrement
    private Long id;

    @Column //Create Column
    private String fullname;

    @Column(unique=true)
    private String email;

    @Column
    private String password;

    @Column
    private String address;

    public Long getId() {
```

```
31          return id;
32      }
33
34      public void setId(Long id) {
35          this.id = id;
36      }
37
38      public String getFullname() {
39          return fullname;
40      }
41
42      public void setFullname(String fullname) {
43          this.fullname = fullname;
44      }
45
46      public String getEmail() {
47          return email;
48      }
49
50      public void setEmail(String email) {
51          this.email = email;
52      }
53
54      public String getPassword() {
55          return password;
56      }
57
58      public void setPassword(String password) {
59          this.password = password;
60      }
61
62      public String getAddress() {
63          return address;
64      }
65
66      public void setAddress(String address) {
67          this.address = address;
68      }
69
70      @Override
71      public String toString() {
72          return "Student [id=" + id + ", fullname=" + fullname + ", email=" + email + ", password=" + password
73                  + ", address=" + address + "]";
74      }
75  }
```

Step 4 :- StudentRepo.java

```java
package com.example.demo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface StudentRepo extends JpaRepository<Student, Long>{


}
```

Step 5 :- StudentService.java

```java
package com.example.demo;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class StudentService {

    @Autowired
    StudentRepo sr;

    public void registration(Student s)
    {
        sr.save(s); //Insert Query Directly
    }

    public List<Student> getallstudents()
    {
        return sr.findAll();    //select * from student
    }
}
```

Step 6 :- MyController.java

```java
package com.example.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

@RestController
@CrossOrigin("*")
public class MyController {

    @GetMapping("/")
    public ResponseEntity<?> test()
    {
        return ResponseEntity.status(HttpStatus.ACCEPTED).body("Tested Ok");
    }

    @Autowired
    StudentService ss;

    @PostMapping("/register")
    public ResponseEntity<?> registration(@RequestBody Student s)
    {
        try
        {
            ss.registration(s);
            return ResponseEntity.status(HttpStatus.ACCEPTED).body("Registration Successfully Completed");
        }
        catch(Exception e)
        {
            return ResponseEntity.status(HttpStatus.BAD_REQUEST).body("Fail");
        }
    }
}
```

Step 7 :- application.properties

```
spring.application.name=Spring_Boot_Rest_API

spring.datasource.url=jdbc:mysql://localhost:3309/springbootdb1
spring.datasource.username=root
spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```

Output :-

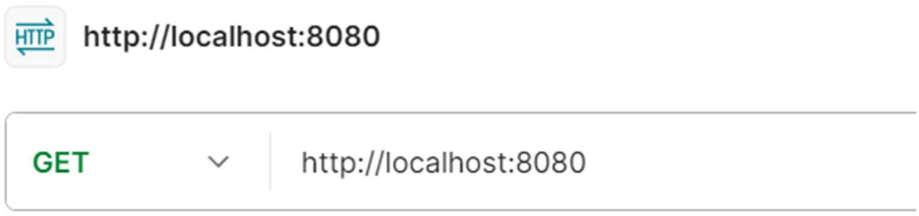
```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v4.0.0)

2025-12-09T15:33:20.124+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] c.e.demo.SpringBootRestApiApplication
2025-12-09T15:33:20.133+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] c.e.demo.SpringBootRestApiApplication
2025-12-09T15:33:20.253+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor
2025-12-09T15:33:20.254+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor
2025-12-09T15:33:21.570+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate
2025-12-09T15:33:21.681+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate
2025-12-09T15:33:22.693+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.s.boot.tomcat.TomcatWebServer
2025-12-09T15:33:22.723+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.apache.catalina.core.StandardService
2025-12-09T15:33:22.724+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.apache.catalina.core.StandardEngine
2025-12-09T15:33:22.840+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] b.w.c.s.WebApplicationContextInitializer
2025-12-09T15:33:23.191+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.hibernate.jpa.internal.util.LogHelper
2025-12-09T15:33:23.366+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] org.hibernate.Version
2025-12-09T15:33:24.552+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo
2025-12-09T15:33:24.637+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] com.zaxxer.hikari.HikariDataSource
2025-12-09T15:33:24.967+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] com.zaxxer.hikari.pool.HikariPool
2025-12-09T15:33:24.971+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] com.zaxxer.hikari.HikariDataSource
2025-12-09T15:33:25.089+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [  restartedMain] org.hibernate.dialect.Dialect
2025-12-09T15:33:25.143+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] org.hibernate.orm.connections.pooling
        Database JDBC URL [jdbc:mysql://localhost:3309/springbootdb]
        Database driver: MySQL Connector/J
        Database dialect: MySQLDialect
        Database version: 5.5.5
        Default catalog/schema: springbootdb1/undefined
        Autocommit mode: undefined/unknown
        Isolation level: REPEATABLE_READ [default REPEATABLE_READ]
        JDBC fetch size: none
        Pool: DatasourceConnectionProviderImpl
        Minimum pool size: undefined/unknown
        Maximum pool size: undefined/unknown
2025-12-09T15:37:38.357+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator
2025-12-09T15:37:38.394+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] j.LocalContainerEntityManagerFactoryBean
2025-12-09T15:37:38.511+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [  restartedMain] JpaBaseConfiguration$JpaWebConfiguration
2025-12-09T15:37:38.767+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] o.s.boot.tomcat.TomcatWebServer
2025-12-09T15:37:38.775+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] c.e.demo.SpringBootRestApiApplication
2025-12-09T15:37:38.778+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [  restartedMain] .ConditionEvaluationDeltaLoggingListener
2025-12-09T15:38:06.301+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-12-09T15:38:06.301+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet
2025-12-09T15:38:06.303+05:30  INFO 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet
Hibernate:
    insert
    into
        student
        (address, email, fullname, password)
    values
        (?, ?, ?, ?)
2025-12-09T15:38:06.318+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-2] org.hibernate.orm.jdbc.error
2025-12-09T15:38:06.319+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-2] org.hibernate.orm.jdbc.error
Hibernate:
    insert
    into
        student
        (address, email, fullname, password)
    values
        (?, ?, ?, ?)
2025-12-09T15:38:10.018+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-6] org.hibernate.orm.jdbc.error
2025-12-09T15:38:10.019+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-6] org.hibernate.orm.jdbc.error
Hibernate:
    insert
    into
        student
        (address, email, fullname, password)
    values
        (?, ?, ?, ?)
2025-12-09T15:38:12.265+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-8] org.hibernate.orm.jdbc.error
2025-12-09T15:38:12.265+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [nio-8080-exec-8] org.hibernate.orm.jdbc.error
Hibernate:
    insert
    into
        student
        (address, email, fullname, password)
    values
        (?, ?, ?, ?)
2025-12-09T15:38:14.209+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [io-8080-exec-10] org.hibernate.orm.jdbc.error
2025-12-09T15:38:14.210+05:30  WARN 29128 --- [Spring_Boot_Rest_API] [io-8080-exec-10] org.hibernate.orm.jdbc.error
Hibernate:
    insert
    into
        student
        (address, email, fullname, password)
    values
```

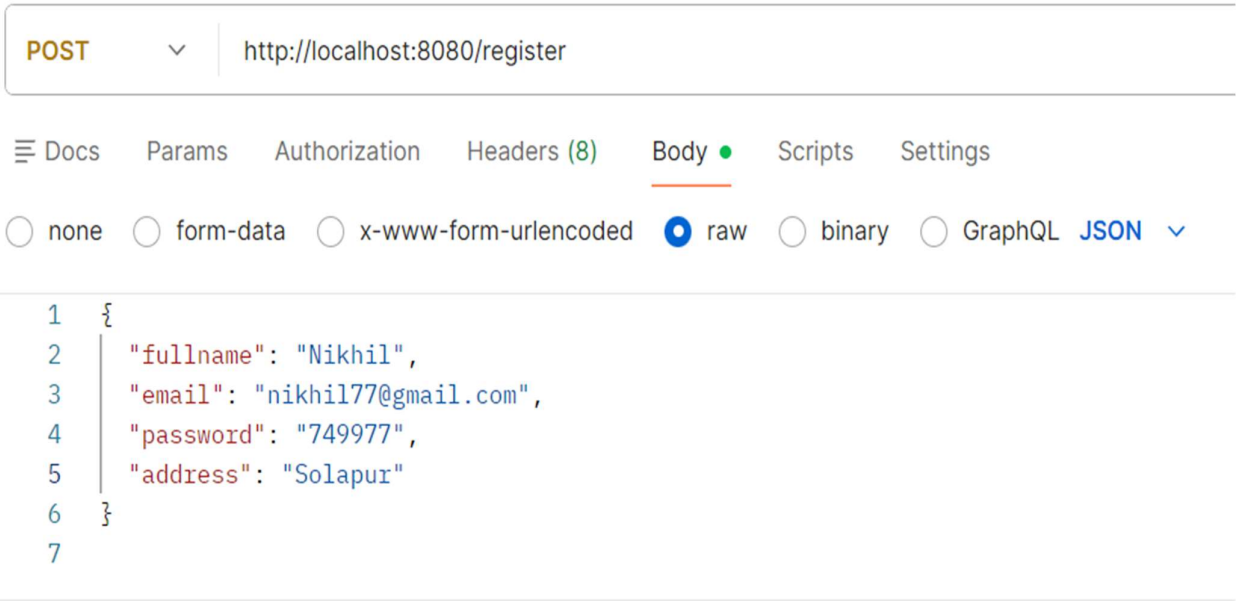
Server Output :-

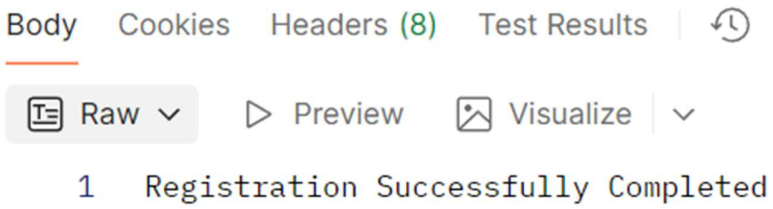
The app runs at http://localhost:8080

Tested Ok

Postman :-







Database :-

id	address	email	fullname	password
3	Solapur	abc@gmail.com	abc	56789
4	Pune	xyz@gmail.com	xyz	749976
6	Pune	nikhil1234@gmail.com	Nikhil 1	123456
18	pune	abc2121@gmail.com	abcd xyz	12345
21	Solapur	nikhil77@gmail.com	Nikhil	749977