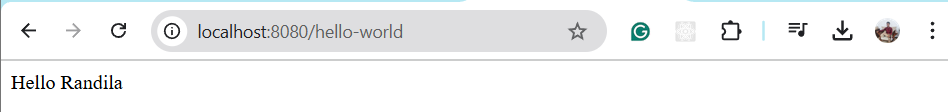
**JAVA SPRING CRUD APPLICATION**

1. First create a folder structure and java-spring from <https://start.spring.io/> and add web dependencies for that and also after run the SpringbootRestApiApplication from inteliJ. This sprinboot is automatically added embedded tomcat server.  
     
   **Rest API**
2. Then create a controller inside the net.Thatfile and create new classes inside of that  
   and also add @RestController beginning of the folder,   
   @RestController is a **specialized version of @Controller** that automatically serializes responses into JSON or XML without requiring @ResponseBody on each method.  
   default port is 8080 in tomcat embedded system in spring
3. @GetMapping("/hello-world")  
   public String helloWorld(){  
    return "Hello Randila";  
   }

Add above @GetMapping to the function, then user can access that get Request using  
<http://localhost:8080/hello-world>



And also we can get some set of details like this, but before using this you need to create that class and constructors and also getters and setters as well  
  
@RestController  
public class StudentController {  
  
 @GetMapping("/students") // Changed endpoint to `/students`  
 public List<Student> getStudents() {  
 List<Student> students = new ArrayList<>();  
  
 students.add(new Student(1, "Randila", "Chathuranga"));  
 students.add(new Student(2, "John", "Doe"));  
  
 return students; // Return list of students  
 }  
}

1. And we can use specific path variable using this  
     
   //spring boot rest API with path variable  
   @GetMapping("/students/{id}")  
   public Student studentPathvariable(@PathVariable(“id”) int id){  
    return new Student(id,"Bhagi","Randi");  
   }  
   A screenshot of a computer

   AI-generated content may be incorrect.
2. And also we can use RequestParam to the this option  
   // Spring Boot REST API with Request Param  
   // Example: http://localhost:8080/students/query?id=1  
   @GetMapping("/students/query") // Added leading `/` for clarity  
   public Student getStudentById(@RequestParam(name = "id", required = true) int id) {  
    return new Student(id, "Randila", "Chathuranga");  
   }