💠

02 Section 2 Using the Property File.

📚 Introduction

In **Spring Boot**, `application.properties` (or `application.yml`) is used to store configuration settings in a **key-value pair** format. This allows us to **externalize configurations**, making our application more flexible and maintainable.

🛠️ Adding Configuration to `application.properties`

Let's define a key-value pair inside our `application.properties` file:

```
my.greeting=hello
```

Now, we will fetch this value inside our **Spring Boot controller** and return it as a REST API response.

🔧 Injecting Property Values in Controller

Spring Boot provides the `@Value` annotation to **inject values from the property file into Java variables.**

🔷 Steps to Fetch the Property Value:

1. **Create a member variable** inside the controller.
2. **Use `@Value` annotation** to fetch the value from `application.properties`.
3. **Expose an API endpoint** to return the value.

📝 Example Code:

```
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class GreetingController {

    // Injecting the value from application.properties
    @Value("${my.greeting}")
    private String greetMessage;

    // Exposing a GET API to return the greeting
    @GetMapping("/greeting")
    public String greeting() {
        return greetMessage;
    }
}
```

🔎 Understanding the Code:

- `@Value("${my.greeting}")` → Fetches the value of `my.greeting` from `application.properties`.
- `private String greetMessage;` → Stores the value retrieved from properties.
- `@GetMapping("/greeting")` → Exposes a REST API endpoint (`/greeting`) to return the greeting message.
- When a request is sent to `http://localhost:8080/greeting`, it responds with **"hello"**.

🎉 What Happens When We Package the Application?

When we create a **JAR file**, `application.properties` is **bundled inside the JAR**. This means the configurations remain the same, even in different environments.

🔍 How to Override Properties in Production?

📢 Stay Tuned for Section 3! 🚀