Application.properties vs Application.yml.

① Syntax —

```
server.port = 8080
spring.app.name = Day4
spring.owner.name = PSR
```

② Pros —

- Simple key-value style
- Easy for small proj.
- Familiar for older spring versions

③ Cons —

- Becomes mess for nested config's.
- Hard to visually organise env.

① Syntax —

```
Server:
    port: 8081
spring:
    app:
        name: Day4
    owner:
        name: PSR.
```

② Pros —

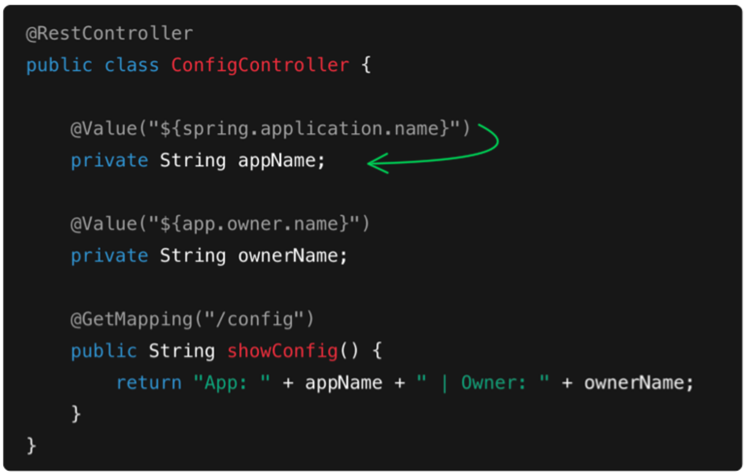
- Human readable
- Nested hierarchy
- Cleaner for multi-env setups
- Supports multiple profil

③ Cons —

- Identat^n sensitive.

→ @Value

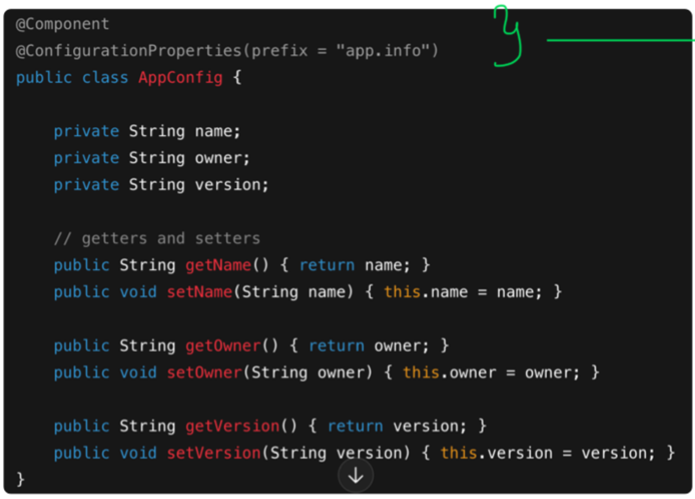
- used to inject config values directly inside code.



```java
@RestController
public class ConfigController {

    @Value("${spring.application.name}")
    private String appName;

    @Value("${app.owner.name}")
    private String ownerName;

    @GetMapping("/config")
    public String showConfig() {
        return "App: " + appName + " | Owner: " + ownerName;
    }
}
```

→ @ConfigurationProperties.

When you have multiple config and a large project then it is used.



```java
@Component
@ConfigurationProperties(prefix = "app.info")
public class AppConfig {

    private String name;
    private String owner;
    private String version;

    // getters and setters
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getOwner() { return owner; }
    public void setOwner(String owner) { this.owner = owner; }

    public String getVersion() { return version; }
    public void setVersion(String version) { this.version = version; }
}
```

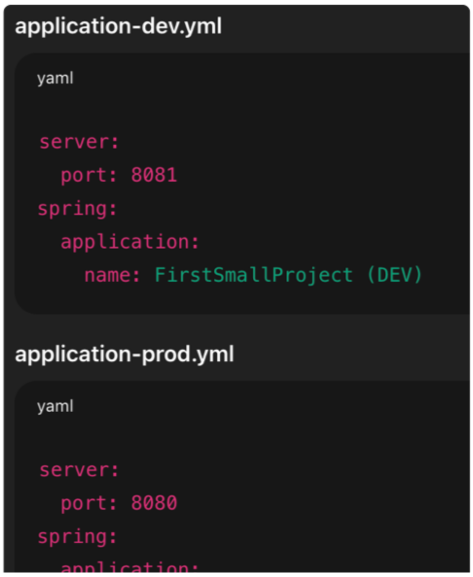
→ This component defines @ config.... for all App Config.

Creating Diffrent profiles

Because we do not use same settings for all env. therefore we create different profiles

→ along with Application.yml create:

- Application-dev.yml
- Application-prod.yml.



application-dev.yml

```yaml
server:
  port: 8081
spring:
  application:
    name: FirstSmallProject (DEV)
```

application-prod.yml

```yaml
server:
  port: 8080
spring:
  application:
```

```
Spring:
  profile:
    active: Dev
```

Defines Dev as active user.

```
  application:
    name: FirstSmallProject (PROD)
```