**Injecting Custom Application Properties**

**Spring Boot Custom Application Properties**:

Here we make our application configurable. No hard-coding of values. Need to read app configuration from a properties file. Then the solution is "Application Properties File".

**Application Properties File**:

* By default, Spring reads information from a standard properties file.
* This file is located at: "src/main/resources/application.properties"
* We can define any custom properties in the file
* Our Spring Boot app can access properties using **@Value**

**Development process**:

1. Define custom properties in application.properties.
2. Inject properties into Spring Boot application using @Value

**1) Define custom properties in application.properties**:

#

# Define My application properties

#

coach.name = Micky Mouse

team.name = The Mouse Club

**2) Inject properties into Spring Boot application using @Value**:

**package** com.ruhul.springboot.demo.mycoolapp.rest;

**import** java.time.LocalDateTime;

**import** org.springframework.beans.factory.annotation.Value;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

**public** **class** FunRestController {

// inject properties for: coach.name and team.name

@Value("${coach.name}")

**private** String coachName;

@Value("${team.name}")

**private** String teamName;

// expose new endpoint for "teaminfo"

@GetMapping("teaminfo")

**public** String getTeaminfo() {

**return** "Coach name: " + coachName + ", Tean Name: " + teamName;

}

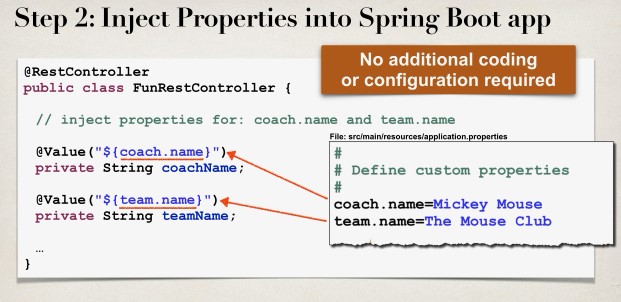
@GetMapping("/")

**public** String sayHello() {

**return** "Hello World! Time on server: " + LocalDateTime.*now*();

}

}



Injecting Custom Application Properties