Spring Boot - Custom Application Properties

Problem

Problem

- You need for your app to be configurable … no hard-coding of values

Problem

- You need for your app to be configurable … no hard-coding of values

- You need to read app configuration from a properties file

Solution: Application Properties file

Solution: Application Properties file

- By default, Spring Boot reads information from a standard properties file

Solution: Application Properties file

- By default, Spring Boot reads information from a standard properties file
  - Located at: **`src/main/resources/application.properties`**

Solution: Application Properties file

- By default, Spring Boot reads information from a standard properties file
  - Located at: **`src/main/resources/application.properties`**

Standard Spring Boot file name

Solution: Application Properties file

- By default, Spring Boot reads information from a standard properties file
  - Located at: **`src/main/resources/application.properties`**
- You can define ANY custom properties in this file

Standard Spring Boot file name

Solution: Application Properties file

- By default, Spring Boot reads information from a standard properties file
  - Located at: **`src/main/resources/application.properties`**
- You can define ANY custom properties in this file
- Your Spring Boot app can access properties using **`@Value`**

Standard Spring Boot file name

Solution: Application Properties file

- By default, Spring Boot reads information from a standard properties file
  - Located at: **`src/main/resources/application.properties`**
- You can define ANY custom properties in this file
- Your Spring Boot app can access properties using **`@Value`**

Standard Spring Boot file name

No additional coding or configuration required

Development Process

Step-By-Step

Development Process

1. Define custom properties in **`application.properties`**

Development Process

1. Define custom properties in **`application.properties`**

2. Inject properties into Spring Boot application using **`@Value`**

Step 1: Define custom application properties

Step 1: Define custom application properties

File: src/main/resources/application.properties

```
#
# Define custom properties
#
```

Step 1: Define custom application properties

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
```

Step 1: Define custom application properties

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```

Step 1: Define custom application properties

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```

You can give ANY custom property names

Step 2: Inject Properties into Spring Boot app

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

}
```

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

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

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

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

  @Value("${coach.name}")
  private String coachName;
```

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

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

  @Value("${coach.name}")
  private String coachName;
```

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

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

  @Value("${coach.name}")
  private String coachName;
```

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

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

  @Value("${coach.name}")
  private String coachName;

  @Value("${team.name}")
  private String teamName;

  ...
}
```

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```

Step 2: Inject Properties into Spring Boot app

```
@RestController
public class FunRestController {

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

  @Value("${coach.name}")
  private String coachName;

  @Value("${team.name}")
  private String teamName;

  ...
}
```

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```

Step 2: Inject Properties into Spring Boot app

No additional coding or configuration required

```
@RestController
public class FunRestController {

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

  @Value("${coach.name}")
  private String coachName;

  @Value("${team.name}")
  private String teamName;

  ...
}
```

File: src/main/resources/application.properties

```
#
# Define custom properties
#
coach.name=Mickey Mouse
team.name=The Mouse Club
```